
NCSTAT Documentation

Release 2.3.0

Pascal Terray (LOCEAN/IPSL and IRD)

May 05, 2026

CONTENTS

1	Introduction	1
1.1	Presentation	1
1.2	Features	2
1.3	Installation	3
1.4	Usage	9
2	Reference manual	25
2.1	comp_clim_3d	25
2.2	comp_clim_4d	30
2.3	comp_clim_miss_3d	35
2.4	comp_clim_miss_4d	40
2.5	comp_composite_3d	45
2.6	comp_composite_4d	52
2.7	comp_composite_miss_3d	59
2.8	comp_cor_1d	65
2.9	comp_cor_3d	73
2.10	comp_cor_4d	83
2.11	comp_cor_miss_3d	92
2.12	comp_cross_spectrum_1d	99
2.13	comp_cross_spectrum_3d	107
2.14	comp_cross_spectrum2_3d	116
2.15	comp_eof_3d	125
2.16	comp_eof_4d	136
2.17	comp_eof_miss_3d	148
2.18	comp_fftfiler_1d	156
2.19	comp_fftfiler_3d	160
2.20	comp_fftfiler_4d	165
2.21	comp_filt_rot_eof_3d	169
2.22	comp_filt_rot_eof_4d	176
2.23	comp_freq_func_1d	183
2.24	comp_index_1d	187
2.25	comp_integral_4d	191
2.26	comp_integral_miss_4d	194
2.27	comp_invert_eof_3d	198
2.28	comp_invert_eof_4d	203
2.29	comp_lanczos_filter_1d	207
2.30	comp_lanczos_filter_3d	211
2.31	comp_lanczos_filter_4d	217
2.32	comp_loess_rot_eof_3d	222
2.33	comp_loess_rot_eof_4d	229

2.34	comp_mask_3d	236
2.35	comp_mask_4d	239
2.36	comp_norm_3d	243
2.37	comp_norm_4d	248
2.38	comp_norm_miss_3d	254
2.39	comp_norm_miss_4d	259
2.40	comp_ortho_rot_eof_3d	265
2.41	comp_ortho_rot_eof_4d	271
2.42	comp_project_eof_3d	278
2.43	comp_project_eof_4d	283
2.44	comp_reg_1d	289
2.45	comp_reg_3d	296
2.46	comp_reg_4d	304
2.47	comp_season_3d	312
2.48	comp_season_4d	315
2.49	comp_season_miss_3d	318
2.50	comp_season_miss_4d	322
2.51	comp_section_3d	326
2.52	comp_section_4d	330
2.53	comp_section_miss_3d	335
2.54	comp_serie_3d	339
2.55	comp_serie_4d	344
2.56	comp_serie_miss_3d	348
2.57	comp_spectrum_1d	352
2.58	comp_spectrum_diff_3d	358
2.59	comp_spectrum_ratio_1d	366
2.60	comp_spectrum_ratio_3d	371
2.61	comp_spectvar_1d	376
2.62	comp_spectvar_3d	382
2.63	comp_spectvar_4d	389
2.64	comp_stat_3d	397
2.65	comp_stat_4d	404
2.66	comp_stat_miss_3d	411
2.67	comp_stl_1d	417
2.68	comp_stl_3d	422
2.69	comp_stl_4d	427
2.70	comp_svd_3d	433
2.71	comp_svd2_3d	448
2.72	comp_symlin_filter_1d	459
2.73	comp_symlin_filter_3d	464
2.74	comp_symlin_filter_4d	469
2.75	comp_trend_1d	475
2.76	comp_trend_3d	479
2.77	comp_trend_4d	483
2.78	pack_masked_data_3d	488
2.79	unpack_masked_data_3d	491

Bibliography	495
---------------------	------------

Index	499
--------------	------------

INTRODUCTION

1.1 Presentation

The NCSTAT software is a collection of many operators for complex statistical processing and analysis of huge climate model outputs and datasets. These statistical tools are written in pure and portable Fortran95/2003 [[fortran](#)] using the NetCDF Fortran90 interface [[netcdf-f90](#)] of the NetCDF library [[netcdf](#)] for input/output data transfer, the OpenMP API [[openmp](#)] for parallel reading of NetCDF files and the STATPACK software [[statpack](#)] for numerical and parallel computations.

Each NCSTAT operator is a stand-alone UNIX command line program executed at the shell-level like, e.g., **ls** or **mkdir**. The NCSTAT operators take NetCDF files as input, perform an operation or a statistical task (e.g., averaging, computing vertical integrals or computing a Principal Component Analysis for example), and produce one or several NetCDF files as output. There are some restrictions for NetCDF datasets that can be processed with NCSTAT. First, NetCDF datasets are only supported for the classic data model and arrays up to 4 dimensions. Most NCSTAT operators also assume that these dimensions are used by the horizontal and vertical grid and the time associated with climate data.

The NCSTAT operators are primarily designed to aid manipulation and complex analysis of climate data at a higher level than the famous NCO [[nco](#)] and CDO [[cdo](#)] softwares already commonly used in the climate community. In deed, after some experience, most of the users will find that NCO, CDO and NCSTAT are not concurrent, but very complementary softwares.

The main characteristics of NCSTAT are:

- Very simple UNIX command line interface like NCO [[nco](#)] and CDO [[cdo](#)]
- Operators can be combined in a flexible way to produce sophisticated statistical analysis including univariate statistics, multivariate analysis, time series, power and cross-power spectrum analyses, filtering and trend computations, correlation and regression analyses of multi-dimensional arrays. Detailed statistical testing of the results is also available in most operators
- Fast parallel and out-of-core processing of large datasets
- Many operators handle datasets with missing values
- Support of many different grid types explicitly or implicitly by the use of mesh-mask files
- Tested on many UNIX/Linux systems and MacOS-X

1.2 Features

1.2.1 Language

NCSTAT is developed in the Fortran95/2003 language [[fortran](#)] and takes full advantage of the novel language features such as interface overloading, kind types, modules, etc. without any obsolescent Fortran77 features. Thus, you must have access to a Fortran95/2003 compiler in order to build the NCSTAT software.

NCSTAT has been built successfully on a variety of UNIX systems (including MacOS-X and AIX) with different Fortran95/2003 compilers and is believed to be a portable software.

1.2.2 Parallelism

NCSTAT is a parallel, multi-threaded software based on the OpenMP standard [[openmp](#)]. Therefore, it will run on multi-core or, more generally, shared memory multi-processor computers. NCSTAT does not run on distributed memory (e.g., clusters) parallel computers and does not use Graphics Processing Units (GPUs), but this possibility may be included in future releases of NCSTAT.

Support for at least OpenMP 3 is requested for activation of OpenMP parallelism in NCSTAT. However, it is also possible to build sequential versions of NCSTAT (e.g., if OpenMP compilation is not activated or if an OpenMP-enabled Fortran compiler is not available), even if it is not recommended for efficiency reasons. The best place to view OpenMP support by a large range of Fortran compilers is [OpenMP compilers](#).

More precisely, parallelism in NCSTAT is achieved and can be controlled at different levels:

- 1) By the use of the STATPACK software [[statpack](#)] for the computational routines used in NCSTAT. The STATPACK software provides parallel, multi-threaded subroutines and functions for nearly all the computations done in NCSTAT. STATPACK is also written in pure and portable Fortran 95/2003 and based on the OpenMP standard. If a multi-threaded version of STATPACK is used, NCSTAT will inherit automatically the parallel capacity and efficiency of STATPACK.
- 2) By the (optional) use of an optimized and multi-threaded BLAS library [[blas](#)], such the open-source OpenBLAS library [[openblas](#)] or BLIS library [[blis](#)] or, alternatively, vendor-like BLAS libraries [[accelerate](#)] [[aocl](#)] [[arm](#)] [[mkl](#)], in addition to the STATPACK software. Some of the operators available in NCSTAT, such as [comp_svd_3d](#), can greatly benefit from such an optimized and multi-threaded BLAS library. NCSTAT will use directly a BLAS library if the UNIX preprocessor **cpp** macro, `_BLAS`, is defined at compilation of NCSTAT (see the section [Preprocessor cpp macros](#) for more details).
- 3) By allowing parallel reading of NetCDF files based on the OpenMP standard [[openmp](#)] when possible (e.g. if input NetCDF files are classic, 64-bit offset or cdf5 format files). NCSTAT will perform parallel reading of NetCDF files based on the OpenMP standard if the UNIX preprocessor **cpp** macro, `_PARALLEL_READ`, is defined at compilation of NCSTAT and input NetCDF files are not NetCDF-4/HDF5 format files (see the section [Preprocessor cpp macros](#) for details).

In the general case, the user is fully responsible for activating the threaded capabilities of their BLAS and STATPACK libraries, and NCSTAT software by using appropriate Shell (e.g., OpenMP) environment variables before executing the NCSTAT operators. More details on how to activate OpenMP support when building NCSTAT is given in sections [OpenMP compilation](#) and [CCP macros](#) below. The basic procedure for activating OpenMP parallelism, when executing the NCSTAT operators, is described in the section [Parallel execution](#).

1.2.3 Dependencies

The following additional libraries are required and must be available or installed before compiling and using NCSTAT:

- 1) The Unidata NetCDF library [[netcdf](#)], including the NetCDF Fortran90 interface [[netcdf-f90](#)] (from version 3.5 of the NetCDF library). This library is needed to read and write NetCDF files with NCSTAT. Do not use a MPI-based parallel version of the NetCDF library, such as [[pnetcdf](#)], with NCSTAT. Since NCSTAT is using OpenMP parallelism and not MPI [[mpi](#)], NCSTAT will not compile correctly with the current parallel versions of the NetCDF library, which are all based on MPI. Information about installing NetCDF and its Fortran90 interface is available at [NETCDF](#) and several other web sites such as [Libs4cdo](#).
- 2) The STATPACK library [[statpack](#)]. This library is required for all the mathematical and statistical computations performed in NCSTAT. At least version 2.3 of STATPACK is required for a successful compilation of version 2.3 of NCSTAT, but version 2.4 of STATPACK is recommended for efficiency reasons. Note, that the STATPACK library also determines the precision (e.g., single, double or eventually extended) of the computations performed in NCSTAT since NCSTAT directly uses the parameterized kind types defined in STATPACK for the specifications of constants and variables used in the executables, subroutines and functions available in NCSTAT. Information about installing STATPACK and the parameterized kind types used in this library is available at [STATPACK](#).

As described above, some NCSTAT operators can also take advantage of an optimized and multi-threaded BLAS library, such as the OpenBLAS library [[openblas](#)], the ATLAS library [[atlas](#)] or vendor BLAS like Intel MKL [[mkl](#)].

Linking the object code of these nonstandard libraries with NCSTAT is usually done with the help of compiler options, which are passed directly to the UNIX linker, `ld` (see below the section [Installation](#) for further details).

1.3 Installation

1.3.1 Basic installation

In this section, we provide a step by step procedure for the installation of the NCSTAT software.

Before compiling the NCSTAT distribution, you must create the NetCDF and STATPACK library (archive) files since NCSTAT depends on NetCDF and STATPACK. Beware that default command-line flags may not be sufficient for compiling the source code of these two libraries, especially the NetCDF library. In order to be compatible with NCSTAT, you may have to use the same (or at least compatible) command-line flags for compiling these two libraries as you will use later for compiling NCSTAT. Also note that version 2.3 or above of STATPACK is required for compilation of version 2.3 of NCSTAT.

Once this is done, please follow the following steps for LINUX/Unix systems:

- 1) Download the latest NCSTAT version at [NCSTAT](#) .
For example, let us call this package **NCSTAT2.3.tar.gz**.
- 2) Put the file in your preferred directory such as `$HOME` directory or, for example, `/opt/` directory if you have `ROOT` privilege.
- 3) Execute the UNIX command:

```
$ tar -xzf NCSTAT2.3.tar.gz
```

to decompress the archive. Let us denote `<NCSTAT directory>` the package's top directory after decompression. For example, it could be `$HOME/NCSTAT2.3` or `/opt/NCSTAT2.3`.

It is not mandatory, but recommended, to set the `NCSTATDIR` Shell environment variable to the path of the NCSTAT top directory:

Table 1: Defining the Shell environment variable NCSTATDIR

Shell	Command line
csh/tcsh	setenv NCSTATDIR <NCSTAT <i>directory</i> >
sh/bash	export NCSTATDIR=<NCSTAT <i>directory</i> >

One of this command can be placed in the appropriate shell startup file in \$HOME (i.e., `.bashrc` or `.cshrc` files).

This directory, <NCSTAT *directory*>, contains the following subdirectories and associated files:

Table 2: Main NCSTAT directory

File/subdirectory	Content
makefile	Generic Makefile
make.inc	User specification options for makefile
LICENSE	NCSTAT License file
README	README file
Changelog.txt	Change log file
doc	html and pdf NCSTAT documentation
makeincs	Template <code>make.inc</code> files for various compilers/platforms
sources	NCSTAT source code

4) In order to proceed to compilation: Go to the `$NCSTATDIR` directory:

```
$ cd $NCSTATDIR
```

and edit the `make.inc` file inside this directory and follow the directions to change appropriately:

- the absolute path of the `$NCSTATDIR` directory (`TOPDIR`);
- the name/path of the Fortran95/2003 compiler and the associated compilation/loader options (`FC`, `FLAGS` and `LDFLAGS`);
- the name/path of your NetCDF, STATPACK and, eventually, BLAS libraries (`NETCDF`, `STATPACK` and `LBLAS`);

- the name/path of the directory for the NCSTAT executables (EXECDIR).

Alternatively, you can look at the `make.inc` examples in the subdirectory `$NCSTATDIR/makeincs` and if one of them matches your compiler/platform, use this file as a template `make.inc` to build your own `make.inc`.

This can be done:

- manually, by overwriting the `make.inc` file in `$NCSTATDIR` by your choice in `$NCSTATDIR/makeincs`;
- by executing the command:

```
$ make
```

in the `$NCSTATDIR` directory, selecting the name for your architecture/compiler in the list printed on the screen and, then, executing the command:

```
$ make <arch>
```

where **<arch>** is the selected name for your architecture/compiler. These steps will also overwrite the `make.inc` file in `$NCSTATDIR` by your choice in `$NCSTATDIR/makeincs`.

After these steps, you still need to customize this new `make.inc` file at least to provide:

- the name/path of your NetCDF, STATPACK and, eventually, BLAS libraries (NETCDF, STATPACK and LBLAS);
- the path of the directory for the NCSTAT executables (EXECDIR).

Two loader options are typically used for linking the object code of the NetCDF, STATPACK and, eventually, BLAS libraries with NCSTAT executables:

- **-lname** causes the compiler to look for a library file named `libname.a` and to link the NCSTAT executables to this library. To find this library file, the compiler searches sequentially through any directories named with the **-L** option explained below;
- **-Ldir** option lets you specify a private directory for libraries specified with the **-l** option, before searching in the standard library directories `/lib` and `/usr/lib`.

Note that your compiler may have other options for specifying libraries, particularly if your UNIX system supports shared libraries and you want to use shared versions of the NetCDF, STATPACK and, eventually, BLAS libraries when creating NCSTAT executables.

Remember also that UNIX linkers search for libraries in the order in which they occur on the command line and only resolve the references that are outstanding at the time when the library is searched. Therefore, the order of libraries and source/object files specified in `LDFLAGS` can be critical and it is almost always a good idea to list first the STATPACK library and, secondly, only the NetCDF and BLAS libraries (or other libraries) in the shell variable `LDFLAGS` when compiling and linking NCSTAT executables in order to avoid “Undefined” symbol messages during the loading or execution of a NCSTAT executable.

Moreover, if NCSTAT is built with OpenMP support, many NCSTAT operators will be multi-threaded and the NetCDF, STATPACK and, eventually, BLAS libraries linked to NCSTAT must be compiled thread-safe, as much as possible, in order to avoid unexpected errors at execution of the NCSTAT operators. A simple way to achieve this, is often to compile these libraries with OpenMP support. See the section [OpenMP compilation](#) for more details.

- 5) For compiling and creating the NCSTAT executables, then execute the **make** command:

```
$ make all
```

in the `$NCSTATDIR` directory.

If no errors are generated during this last step, NCSTAT is now installed successfully on your computer and the NCSTAT executables are in the directory that you have specified in the Shell variable `EXECDIR` defined in your `$NCSTATDIR/make.inc` file.

More details on the available commands for compiling and managing NCSTAT code can be found in the headers of the makefiles `$NCSTATDIR/makefile` and `$NCSTATDIR/sources/makefile`.

Note, finally, that if you want to change the precision (i.e., single, double or extended precision) of the computations performed in the NCSTAT operators, you have to recompile first your STATPACK library with the desired precision. This is because NCSTAT uses directly the standard kind type for real numbers defined in STATPACK (i.e., the parameterized `stnd` type) for all definitions/allocations of real variables and arrays in the NCSTAT code. Once the STATPACK library has been recompiled, you must then recompile and link NCSTAT with this new version of the STATPACK library as described above.

The following subsections provide more details on how to activate OpenMP support when compiling NCSTAT, and on the UNIX preprocessor **c**pp macros, which can be used to compile/optimize NCSTAT or solve some compilation problems.

1.3.2 OpenMP compilation

In order to activate OpenMP parallelism in the NCSTAT operators, all compilers require you to use an appropriate compiler flag to turn on OpenMP compilation.

The table below shows what to use for several well-known Fortran compilers:

Table 3: OpenMP compilation flags

Compiler	Compiler commands	OpenMP flag
GNU	<code>gfortran</code>	<code>-fopenmp</code>
LLVM	<code>flang</code>	<code>-fopenmp</code>
Intel	<code>ifort</code>	<code>-openmp</code> or <code>-qopenmp</code>
Intel	<code>ifx</code>	<code>-openmp</code> or <code>-qopenmp</code>
AMD AOCC	<code>flang</code>	<code>-mp</code>
PGI	<code>pgfortran</code> , <code>pgf95</code> , <code>pgf90</code>	<code>-mp</code>
NVIDIA	<code>nvfortran</code>	<code>-mp</code>
NAG	<code>nagfor</code>	<code>-openmp</code>
IBM XL	<code>xlf90_r</code> , <code>xlf95_r</code> , <code>xlf2003_r</code>	<code>-qsmp=omp</code>

Additional information on OpenMP support provided by a large range of current Fortran compilers can be found at <https://www.openmp.org/resources/openmp-compilers-tools/>. You will also find several examples of how to activate OpenMP compilation for various compilers/platforms in the template *make.inc* files under the subdirectory \$NCSTATDIR/makeincs.

How to activate parallelism when executing the NCSTAT operators compiled with OpenMP support is described below in the section *Parallel execution*.

The following NCSTAT operators are parallelized if NCSTAT has been built with OpenMP support and if the preprocessor **cpp** macro `_PARALLEL_READ` has been defined during compilation:

- `comp_clim_3d`, `comp_clim_4d`, `comp_clim_miss_3d`, `comp_clim_miss_4d`, `comp_stat_3d`, `comp_stat_4d`, `comp_stat_miss_3d`, which compute univariate statistics from a NetCDF variable;
- `comp_serie_3d`, `comp_serie_4d`, `comp_serie_miss_3d`, `comp_section_3d`, `comp_section_4d`, `comp_section_miss_3d`, `comp_integral_4d`, `comp_integral_miss_4d`, which compute time series and cross-sections from a NetCDF variable;
- `comp_fftfiler_filter_3d`, `comp_fftfiler_filter_4d`, `comp_lanczos_filter_3d`, `comp_lanczos_filter_4d`, `comp_symlin_filter_3d`, `comp_symlin_filter_4d`, which filter time series from a NetCDF variable;
- `comp_stl_3d`, `comp_stl_4d`, `comp_trend_3d`, `comp_trend_4d`, which decompose time series from a NetCDF variable;
- `comp_composite_3d`, `comp_composite_4d`, which compute composite analysis from a NetCDF variable;
- `comp_cor_1d`, `comp_cor_3d`, `comp_cor_4d`, `comp_cor_miss_3d`, `comp_reg_1d`, `comp_reg_3d`, `comp_reg_4d`, which compute correlation and regression from two NetCDF variables;
- `comp_eof_3d`, `comp_eof_4d`, `comp_eof_miss_3d`, `comp_svd_3d`, `comp_svd2_3d`, `comp_project_eof_3d`, `comp_project_eof_4d`, `comp_ortho_rot_eof_3d.F90`, `comp_ortho_rot_eof_4d.F90`, `comp_filt_rot_eof_3d.F90`, `comp_filt_rot_eof_4d.F90`, `comp_loess_rot_eof_3d.F90`, `comp_loess_rot_eof_4d.F90`, which compute multivariate statistics from one or two NetCDF variables.
- `comp_cross_spectrum_1d`, `comp_cross_spectrum_3d`, `comp_cross_spectrum2_3d`, which perform power or cross-power spectrum analysis from two NetCDF variables;
- `comp_spectvar_1d`, `comp_spectvar_3d`, `comp_spectvar_4d`, which perform a spectral analysis of variance from one NetCDF variable;

Note that some NCSTAT operators are still not parallelized, because they are I/O-bound (meaning that most of the time is spent in reading and writing the data) and shared memory parallelized writing of NetCDF files with OpenMP is not (yet) implemented in the NCSTAT source code.

1.3.3 Preprocessor **cpp** macros

The NCSTAT software uses the standard UNIX preprocessor, **cpp**, in order to allow some flexibility in the compilation of the NCSTAT operators. The **cpp** preprocessor is only used for conditional compilation of some parts of the NCSTAT source code at the user option. This is typically done by defining some UNIX preprocessor **cpp** macros (e.g., variables governing conditional compilation in the NCSTAT source files) at the compilation step of NCSTAT, usually by specifying `-Dname` as a compilation option, where `name` is a preprocessor **cpp** macro. Note that there is no space between `-D` and `name`. Each occurrence of `-D` defines a single macro and the `-D` option can appear many times on a command line. Please note that your compiler may have other options for specifying UNIX preprocessor **cpp** macros (this is for example the case of the IBM XL Fortran compiler on IBM UNIX-like systems).

The following preprocessor **cpp** macros are currently used in the NCSTAT source code and can be defined at compilation of NCSTAT software:

- `_USE_NETCDF36` lets you create 64-bit offset format files instead of NetCDF classic format files on output of the NCSTAT operators, if the NCSTAT software has been linked to the NetCDF 3.6 library or higher. When the **cpp** macro `_USE_NETCDF36` is defined at compilation and the version of your NetCDF library is higher than 3.6, many NCSTAT operators recognized the command line option **-bigfile**, which tells to these operators to produce NetCDF 64-bit offset format files instead of NetCDF classic format files. The use of the **cpp** macro `_USE_NETCDF36` is recommended as soon as the version of your NetCDF library is higher than 3.6, since this allows the processing of huge NetCDF files, commonly produced as climate model outputs.
- `_USE_NETCDF4` lets you create NetCDF-4/HDF5 classic model format files instead of NetCDF classic format files on output of the NCSTAT operators, if the NCSTAT software has been linked to the NetCDF 4 library or higher. When the **cpp** macro `_USE_NETCDF4` is defined at compilation and the version of your NetCDF library is higher than 4, many NCSTAT operators recognized the command line option **-hdf5**, which tells to these operators to produce NetCDF-4/HDF5 classic model format files instead of NetCDF classic format files. The use of the **cpp** macro `_USE_NETCDF4` is recommended as soon as the version of your NetCDF library is higher than 4, since this allows the processing of huge NetCDF files, commonly produced as climate model outputs. Note also that the **cpp** macro `_USE_NETCDF36` is also automatically defined when the **cpp** macro `_USE_NETCDF4` is defined.
- `_USE_NETCDF44` lets you create CDF5 format files instead of NetCDF classic format files on output of the NCSTAT operators, if the NCSTAT software has been linked to the NetCDF 4.4 library or higher. When the **cpp** macro `_USE_NETCDF44` is defined at compilation and the version of your NetCDF library is higher than 4.4, many NCSTAT operators also recognized the command line option **-cdf5**, which tells to these operators to produce CDF5 format files instead of NetCDF classic format files. The use of the **cpp** macro `_USE_NETCDF44` is recommended as soon as the version of your NetCDF library is higher than 4.4, since this allows the processing of huge NetCDF files, commonly produced by climate models. Note also that the **cpp** macros `_USE_NETCDF36` and `_USE_NETCDF4` are also automatically defined when the **cpp** macro `_USE_NETCDF44` is defined.
- `_USE_NAGWARE` lets you compile the NCSTAT software with the NAG Fortran Compiler. The UNIX system subroutine and function, **getarg()** and **iargc()**, which are currently used by NCSTAT operators to process their command line arguments, are normally external programs without any explicit Fortran90 interfaces. However, these two UNIX system programs are part of the *f90_unix_env* Fortran90 module when the NAG Fortran compiler is used. The **cpp** macro `_USE_NAGWARE` takes care of this difference. Don't use the **cpp** macro `_USE_NAGWARE` with other Fortran compilers since this will generate compilation errors.
- `_WHERE` replaces some **where** Fortran90 constructs by do loops in the source code when OpenMP is used. This **cpp** macro is useful for activating OpenMP with some Fortran compilers, like the INTEL ifort compiler, which have some restrictions about the Fortran90 instructions, which can be used inside an OpenMP construct.
- `_BLAS` lets you activate the use of an optimized and multi-threaded BLAS library [*blas*] inside NCSTAT as described in the section [Parallelism](#). Note that the name and path of this BLAS library must also be specified with the help of compiler/loader options on the command line as described in the section [Basic installation](#).
- `_TRANPOSE` tells to the Fortran compiler to replace each instance of the Fortran90 intrinsic function, **transpose()**, in the source code by the corresponding STATPACK function, **transpose2()**, which is multi-threaded when OpenMP is used. Use the **cpp** macro `_TRANPOSE`, if you suspect that the intrinsic Fortran90 functions of your Fortran compiler are not optimized or efficient.
- `_MATMUL` tells to the Fortran compiler to replace each instance of the Fortran90 intrinsic function, **matmul()**, in the source code by the corresponding STATPACK function, **matmul2()**, which is multi-threaded when OpenMP is used. Use the **cpp** macro `_MATMUL`, if you suspect that the intrinsic Fortran90 functions of your Fortran compiler are not optimized or efficient. If the **cpp** macro `_BLAS` is also defined, the BLAS subroutine **Xgemv()** will be used instead of an OpenMP multi-threaded version of **matmul()**.
- `_PARALLEL_READ` lets you activate parallel reading of NetCDF files based on the OpenMP standard [*openmp*]

, as described in the section [Parallelism](#), if the NCSTAT source code has been compiled with OpenMP support as described in the section [OpenMP compilation](#). If OpenMP compilation has not been activated this preprocessor **cpp** macro has no effect. Note that OpenMP parallel reading of input NetCDF files is only used for classic, 64-bit offset or CDF5 format files and is deactivated for NetCDF-4/HDF5 input files as it does not work properly for this format. Finally, if multi-threaded versions of the NCSTAT operators are not working properly on your machine, desactivating the **cpp** macro `_PARALLEL_READ` at compilation is a good choice, since you will still benefit from the parallelism of the STATPACK and BLAS libraries.

Examples of use of these preprocessor **cpp** macros for the compilation of NCSTAT can be found in the template *make.inc* files under the subdirectory `$NCSTATDIR/makeincs`.

1.4 Usage

1.4.1 Basics

This section describes how to use NCSTAT operators and gives an overview of the available functionalities.

After a successful compilation of NCSTAT, it is not mandatory, but recommended to update your executable search path with the directory containing the NCSTAT executables if this directory is not already in your executable search path. For example, to do this if your Shell is **sh** or **bash**, execute the following command (preferably in your startup file):

```
$ export PATH="$EXECDIR:$PATH"
```

where `EXECDIR` is a Shell variable containing the name/path you have specified in your `$NCSTATDIR/make.inc` file for the location of the NCSTAT executables.

Assuming that you have updated your search path as described above, all the NCSTAT operators can be directly executed at the command line and follow this syntax:

NCSTAT_operator -arg1 -arg2 ... -argN

The last three characters of each NCSTAT operator indicates implicitly, the NetCDF variables, which can be processed by this NCSTAT operator. For example, *comp_clim_3d* can process tridimensional NetCDF variables; *comp_clim_4d* can process fourdimensional NetCDF variables and so on.

The number and meaning of the arguments, **-arg1 -arg2 ... -argN**, depend on the **NCSTAT_operator**, but all the arguments begin with a - and can be specified in any order. There are two categories of arguments in the NCSTAT operators:

- the arguments, which are used to switch on/off a flag and, thus don't take any values
- the arguments, which take a value (e.g., a string, an integer, a real number). In that case, the name of the argument is followed by = and you must give the value or values immediately after, with no space between = and the value or values (several values must be separated by a comma).

In order to know quickly, the available arguments for a given NCSTAT operator, just execute this NCSTAT operator without any argument:

```
$ NCSTAT_operator
```

This will print on the screen the purpose of the NCSTAT operator, the list of available arguments for this operator and, finally, if this operator is parallelized with OpenMP (taking into account how you have compiled NCSTAT). As an illustration, if you execute:

```
$ comp_clim_3d
```

You will obtain the following informations on the screen (output may differ slightly depending on your compilation options):

Purpose :

Compute a climatology from a tridimensional variable extracted from a NetCDF dataset.

Usage :

```
comp_clim_3d -f=input_netcdf_file
              -v=netcdf_variable
              -p=periodicity           (optional)
              -x=lon1,lon2             (optional)
              -y=lat1,lat2             (optional)
              -t=time1,time2           (optional)
              -c=output_climatology_netcdf_file (optional)
              -m=output_mesh_mask_netcdf_file (optional)
              -yl=lat11,lat12          (optional)
              -mi=missing_value        (optional)
              -fmsk=input_mesh_mask_netcdf_file (optional)
              -vmsk=mesh_mask_netcdf_variable (optional)
              -val=mask_value           (optional)
              -rel=mask_relation        (optional : eq, gt, ge, lt, le)
              -ntr=number_of_time_records (optional)
              -double                   (optional)
              -bigfile                  (optional)
              -hdf5                    (optional)
              -cdf5                    (optional)
              -tlimited                  (optional)
```

By default :

```
-p= the periodicity is set to 1
-x= the whole longitude domain associated with the netcdf_variable
-y= the whole latitude domain associated with the netcdf_variable
-t= the whole time period associated with the netcdf_variable
-c=clim_netcdf_variable.nc
-m= the mesh-mask NetCDF file is not created
-yl= it is assumed that the domain is the whole globe or that the grid is regular
-mi= the missing_value for the STD variable is equal to 1.e+20
-fmsk= an input_mesh_mask_netcdf_file is not used
-vmsk= an input mesh_mask_netcdf_variable is not used
-val=1.
-rel=eq
-ntr= the number_of_time_records is equal to the periodicity
-double : the STD variable is stored as single floating numbers
-bigfile : a NetCDF classical format file is created
-hdf5 : a NetCDF classical format file is created
-cdf5 : a NetCDF classical format file is created
-tlimited : the time dimension is defined as unlimited
```

This procedure is parallelized with OpenMP

As you can see, some arguments are mandatory and other are optional. The mandatory arguments always take a value, but optional arguments can be of both types. Finally, you know if this operator supports OpenMP parallelism or not.

1.4.2 Common arguments

The following arguments are available for almost all the NCSTAT operators and have the same meaning across the operators.

The common arguments with values of the NCSTAT operators are:

Table 4: Common arguments with values of the NCSTAT operators

Argument	Meaning or use
-f=	specify an input NetCDF file
-v=	specify an input NetCDF variable
-c=	specify an input NetCDF climatology file produced by <i>comp_clim_3d</i> or similar operators
-x=	specify a longitude domain associated with an input NetCDF variable
-y=	specify a latitude domain associated with an input NetCDF variable
-z=	specify a vertical extension associated with an input NetCDF variable
-t=	specify a time interval associated with an input NetCDF variable
-p=	specify the periodicity of the input data
-m=	specify a mesh-mask NetCDF file produced by <i>comp_clim_3d</i> , <i>comp_mask_3d</i> or similar operators
-o=	specify an output NetCDF file
-mi=	specify a missing indicator value/attribute in the data

Note that for the -x=, -y= and -t= arguments, the latitude/longitude domains and the time interval are specified as integer indices and that the coordinate NetCDF variables, if they exist, are not used, even if they are always copied from the input NetCDF file in the output NetCDF files.

You can use the operators *comp_mask_3d* and *comp_mask_4d* to transform geographical coordinates as integer indices for use with the NCSTAT operators. Alternatively, you can select the region of interest for your analysis with a NCSTAT operator by specifying a (land-sea) mask with the -m= argument on input of the NCSTAT operator and the appropriate mask can also be constructed with the help of the *comp_mask_3d* and *comp_mask_4d* operators.

The common arguments without value of the NCSTAT operators are:

Table 5: Common “switch” arguments of the NCSTAT operators

Argument	Meaning or use
-double	specify that the results of a NCSTAT operator must be stored as double floating-point numbers in the output NetCDF files
-tlimited	specify that the time dimension must be defined as limited in the output NetCDF files
-bigfile	specify that the output files must be 64-bit offset format NetCDF files
-hdf5	specify that the output files must be NetCDF-4/HDF5 classical model format NetCDF files
-cdf5	specify that the output files must be CDF5 format NetCDF files

1.4.3 Parallel execution

Users may request a specific number of OpenMP threads to distribute the work done by the NCSTAT operators, when OpenMP support has been activated at compilation of NCSTAT. As a general rule, don’t request more OpenMP threads than the number of processors available on your machine (excluding also processors used for hyperthreading), this will result in large loss of performance. Keep also in mind that the efficiency of shared memory parallelism as implemented in NCSTAT with OpenMP also depends heavily on the workload of your shared memory computer at runtime.

More generally, threading performance of the NCSTAT operators will depend on a variety of factors including the compiler, the version of the OpenMP library, the processor type, the number of cores, the amount of available memory, whether hyperthreading is enabled and the mix of applications that are executing concurrently with the NCSTAT operator.

At the simplest level, the number of OpenMP threads used by the NCSTAT operators can be controlled by setting the `OMP_NUM_THREADS` OpenMP environment variable to the desired number of threads and the number of threads will be the same throughout the execution of the commands. The `OMP_NUM_THREADS` OpenMP environment variable must be defined before the use of the NCSTAT operators to activate OpenMP parallelism.

Setting OpenMP environment variables is done the same way you set any other environment variables, and depends upon which Shell you use:

Table 6: Setting the number of OpenMP threads to be used

Shell	Command line
csh/tcsh	<code>setenv OMP_NUM_THREADS 8</code>
sh/bash	<code>export OMP_NUM_THREADS=8</code>

In some cases, an OpenMP application will perform better if its OpenMP threads are bound to processors/cores (this

is called “thread affinity”, “thread binding” or “processor affinity”) because this can result in better cache utilization, thereby reducing costly memory accesses. OpenMP version 3.1 API provides an environment variable to turn processor binding “on” or “off”. For example, to turn “on” thread binding you can use:

```
$ export OMP_PROC_BIND=TRUE    #if you are using a sh/bash Shell
```

Keep also in mind, that the OpenMP standard does not specify how much stack space an OpenMP thread should have (this stack space is used to store private variables and arrays used only by each OpenMP thread). Consequently, OpenMP implementations will differ in the default thread stack size and the default thread stack size can be easily exhausted for moderate/large applications on some systems. Threads that exceed their stack allocation may give a segmentation fault or the application may continue to run while data is being corrupted. If your OpenMP environment supports the OpenMP 3.0 `OMP_STACKSIZE` environment variable, you can use it to set the thread stack size prior to program execution. For example:

```
$ export OMP_STACKSIZE=10M    #if you are using a sh/bash Shell
$ export OMP_STACKSIZE=3000k  #if you are using a sh/bash Shell
```

More generally, the run-time behaviour of NCSTAT operators is also determined by setting some other OpenMP environment variables (e.g., `OMP_NESTED`, `OMP_MAX_ACTIVE_LEVELS` or `OMP_DYNAMIC` for example) just before the execution of the operators. However, starting with the OpenMP 5.0 API, the use of the `OMP_NESTED` variable is deprecated and must be replaced by the use of the `OMP_MAX_ACTIVE_LEVELS` OpenMP variable, already available in the OpenMP 3.0 API. See [OpenMP Environment Variables](#) for more details on the use of the `OMP_MAX_ACTIVE_LEVELS` variable to turn off/on nested OpenMP parallelism and the level of nested parallelism in Fortran or C programs using OpenMP.

See also the official OpenMP documentation available at [OpenMP](#) or the more friendly tutorial [OpenMP tutorial](#) for more details and examples.

Note, in particular, that the STATPACK subroutines and functions used in the NCSTAT code may use OpenMP nested parallelism if the `OMP_NESTED` variable is set to `TRUE` or if the `OMP_MAX_ACTIVE_LEVELS` is set to 1, but that the usage of OpenMP nested parallelism is not recommended if you have compiled the NCSTAT operators or the STATPACK library with BLAS support and you have linked with a multi-threaded version of BLAS, such as [\[gotoblas\]](#), [\[openblas\]](#) or vendor BLAS like Intel MKL [\[mkl\]](#). In such cases, it is better to first deactivate OpenMP nested parallelism before executing the NCSTAT operators by using first the command:

```
$ export OMP_NESTED=FALSE    #if you are using a sh/bash Shell
```

or the command (for new versions of the OpenMP library):

```
$ export OMP_MAX_ACTIVE_LEVELS=1    #if you are using a sh/bash Shell
```

and also to let OpenMP controls the multi-threading in the BLAS library, if possible.

In the case of OpenBLAS [\[openblas\]](#) or GotoBLAS [\[gotoblas\]](#), this can be done by using the makefile `USE_OPENMP=1` option when compiling OpenBLAS or GotoBLAS. Consult the OpenBLAS manual for more details [\[openblas\]](#). On the other hand, if your OpenBLAS or GotoBLAS library has already been compiled with multi-threading enabled, but no support for OpenMP (this is the default setting), it is better to deactivate OpenBLAS or GotoBLAS multi-threading before execution of your application because, otherwise, OpenMP will not control the multi-threading in the BLAS library and this will likely results in significant loss of performance. To do this, use a command like (for OpenBLAS):

```
$ export OPENBLAS_NUM_THREADS=1    #if you are using a sh/bash Shell
```

or (for GotoBLAS):

```
$ export GOTO_NUM_THREADS=1    #if you are using a sh/bash Shell
```

Similarly, for Intel MKL BLAS [mkl], it is better to let OpenMP controls the multi-threading in the MKL BLAS. This can be done simply by undefining the Shell variable MKL_NUM_THREADS:

```
$ unset MKL_NUM_THREADS    #if you are using sh/bash Shell
```

Finally, if you suspect an error in the computations performed by a parallelized NCSTAT operator on your machine, a good strategy is to compare the results of a serial execution of this operator (i.e., by setting OMP_NUM_THREADS to 1) with those of a parallel execution of this operator (i.e., by setting OMP_NUM_THREADS to a value greater than 1). If the results differ, a second step is to recompile NCSTAT without the preprocessor **cpp** macro `_PARALLEL_READ` in order to detect if the origin of the problem is due to the OpenMP parallel reading of the NetCDF files and to some incompatibilities between the compiler options used to build the NetCDF library and NCSTAT.

1.4.4 NCSTAT summary

This section re-organizes the NCSTAT operators by task, with a brief note indicating what each operator does. More details about each operator can be found in Chapter 2, where the full documentation for each NCSTAT operator is given in alphabetical order.

NCSTAT operators for compression/decompression of a NetCDF variable:

Table 7: NCSTAT operators for compression/decompression

Operator	OpenMP	Description
<i>pack_masked_data_3d</i>	no	pack a tridimensional NetCDF variable
<i>unpack_masked_data_3d</i>	no	unpack a tridimensional NetCDF variable

NCSTAT operators for computing mesh-mask files:

Table 8: NCSTAT operators for computing mesh-mask files

Operator	OpenMP	Description
<i>comp_mask_3d</i>	no	compute a mesh-mask from a tridimensional NetCDF variable
<i>comp_mask_4d</i>	no	compute a mesh-mask from a fourdimensional NetCDF variable

NCSTAT operators for univariate statistics and computation of mesh-mask files:

Table 9: NCSTAT operators for univariate statistics

Operator	OpenMP	Description
<i>comp_clim_3d</i>	yes	compute means and standard-deviations from a tridimensional NetCDF variable
<i>comp_clim_4d</i>	yes	compute means and standard-deviations from a fourdimensional NetCDF variable
<i>comp_clim_miss_3d</i>	yes	compute means and standard-deviations from a tridimensional NetCDF variable with missing values
<i>comp_clim_miss_4d</i>	yes	compute means and standard-deviations from a fourdimensional NetCDF variable with missing values
<i>comp_stat_3d</i>	yes	compute univariate statistics from a tridimensional NetCDF variable
<i>comp_stat_4d</i>	yes	compute univariate statistics from a fourdimensional NetCDF variable
<i>comp_stat_miss_3d</i>	yes	compute univariate statistics from a tridimensional NetCDF variable with missing values

NCSTAT operators for composite analysis:

Table 10: NCSTAT operators for composite analysis

Operator	OpenMP	Description
<i>comp_composite_3d</i>	yes	compute a composite analysis from a tridimensional NetCDF variable
<i>comp_composite_4d</i>	yes	compute a composite analysis from a fourdimensional NetCDF variable
<i>comp_composite_miss_3d</i>	yes	compute a composite analysis from a tridimensional NetCDF variable with missing values

NCSTAT operators for transforming and time averaging of time series:

Table 11: NCSTAT operators for computing, transforming and time averaging of time series

Operator	OpenMP	Description
<i>comp_norm_3d</i>	no	transform multi-channel time series from a tridimensional NetCDF variable
<i>comp_norm_4d</i>	no	transform multi-channel time series from a fourdimensional NetCDF variable
<i>comp_norm_miss_3d</i>	no	transform multi-channel time series from a tridimensional NetCDF variable with missing values
<i>comp_norm_miss_4d</i>	no	transform multi-channel time series from a fourdimensional NetCDF variable with missing values
<i>comp_season_3d</i>	no	compute time-averages of multi-channel time series from a tridimensional NetCDF variable
<i>comp_season_4d</i>	no	compute time-averages of multi-channel time series from a fourdimensional NetCDF variable
<i>comp_season_miss_3d</i>	no	compute time-averages of multi-channel time series from a tridimensional NetCDF variable with missing values
<i>comp_season_miss_4d</i>	no	compute time-averages of multi-channel time series from a fourdimensional NetCDF variable with missing values

NCSTAT operators for computing time series, cross-sections and vertical integrals:

Table 12: NCSTAT operators for computing time series, cross-sections and vertical integrals

Operator	OpenMP	Description
<i>comp_serie_3d</i>	yes	compute a time series from a tridimensional NetCDF variable
<i>comp_serie_4d</i>	yes	compute a time series from a fourdimensional NetCDF variable
<i>comp_serie_miss_3d</i>	yes	compute a time series from a tridimensional NetCDF variable with missing values
<i>comp_index_1d</i>	yes	compute a index time series from two unidimensional NetCDF variables
<i>comp_section_3d</i>	yes	compute a cross-section from a tridimensional NetCDF variable
<i>comp_section_4d</i>	yes	compute a cross-section from a fourdimensional NetCDF variable
<i>comp_section_miss_3d</i>	yes	compute a cross-section from a tridimensional NetCDF variable with missing values
<i>comp_integral_4d</i>	yes	compute vertical integrals from a fourdimensional NetCDF variable
<i>comp_integral_miss_4d</i>	yes	compute vertical integrals from a fourdimensional NetCDF variable with missing values

NCSTAT operators for decomposing time series:

Table 13: NCSTAT operators for decomposing time series

Operator	OpenMP	Description
<i>comp_stl_1d</i>	no	decompose a time series from a unidimensional NetCDF variable by the STL method
<i>comp_stl_3d</i>	yes	decompose multi-channel time series from a tridimensional NetCDF variable by the STL method
<i>comp_stl_4d</i>	yes	decompose multi-channel time series from a fourdimensional NetCDF variable by the STL method
<i>comp_trend_1d</i>	no	estimate a trend from a unidimensional NetCDF variable by the LOESS method
<i>comp_trend_3d</i>	yes	estimate multi-channel trend from a tridimensional NetCDF variable by the LOESS method
<i>comp_trend_4d</i>	yes	estimate multi-channel trend from a fourdimensional NetCDF variable by the LOESS method

NCSTAT operators for filtering time series:

Table 14: NCSTAT operators for filtering time series

Operator	OpenMP	Description
<i>comp_fftfilter_1d</i>	no	filter a time series from a unidimensional NetCDF variable in a selected frequency band by windowed filtering
<i>comp_fftfilter_3d</i>	yes	filter multi-channel time series from a tridimensional NetCDF variable in a selected frequency band by windowed filtering
<i>comp_fftfilter_4d</i>	yes	filter multi-channel time series from a four-dimensional NetCDF variable in a selected frequency band by windowed filtering
<i>comp_lanczos_filter_1d</i>	no	filter a time series from a unidimensional NetCDF variable in a selected frequency band by Lanczos filtering
<i>comp_lanczos_filter_3d</i>	yes	filter multi-channel time series from a tridimensional NetCDF variable in a selected frequency band by Lanczos filtering
<i>comp_lanczos_filter_4d</i>	yes	filter multi-channel time series from a four-dimensional NetCDF variable in a selected frequency band by Lanczos filtering
<i>comp_symlin_filter_1d</i>	no	filter a time series from a unidimensional NetCDF variable in a selected frequency band by linear symmetric filtering
<i>comp_symlin_filter_3d</i>	yes	filter multi-channel time series from a tridimensional NetCDF variable in a selected frequency band by linear symmetric filtering
<i>comp_symlin_filter_4d</i>	yes	filter multi-channel time series from a four-dimensional NetCDF variable in a selected frequency band by linear symmetric filtering
<i>comp_freq_func_1d</i>	no	estimate the transfer function of a Lanczos or linear symmetric filter

NCSTAT operators for correlation and regression analysis:

Table 15: NCSTAT operators for correlation and regression analysis

Operator	OpenMP	Description
<i>comp_cor_1d</i>	yes	compute correlation and regression from an index time series and a unidimensional NetCDF variable
<i>comp_cor_3d</i>	yes	compute correlation and regression from an index time series and a tridimensional NetCDF variable
<i>comp_cor_4d</i>	yes	compute correlation and regression from an index time series and a fourdimensional NetCDF variable
<i>comp_cor_miss_3d</i>	yes	compute correlation and regression from an index time series and a tridimensional NetCDF variable with missing values
<i>comp_reg_1d</i>	no	compute trend and regression from an index time series and a unidimensional NetCDF variable
<i>comp_reg_3d</i>	yes	compute trend and regression from an index time series and a tridimensional NetCDF variable
<i>comp_reg_4d</i>	yes	compute trend and regression from an index time series and a fourdimensional NetCDF variable

NCSTAT operators for multivariate statistics:

Table 16: NCSTAT operators for multivariate statistics

Operator	OpenMP	Description
<i>comp_eof_3d</i>	yes	compute a Principal Component Analysis (PCA) from a tridimensional NetCDF variable
<i>comp_eof_4d</i>	yes	compute a Principal Component Analysis (PCA) from a fourdimensional NetCDF variable
<i>comp_eof_miss_3d</i>	yes	compute a Principal Component Analysis (PCA) from a tridimensional NetCDF variable with missing values
<i>comp_svd_3d</i>	yes	compute a Maximum Covariance Analysis (MCA) from two tri- or fourdimensional NetCDF variables
<i>comp_svd2_3d</i>	yes	compute a Maximum Covariance Analysis (MCA) from two tri- or fourdimensional NetCDF variables
<i>comp_invert_eof_3d</i>	no	compute a PCA or MCA approximation of a tridimensional NetCDF variable
<i>comp_invert_eof_4d</i>	no	compute a PCA or MCA approximation of a fourdimensional NetCDF variable
<i>comp_project_eof_3d</i>	yes	compute a PCA or MCA projection of a tridimensional NetCDF variable
<i>comp_project_eof_4d</i>	yes	compute a PCA or MCA projection of a fourdimensional NetCDF variable

NCSTAT operators for specialized multivariate statistics:

Table 17: NCSTAT operators for orthogonal rotation of a partial PCA model

Operator	OpenMP	Description
<i>comp_ortho_rot_eof_3d</i>	yes	perform an orthogonal rotation of a (partial) PCA model computed from a tridimensional NetCDF variable
<i>comp_ortho_rot_eof_4d</i>	yes	perform an orthogonal rotation of a (partial) PCA model computed from a fourdimensional NetCDF variable
<i>comp_filt_rot_eof_3d</i>	yes	perform an orthogonal rotation of selected standardized Principal Components (PC) time series from a tridimensional NetCDF variable
<i>comp_filt_rot_eof_4d</i>	yes	perform an orthogonal rotation of selected standardized Principal Components (PC) time series from a fourdimensional NetCDF variable
<i>comp_loess_rot_eof_3d</i>	yes	perform an orthogonal rotation of selected standardized Principal Components (PC) time series from a tridimensional NetCDF variable
<i>comp_loess_rot_eof_4d</i>	yes	perform an orthogonal rotation of selected standardized Principal Components (PC) time series from a fourdimensional NetCDF variable

NCSTAT operators for power and cross-power spectrum analysis:

Operator	OpenMP	Description
<i>comp_spectrum_1d</i>	no	compute a power spectrum analysis from a unidimensional NetCDF variable
<i>comp_cross_spectrum_1d</i>	no	compute a power or cross-power spectrum analysis from two unidimensional NetCDF variables
<i>comp_cross_spectrum_3d</i>	yes	compute a power or cross-power spectrum analysis from an index time series and a tridimensional NetCDF variable
<i>comp_cross_spectrum2_3d</i>	yes	compute a power or cross-power spectrum analysis from an index time series and a tridimensional NetCDF variable
<i>comp_spectvar_1d</i>	no	compute variance estimates in a selected frequency band from a unidimensional NetCDF variable
<i>comp_spectvar_3d</i>	yes	compute variance estimates in a selected frequency band from a tridimensional NetCDF variable
<i>comp_spectvar_4d</i>	yes	compute variance estimates in a selected frequency band from a fourdimensional NetCDF variable
<i>comp_spectrum_ratio_1d</i>	no	compute power spectrum density ratios and their confidence intervals from power spectrum analyses of two unidimensional NetCDF variables
<i>comp_spectrum_ratio_3d</i>	no	compute power spectrum density ratios and their confidence intervals from power spectrum analyses of two tridimensional NetCDF variables
<i>comp_spectrum_diff_3d</i>	no	perform nonparametric statistical tests about the difference and shape of two power spectra of two tridimensional NetCDF variables

REFERENCE MANUAL

2.1 comp_clim_3d

2.1.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.1.2 Latest revision

29/04/2026

2.1.3 Purpose

Compute a climatology (e.g., means and standard-deviations) from a tridimensional variable extracted from a NetCDF dataset and, optionally, the mask and scale factors of the 2-D grid-mesh associated with the input NetCDF variable.

Mean and standard-deviation are computed for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable. These means and standard-deviations may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

The mean is a simple, but informative, measure, of the central tendency of a variable [vonStorch_Zwiers]. The standard-deviation is a conventional measure of variation of a variable [vonStorch_Zwiers]. If $X(:)$ is a vector of $ntime$ observations for one grid-point in the time series of the 2-D grid-mesh, the mean and standard-deviation statistics for this grid-point are estimated, respectively, by

- **MEAN** = $\text{sum}(X(:)) / ntime$
- **STD** = $\sqrt{\text{sum}([X(:) - \text{MEAN}]^2) / ntime}$

Note that the divisor used in calculating standard-deviation is the number of observations, this is in contrast with the formulae used in [comp_stat_3d](#), which uses the number of degrees of freedom.

If your data contains missing values, use [comp_clim_miss_3d](#) instead of [comp_clim_3d](#) to estimate means and standard deviations from your gappy dataset.

If you need more univariate statistics on your input variable such as skewness, kurtosis, etc., refer to [comp_stat_3d](#). Finally, if the NetCDF variable is fourdimensional use [comp_clim_4d](#) instead of [comp_clim_3d](#).

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the means and standard-deviations with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence mask and scale factor variables, which may be used to compute the surface of each cell in the 2-D grid-mesh associated

with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures such as *comp_serie_3d*, *comp_eof_3d*, etc. When computing the scale factors, it is assumed that the 2-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

2.1.4 Further Details

Usage

```
$ comp_clim_3d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-p=periodicity                (optional) \  
-x=lon1,lon2                  (optional) \  
-y=lat1,lat2                  (optional) \  
-t=time1,time2                (optional) \  
-c=output_climatology_netcdf_file (optional) \  
-m=output_mesh_mask_netcdf_file  (optional) \  
-yl=lat11,lat12               (optional) \  
-mi=missing_value             (optional) \  
-fmsk=input_mesh_mask_netcdf_file (optional) \  
-vmsk=mesh_mask_netcdf_variable (optional) \  
-val=mask_value               (optional) \  
-rel=mask_relation            (optional : eq, gt, ge, lt, le) \  
-ntr=number_of_time_records    (optional) \  
-double                        (optional) \  
-bigfile                       (optional) \  
-hdf5                          (optional) \  
-cdf5                          (optional) \  
-tlimited                       (optional)
```

By default

- p=** the *periodicity* is equal to 1
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- c=** the *output_climatology_netcdf_file* is named `clim_netcdf_variable.nc`
- m=** the *output_mesh_mask_netcdf_file* is not created
- yl=** it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors
- mi=** the *missing_value* for the STD variable in the output NetCDF file is set to `1.e+20`
- fmsk=** an *input_mesh_mask_netcdf_file* is not used when computing the presence-absence mask
- vmsk=** a *mesh_mask_netcdf_variable* is not used when computing the presence-absence mask
- val=** this argument is set to 1 when computing the presence-absence mask
- rel=** this argument is set to `eq` when computing the presence-absence mask

- ntr=** the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity*
- double** the standard-deviations are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the standard-deviations are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.

- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.

- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to construct the climatology.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_clim_3d*.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to construct the climatology and to compute the statistics.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

- 5) It is assumed that the input data has no missing values (excepted missing values associated with a constant land-sea mask as indicated by a *missing_value* or *_FillValue* attribute).

If it is the case, use [comp_clim_miss_3d](#) instead of *comp_clim_3d*.

- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 2-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 2-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 7) If the **-m=output_mesh_mask_netcdf_file** argument is present, and if the **-fmsk=** and **-vmsk=** arguments are also specified, the presence-absence mask in the *output_mesh_mask_netcdf_file* is computed from the input *mesh_mask_netcdf_variable* (as specified by the **-vmsk=** and **-fmsk=** arguments) as follows :

- **output_mask**(i,j) = 1 if **input_mask**(i,j) .*mask_relation*. *mask_value* is true
- **output_mask**(i,j) = 0 otherwise

where *mask_relation* is determined from the **-rel=** argument and *mask_value* from the **-val=** argument (*mask_value* is a real number).

By default, *mask_relation* is eq and *mask_value* is 1. . Both the **-fmsk=** and **-vmsk=** arguments must be present, otherwise the procedure will stop with an error message.

- 8) If the **-m=output_mesh_mask_netcdf_file** argument is present and some scale factors can not be computed, these scale factors are set to 1.
- 9) The **-mi=missing_value** argument specifies the missing value indicator for the standard-deviation (STD) variable in the *output_climatology_netcdf_file*.

If the **-mi=** argument is not specified and the NetCDF variable has a *missing_value* or *_FillValue* attribute, the *missing_value* is set to 1.e+20. This argument is not used if the NetCDF variable specified in the **-v=** argument has no *missing_value* or *_FillValue* attribute.

- 10) The **-ntr=number_of_time_records** argument specifies how many time records are read in each iteration of the loop for reading the input NetCDF variable.

By default, the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity* (as specified by the **-p=** argument). On very large dataset, it may be useful to reduce the *number_of_time_records* in order to decrease the memory used by the procedure.

- 11) The **-double** argument specifies that the standard-deviation variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_climatology_netcdf_file*.
- 12) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 13) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 14) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 15) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 16) For more details on the use of univariate statistics in the climate literature, see

- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: 9780521012300

Outputs

`comp_clim_3d` creates an output NetCDF file that contains the means, standard-deviations and number of observations of the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_mean(periodicity, nlat, nlon)` : the means for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_std(periodicity, nlat, nlon)` : the standard-deviations for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_nobs(periodicity)` : the number of observations used to compute the statistics.

The means and standard-deviations are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values.

Optionally, `comp_clim_3d` can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) `netcdf_variable_nmask(nlat, nlon)` : a presence-absence or land-sea 2-D mask associated with the input NetCDF variable.
- 2) `netcdf_variable_e1n(nlat, nlon)` : the first scale factor associated with the 2-D grid-mesh of the input NetCDF variable.
- 3) `netcdf_variable_e2n(nlat, nlon)` : the second scale factor associated with the 2-D grid-mesh of the input NetCDF variable.

Multiplying the two scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly means and standard-deviations from the NetCDF file `ST7_1m_00101_20012_grid_T_sosstsst.nc`, which includes a NetCDF variable `sosstsst` and store the results in the NetCDF file `clim_ST7_1m_00101_20012_grid_T_sosstsst.nc`, use the following command :

```
$ comp_clim_3d \
-f=ST7_1m_00101_20012_grid_T_sosstsst.nc \
-v=sosstsst \
-p=12 \
-c=clim_ST7_1m_00101_20012_grid_T_sosstsst.nc
```

- 2) For computing monthly means and standard-deviations from the NetCDF file `sst.mnmean.nc`, which includes a NetCDF variable `sst` and store the results in a NetCDF file named `clim_sst.nc` and, in addition, generate an associated *mesh_mask_netcdf_file* named `mesh_mask_sst.nc`, use the following command :

```
$ comp_clim_3d \
-f=sst.mnmean.nc \
-v=sst \
-p=12 \
-m=mesh_mask_sst.nc
```

2.2 comp_clim_4d

2.2.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.2.2 Latest revision

29/04/2026

2.2.3 Purpose

Compute a climatology (e.g., means and standard-deviations) from a fourdimensional variable extracted from a NetCDF dataset and, optionally, the associated mask and scale factors of the 3-D grid-mesh associated with the input NetCDF variable.

Means and standard-deviations are computed for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable. These means and standard-deviations may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

The mean is a simple, but informative, measure, of the central tendency of a variable. The standard-deviation is a conventional measure of variation of a variable. If $X(:)$ is a vector of *ntime* observations for one grid-point in the time series of the 3-D grid-mesh, the mean and standard-deviation statistics for this grid-point are estimated, respectively, by

- $\text{MEAN} = \text{sum}(X(:)) / \text{ntime}$
- $\text{STD} = \text{sqrt}(\text{sum}([X(:)-\text{MEAN}]^*2) / \text{ntime})$

Note that the divisor used in calculating standard-deviation is the number of observations, this is in contrast with the formulae used in [comp_stat_4d](#), which uses the number of degrees of freedom.

If your data contains missing values use [comp_clim_miss_4d](#) instead of `comp_clim_4d` to estimate means and standard deviations from your gappy dataset.

If the NetCDF variable is tridimensional use [comp_clim_3d](#) instead of `comp_clim_4d`. If you need more univariate statistics on your input variable such as skewness, kurtosis, etc., refer to [comp_stat_4d](#).

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the means and standard-deviations with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence mask and scale factor variables, which may be used to compute the surface or volume of each cell in the 3-D grid-mesh associated with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures such as [comp_serie_4d](#), [comp_eof_4d](#), etc. When computing the scale factors, it is assumed that the 3-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

2.2.4 Further Details

Usage

```
$ comp_clim_4d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -p=periodicity (optional) \
  -x=lon1,lon2 (optional) \
  -y=lat1,lat2 (optional) \
  -z=level1,level2 (optional) \
  -t=time1,time2 (optional) \
  -c=output_climatology_netcdf_file (optional) \
  -m=output_mesh_mask_netcdf_file (optional) \
  -vz=name_of_the_vertical_netcdf_variable (optional) \
  -z0=value_of_the_highest_level (optional) \
  -sf=method_for_computing_the_third_scale_factor (optional : method1, method2) \
  -yl=lat11,lat12 (optional) \
  -mi=missing_value (optional) \
  -fmsk=input_mesh_mask_netcdf_file (optional) \
  -vmsk=mesh_mask_netcdf_variable (optional) \
  -val=mask_value (optional) \
  -rel=mask_relation (optional : eq, gt, ge, lt, le) \
  -ntr=number_of_time_records (optional) \
  -double (optional) \
  -bigfile (optional) \
  -hdf5 (optional) \
  -cdf5 (optional) \
  -tlimited (optional)
```

By default

- p=** the *periodicity* is equal to 1
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- c=** the *output_climatology_netcdf_file* is named `clim_netcdf_variable.nc`
- m=** the *output_mesh_mask_netcdf_file* is not created
- vz=** the variable with the same name as the third dimension, if any (e.g., the associated coordinate variable)
- z0=** a value of 0 is assumed for the highest level when computing the third scale factor
- sf=** `method2` (e.g., it is assumed that each level or depth is located in the middle of each layer and the third scale factor is computed accordingly)
- yl=** it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors
- mi=** the *missing_value* for the STD variable in the output NetCDF file is set to `1.e+20`
- fmsk=** an *input_mesh_mask_netcdf_file* is not used when computing the presence-absence mask
- vmsk=** a *mesh_mask_netcdf_variable* is not used when computing the presence-absence mask

- val=** this argument is set to 1 when computing the presence-absence mask
- rel=** this argument is set to `eq` when computing the presence-absence mask
- ntr=** the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity*
- double** the standard-deviations are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the standard-deviations are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.
- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to construct the climatology.

The longitude, latitude or depth range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using `comp_clim_4d`.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to construct the climatology.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

- 5) It is assumed that the data has no missing values (excepted missing values associated with a constant land-sea mask as indicated by a *missing_value* or *_FillValue* attribute).

If it is the case, use [comp_clim_miss_4d](#) instead of `comp_clim_4d`

- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 3-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 3-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 7) If the **-m=output_mesh_mask_netcdf_file** argument is present, the third scale factor is computed with the help of the vertical coordinate variable (or the NetCDF variable specified with **-vz=** argument) if this vertical coordinate variable is strictly monotonic.
- 8) If the **-m=output_mesh_mask_netcdf_file** argument is present and if some scale factors can not be computed, these scale factors are set to 1.
- 9) The **-z0=value_of_the_highest_level** argument specifies a value for the highest level or depth in order to compute the first/last element of the third scale factor. The default value is 0.
- 10) The **-sf=method_for_computing_the_third_scale_factor** argument allows to specify the method for computing the third scale factor, if a mesh-mask NetCDF file is created:
 - **-sf=method1** : the third scale factor is computed as the differences between successive levels (or depths)
 - **-sf=method2** : the third scale factor is computed by assuming that each level or depth is located at the middle of the corresponding layer.
- 11) If the **-m=output_mesh_mask_netcdf_file** argument is present, and if the **-fmsk=** and **-vmsk=** arguments are also specified, the presence-absence mask in the *output_mesh_mask_netcdf_file* is computed from the input *mesh_mask_netcdf_variable* (as specified by the **-vmsk=** and **-fmsk=** arguments) as follows :
 - **output_mask(i,j,k)** = 1 if **input_mask(i,j,k)** .*mask_relation* . *mask_value* is true
 - **output_mask(i,j,k)** = 0 otherwise

where *mask_relation* is determine from the **-rel=** argument and *mask_value* from the **-val=** argument (*mask_value* is a real number).

By default, *mask_relation* is eq and *mask_value* is 1. . Both the **-fmsk=** and **-vmsk=** arguments must be present, otherwise the procedure will stop with an error message.
- 12) The **-mi=missing_value** argument specifies the missing value indicator for the standard-deviation (STD) variable in the *output_climatology_netcdf_file*.

If the **-mi=** argument is not specified and the NetCDF variable has a *missing_value* or *_FillValue* attribute, the *missing_value* is set to 1.e+20. This argument is not used if the NetCDF variable specified in the **-v=** argument has no *missing_value* or *_FillValue* attribute.
- 13) The **-ntr=number_of_time_records** argument specifies how many time records are read in each iteration of the loop for reading the input NetCDF variable.

By default, the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity* (as specified by the **-p=** argument). On very large dataset, it may be useful to reduce the *number_of_time_records* in order to decrease the memory used by the procedure.
- 14) The **-double** argument specifies that the standard-deviation (STD) variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_climatology_netcdf_file*.
- 15) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 16) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The `-cdf5` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 18) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 19) For more details on the use of univariate statistics in the climate literature, see
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](#)

Outputs

`comp_clim_4d` creates an output NetCDF file that contains the means, standard-deviations and number of observations of the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the vertical and spatial dimensions of the input NetCDF variable) :

- 1) *netcdf_variable_mean*(`periodicity`, `nlev`, `nlat`, `nlon`) : the means for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 2) *netcdf_variable_std*(`periodicity`, `nlev`, `nlat`, `nlon`) : the standard-deviations for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 3) *netcdf_variable_nobs*(`periodicity`) : the number of observations used to compute the statistics.

The means and standard-deviations are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values.

Optionally, `comp_clim_4d` can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) *netcdf_variable_nmask*(`nlev`, `nlat`, `nlon`) : a presence-absence or height-land-sea 3-D mask associated with the input NetCDF variable.
- 2) *netcdf_variable_e1n*(`nlat`, `nlon`) : the first scale factor associated with the 3-D grid-mesh of the input NetCDF variable.
- 3) *netcdf_variable_e2n*(`nlat`, `nlon`) : the second scale factor associated with the 3-D grid-mesh of the input NetCDF variable.
- 4) *netcdf_variable_e3n*(`nlev`, 1, 1) : the third scale factor associated with the 3-D grid-mesh of the input NetCDF variable.

Multiplying the two first scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable. Multiplying the three scale factors together gives the volume (or a quantity proportional to the weight if the unit of the vertical coordinate variable is in hPa) of each parcel in the 3-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly means and standard-deviations from the NetCDF file `ST7_1m_00101_20012_grid_T_votemper.nc`, which includes a NetCDF variable `votemper` and store the results in the file `clim_ST7_1m_00101_20012_grid_T_votemper.nc`, use the following command :

```
$ comp_clim_4d \
-f=ST7_1m_00101_20012_grid_T_votemper.nc \
-v=votemper \
-p=12 \
-c=clim_ST7_1m_00101_20012_grid_T_votemper.nc
```

- 2) For computing monthly means and standard-deviations from the NetCDF file `vwnd.mon.mean.nc`, which includes a NetCDF variable `vwnd` and store the results in a NetCDF file named `clim_vwnd.nc` and, in addition, generate an associated mesh_mask_NetCDF_file named `mesh_mask_wind_ncep2.nc`, use the following command :

```
$ comp_clim_4d \
-f=vwnd.mon.mean.nc \
-v=vwnd \
-p=12 \
-m=mesh_mask_wind_ncep2.nc
```

2.3 comp_clim_miss_3d

2.3.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.3.2 Latest revision

29/04/2026

2.3.3 Purpose

Compute a climatology (e.g., means and standard-deviations) from a tridimensional variable with missing values extracted from a NetCDF dataset and, optionally, the mask and scale factors of the 2-D grid-mesh associated with the input NetCDF variable.

It is assumed that the input NetCDF variable has a scalar *missing_value* or *_FillValue* attribute, which allows a proper identification of the missing values in the dataset, otherwise the procedure will exit with an error message.

Mean and standard-deviation are computed for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable. These statistics are computed by using all available observations for each grid-point time series. Since missing values are present, the number of observations used to compute the means and standard-deviations may vary from one grid-point to another in the 2-D grid-mesh associated with the NetCDF variable.

These means and standard-deviations may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

Refer to [comp_clim_3d](#), for details on how these univariate statistics are calculated in `comp_clim_miss_3d`. Moreover, if your data does not contain missing values in addition to those associated with a constant land-sea mask, use [comp_clim_3d](#) instead of `comp_clim_miss_3d` to estimate means and standard deviations from your dataset.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the means and standard-deviations with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence mask and scale factor variables which may be used to compute the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures such as [comp_serie_miss_3d](#), [comp_eof_miss_3d](#), etc. When computing the scale factors, it is assumed that the 2-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

2.3.4 Further Details

Usage

```
$ comp_clim_miss_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -p=periodicity           (optional) \
  -x=lon1,lon2             (optional) \
  -y=lat1,lat2             (optional) \
  -t=time1,time2           (optional) \
  -c=output_climatology_netcdf_file (optional) \
  -m=output_mesh_mask_netcdf_file  (optional) \
  -np=nobs_limit_by_period (optional) \
  -yl=lat11,lat12          (optional) \
  -mi=missing_value        (optional) \
  -ntr=number_of_time_records (optional) \
  -double                  (optional) \
  -bigfile                  (optional) \
  -hdf5                     (optional) \
  -cdf5                     (optional) \
  -tlimited                  (optional)
```

By default

- p=** the *periodicity* is equal to 1
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- c=** the *output_climatology_netcdf_file* is named `clim_netcdf_variable.nc`
- m=** the *output_mesh_mask_netcdf_file* is not created
- np=** this argument is equal to 0
- yl=** it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors

- mi=** the *missing_value* for the STD variable is equal to $1.e+20$
- ntr=** the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity*
- double** the standard-deviations are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the standard-deviations are stored as double-precision floating point numbers in the *output_climatology_netcdf_file*
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to construct the climatology.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices before using *comp_clim_miss_3d*.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to construct the climatology. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) It is assumed that the specified *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the value of this missing attribute.
- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and if some scale factors can not be computed, these scale factors are set to 1.
- 7) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 2-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 2-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 8) If the **-np=nobs_limit_by_period** and **-m=output_mesh_mask_netcdf_file** arguments are present, the mask in the *output_mesh_mask_netcdf_file* is constructed as follow:

- if the number of observations by period (as determined by the **-p=** argument) is less than *nobs_limit_by_period*, the corresponding mask value is set to 0 (e.g., missing) otherwise, the mask value is set to 1.

If the **-np=nobs_limit_by_period** argument is not specified and the **-m=output_mesh_mask_netcdf_file** argument is present, the mask is constructed as follow:

- if the total number of non-missing observations is 0, the corresponding mask value is set to 0 (e.g., missing), otherwise the mask value is set to 1.

- 9) The **-mi=missing_value** argument specifies the missing value indicator for the standard-deviation (STD) variable in the *output_climatology_netcdf_file*.

If the **-mi=** argument is not specified and the NetCDF variable has a *missing_value* or *_FillValue* attribute, the *missing_value* is set to $1.e+20$. This argument is not used if the NetCDF variable specified in the **-v=** argument has no *missing_value* or *_FillValue* attribute.

- 10) The **-ntr=number_of_time_records** argument specifies how many time records are read in each iteration of the loop for reading the input NetCDF variable.

By default, the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity* (as specified by the **-p=** argument). On very large dataset, it may be useful to reduce the *number_of_time_records* in order to decrease the memory used by the procedure.

- 11) The **-double** argument specifies that the standard-deviation variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_climatology_netcdf_file*.

- 12) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 13) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 14) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 15) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 16) For more details on the use of univariate statistics in the climate literature, see

- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](#)

Outputs

`comp_clim_miss_3d` creates an output NetCDF file that contains the means, standard-deviations and number of observations of the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_mean(periodicity,nlat,nlon)` : the means for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_std(periodicity,nlat,nlon)` : the standard-deviations for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_nobs(periodicity,nlat,nlon)` : the number of observations used to compute the statistics for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

The means, standard-deviations and numbers of observations are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values (zero values for the numbers of observations).

Optionally, `comp_clim_miss_3d` can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) `netcdf_variable_nmask(nlat,nlon)` : a presence-absence or land-sea 2-D mask associated with the input NetCDF variable.
- 2) `netcdf_variable_e1n(nlat,nlon)` : the first scale factor associated with the 2-D grid-mesh of the input NetCDF variable.
- 3) `netcdf_variable_e2n(nlat,nlon)` : the second scale factor associated with the 2-D grid-mesh of the input NetCDF variable.

Multiplying the two scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly means and standard-deviations from the NetCDF file `ST7_1m_00101_20012_grid_T_sosstsst.nc` which includes a NetCDF variable `sosstsst`, which may possibly contain missing values and store the results in a NetCDF file named `clim_ST7_1m_00101_20012_grid_T_sosstsst.nc`, use the following command :

```
$ comp_clim_miss_3d \
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \
-v=sosstsst \
-p=12 \
-c=clim_ST7_1m_00101_20012_grid_T_sosstsst.nc
```

- 2) For computing monthly means and standard-deviations from the NetCDF file `precip.mon.mean.nc`, which includes a NetCDF variable `precip` and store the results in a NetCDF file named `clim_precip_cmapstd.nc` and, in addition, generate an associated `*mesh_mask_netcdf_file*` named `meshmask.cmapstd.3d.nc`, use the following command (here the `-np=` argument is used to specify how to construct the `mesh_mask_netcdf_file`, see remarks above for further details) :

```
$ comp_clim_miss_3d \  
-f=precip.mon.mean.nc \  
-v=precip \  
-p=12 \  
-np=10 \  
-c=clim_precip_cmapstd.nc \  
-m=meshmask.cmapstd.3d.nc
```

2.4 comp_clim_miss_4d

2.4.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.4.2 Latest revision

29/04/2026

2.4.3 Purpose

Compute a climatology (e.g., means and standard-deviations) from a fourdimensional variable with missing values extracted from a NetCDF dataset and, optionally, the mask and scale factors of the 3-D grid-mesh associated with the input NetCDF variable.

It is assumed that the input NetCDF variable has a scalar *missing_value* or *_FillValue* attribute, which allows a proper identification of the missing values in the dataset, otherwise the procedure will exit with an error message.

Mean and standard-deviation are computed for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable. These statistics are computed by using all available observations for each grid-point time series. Since missing values are present, the number of observations used to compute the means and standard-deviations may vary from one grid-point to another in the 3-D grid-mesh associated with the NetCDF variable.

These means and standard-deviations may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

Refer to [comp_clim_4d](#), for details on how these univariate statistics are calculated in `comp_clim_miss_4d`. Moreover, if your data does not contain missing values in addition to those associated with a constant land-sea mask, use [comp_clim_4d](#) instead of `comp_clim_miss_4d` to estimate means and standard deviations from your dataset.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the means and standard-deviations with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence mask and scale factor variables which may be used to compute the surface of each cell in the 3-D grid-mesh associated with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures. When computing the scale factors, it is assumed that the 3-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

2.4.4 Further Details

Usage

```
$ comp_clim_miss_4d \
-f=input_netcdf_file \
-v=netcdf_variable \
-p=periodicity (optional) \
-x=lon1,lon2 (optional) \
-y=lat1,lat2 (optional) \
-z=level1,level2 (optional) \
-t=time1,time2 (optional) \
-c=output_climatology_netcdf_file (optional) \
-m=output_mask_netcdf_file (optional) \
-np=nobs_limit_by_period (optional) \
-vz=name_of_the_vertical_netcdf_variable (optional) \
-z0=value_of_the_highest_level (optional) \
-sf=method_for_computing_the_third_scale_factor (optional : method1, method2) \
-yl=lat11,lat12 (optional) \
-mi=missing_value (optional) \
-ntr=number_of_time_records (optional) \
-double (optional) \
-bigfile (optional) \
-hdf5 (optional) \
-cdf5 (optional) \
-tlimited (optional)
```

By default

- p=** the *periodicity* is equal to 1
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- c=** the *output_climatology_netcdf_file* is named `clim_netcdf_variable.nc`
- m** the *output_mesh_mask_netcdf_file* is not created
- np=** this argument is equal to 0
- vz=** the variable with the same name as the third dimension, if any (e.g., the associated coordinate variable)
- z0=** a value of 0 is assumed for the highest level when computing the third scale factor
- sf=** method2 (e.g., it is assumed that each level or depth is located in the middle of each layer and the third scale factor is computed accordingly)
- yl=** it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors
- mi=** the *missing_value* for the STD variable in the output NetCDF file is set to `1.e+20`
- ntr=** the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity*

- double** the standard-deviations are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the standard-deviations are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.
- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to construct the climatology.

The longitude, latitude or depth range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_clim_miss_4d*.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to construct the climatology. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) It is assumed that the specified *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the value of this missing attribute.
- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 3-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 3-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 7) If the **-m=output_mesh_mask_netcdf_file** argument is present, the third scale factor is computed with the help of the vertical coordinate variable (or the **-vz=name_of_the_vertical_netcdf_variable**) if this vertical coordinate variable is strictly monotonic.
- 8) If the **-m=output_mesh_mask_netcdf_file** argument is present and if some scale factors can not be computed, these scale factors are set to 1.

- 9) The **-z0=value_of_the_highest_level** argument specifies a value for the highest level or depth in order to compute the first/last element of the third scale factor. The default value is 0.
- 10) The **-sf=method_for_computing_the_third_scale_factor** argument allows to specify the method for computing the third scale factor if a mesh-mask NetCDF file is created:
 - **-sf=method1** : the third scale factor is computed as the differences between successive levels (or depths).
 - **-sf=method2** : the third scale factor is computed by assuming that each level or depth is located at the middle of the corresponding layer.
- 11) If the **-np=nobs_limit_by_period** and **-m=output_mesh_mask_netcdf_file** arguments are present, the mask in the *output_mesh_mask_netcdf_file* is constructed as follow:
 - If the number of observations by period (as determined by the **-p=** argument) is less than *nobs_limit_by_period*, the corresponding mask value is set to 0 (e.g., missing), otherwise the mask value is set to 1.

If the **-np=nobs_limit_by_period** argument is not specified and the **-m=output_mesh_mask_netcdf_file** argument is present, the mask is constructed as follow:

 - If the total number of non-missing observations is 0, the corresponding mask value is set to 0 (e.g., missing), otherwise the mask value is set to 1.
- 12) The **-mi=missing_value** argument specifies the missing value indicator for the standard-deviation (STD) variable in the *output_climatology_netcdf_file*. If the **-mi=** argument is not specified and the NetCDF variable has a *missing_value* or *_FillValue* attribute, the *missing_value* is set to 1.e+20. This argument is not used if the NetCDF variable specified in the **-v=** argument has no *missing_value* or *_FillValue* attribute.
- 13) The **-ntr=number_of_time_records** argument specifies how many time records are read in each iteration of the loop for reading the input NetCDF variable. By default, the *number_of_time_records* read in each iteration of the procedure is set to the *periodicity* (as specified by the **-p=** argument). On very large dataset, it may be useful to reduce the *number_of_time_records* in order to decrease the memory used by the procedure.
- 14) The **-double** argument specifies that the standard-deviation (STD) variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_climatology_netcdf_file*.
- 15) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 16) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 17) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 18) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 19) For more details on the use of univariate statistics in the climate literature, see
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](#)

Outputs

`comp_clim_miss_4d` creates an output NetCDF file that contains the means, standard-deviations and number of observations of the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the vertical and spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_mean(periodicity, nlev, nlat, nlon)` : the means for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_std(periodicity, nlev, nlat, nlon)` : the standard-deviations for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_nobs(periodicity, nlev, nlat, nlon)` : the number of observations used to compute the statistics for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.

The means, standard-deviations and numbers of observations are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values (zero values for the numbers of observations).

Optionally, `comp_clim_miss_4d` can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) `netcdf_variable_nmask(nlev, nlat, nlon)` : a presence-absence or height-land-sea 3-D mask associated with the input NetCDF variable.
- 2) `netcdf_variable_e1n(nlat, nlon)` : the first scale factor associated with the 3-D grid-mesh of the input NetCDF variable.
- 3) `netcdf_variable_e2n(nlat, nlon)` : the second scale factor associated with the 3-D grid-mesh of the input NetCDF variable.
- 4) `netcdf_variable_e3n(nlev, 1, 1)` : the third scale factor associated with the 3-D grid-mesh of the input NetCDF variable.

Multiplying the two first scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable. Multiplying the three scale factors together gives the volume (or a quantity proportional to the weight if the unit of the vertical coordinate variable is in hPa) of each parcel in the 3-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly means and standard-deviations from the NetCDF file `ST7_1m_00101_20012_grid_T_votemper.nc` which includes a NetCDF variable

votemper, which may contain missing values and store the results in a NetCDF file named `clim_ST7_1m_00101_20012_grid_T_votemper.nc`, use the following command :

```
$ comp_clim_miss_4d \
-f=ST7_1m_0101_20012_grid_T_votemper.nc \
-v=votemper \
-p=12 \
-c=clim_ST7_1m_00101_20012_grid_T_votemper.nc
```

- 2) For computing monthly means and standard-deviations from the NetCDF file `vwnd.mon.mean.nc`, which includes a NetCDF variable `vwnd` and store the results in a NetCDF file named `clim_vwnd.nc` and, in addition, generate an associated mesh_mask_NetCDF_file named `mesh_mask_wind_ncep2.nc`, use the following command :

```
$ comp_clim_miss_4d \
-f=vwnd.mon.mean.nc \
-v=vwnd \
-p=12 \
-m=mesh_mask_wind_ncep2.nc
```

2.5 comp_composite_3d

2.5.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.5.2 Latest revision

30/04/2026

2.5.3 Purpose

Compute a composite analysis from a tridimensional variable extracted from a NetCDF dataset and perform statistical tests on the differences between the composite mean and the overall mean (e.g., the mean of the parent finite population) for each cell of the 2-D grid-mesh associated with the tridimensional NetCDF variable. The composite statistics and their associated critical probabilities may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

Composite analysis is a classical statistical tool in climatology. Its purpose is to highlight the space-time evolution of a time series or a gridded dataset (for instance surface temperature) according to the variations of a given index time series (for instance an ENSO index).

For the sake of simplicity, in the following, we take “years” as our time samples but the approach would also be valid with days, months,...

The first step of the method consists in defining groups of years according to the values of the index time series. The second step involves the description of each group of years with the help of the gridded dataset (or other time series). Usually, this description is obtained by computing composite means for each group of years. As the years used to estimate the composite means are restricted to those years belonging to each group, the resulting maps may be useful to describe the spatial-time variability associated with each group of years identified by the index time series.

While it is easy to compute composite means, assessing the statistical significance of these composite maps is a more difficult task. This is often done with the help of a classical Student’s two sample t-test, where one sample consists of

the years belonging to one group and the second sample of the other years in the observed time series. In the usual context of statistical inference, this procedure is used to test the hypothesis of equal population means on the basis of two random samples independently drawn from two normal populations with a common variance, but possibly different means [vonStorch_Zwiers]. The assumptions of random selection and Gaussian distribution are essential for the validity of the test. As noted by [Brown_Hall], the use of the Student's two sample t-test for statistical inference in composite analyses, as is traditionally used in the climate literature, is not valid since composite means are computed from groups of years which are not randomly selected, but rather by the value of the index time series. The assumption of Gaussian distribution is also difficult to verify and inappropriate here since the data distribution in each composite group is unknown even if the original data distribution is assumed to be Gaussian.

Therefore, we need an alternative procedure for statistical significance testing of the composite maps in order to overcome the drawbacks of the Student's t-test.

To obtain more meaningful results, `comp_composite_3d` uses a resampling scheme [Noreen] or, alternatively, critical probabilities based on the theory of sampling in a finite population (e.g., the observed time observations) and a normal approximation, to estimate the statistical significance of the composite means [Terray_etal] .

The following approach to assess chance of occurrence of the composite maps is suggested.

Let $x(1), x(2), \dots, x(n)$ be the raw time series of one grid point X in the dataset observed during the n years included in the whole finite population, which is used in the composite analysis and

- **MEAN** = $\text{sum}(X(:))/n$
- **VAR** = $\text{sum}([X(:)-\text{MEAN}]^2)/n$
- **STD** = $\text{sqrt}(\text{VAR})$

the sample mean, variance and standard-deviation estimated from the n data values. Suppose now that the n years are classified by means of an index time series into k groups and let $C(1), C(2), \dots, C(k)$ be the k groups of years. In common applications k is less than 3. Finally, let $n(1), n(2), \dots, n(k)$ be the number of years in each group and $\mathbf{M}(1), \mathbf{M}(2), \dots, \mathbf{M}(k)$ be the overall mean computed from the years belonging to each group.

Suppose now that we want to assess if the $n(j)$ values for the grid point X observed during the years in class $C(j)$ are significantly different from the values observed on all the years.

This problem can be treated as the statistical testing of a null hypothesis. The null hypothesis **H0**, as usually stated, is that these $n(j)$ values were allocated in class $C(j)$ at random and without replacement among the finite population of the n years included in the composite analysis. In order to test this hypothesis, we may compare the actual mean $\mathbf{M}(j)$ with the expected value of the mean, assuming the null hypothesis **H0** is true. More precisely, it may be shown with the help of the theory of random sampling without replacement in a finite population, that **MEAN** is the expected value of the mean for the years in class $C(j)$ and that

$$\text{STD-ERROR}(j) = \text{sqrt}([n - n(j)] / [n(j) \cdot (n - 1)]) \cdot \text{STD}$$

is the standard-error of the mean for the years in class $C(j)$ if the null hypothesis **H0** is true. In order to assess the validity of this null hypothesis, we may then compute the following sample criterion for each grid-point X in the gridded dataset :

$$U(j) = [\mathbf{M}(j) - \text{MEAN}] / \text{STD-ERROR}(j)$$

In statistical terms, large absolute values of this **U** statistic indicate a strong departure from the null hypothesis. For example, $U(j) = 8$, means that the overall mean $\mathbf{M}(j)$ in class $C(j)$ deviates from the finite population mean **MEAN** by over 8 standard-deviations under the null hypothesis of a random selection of years in class $C(j)$ from the finite population of the n years.

Finally, the **U** statistic may be used to compute a critical probability, that is to say, the probability that, under the null hypothesis of a random selection of the years in group $C(j)$ from the finite population, the **U** statistic takes values more discordant than the observed sampled criterion $U(j)$. More precisely, if we assume a two-sided test, this critical probability is defined by

$$\text{PROB}(j) = \text{P}(\text{abs}(U) > \text{abs}(U(j))) \text{ if the null hypothesis } \mathbf{H0} \text{ is true}$$

For this purpose, we need to find the null distribution of the **U** statistic. This null distribution may be estimated by means of simulation, as in bootstrapping techniques and (approximate) randomization tests [Noreen], or approximated since it is possible to demonstrate that **U** is approximately distributed as a Gaussian distribution with mean zero and standard-deviation unity if **H0** is true by the use of an extension of the central limit theorem.

In other words, the **U** statistic and its associated critical probability **PROB** estimated from the theory of random sampling in a finite population may be used to determine whether the null hypothesis **H0** is to be retained or rejected as in a classical statistical test.

The procedure outlined above may be applied to each grid-point in the gridded dataset and to each group of years defined in the composite analysis in order to assess the statistical significance of the composite maps. Finally, it should be noted that if there are only two groups in the composite analysis, this test procedure may also be used to test the difference between the means in the two groups. In other words, testing the significance of the mean in one group is the same as testing the mean in the other group, or the difference between the two group means in this case.

For more details and examples of use of the statistical test described above, and used in `comp_composite_3d` to assess the statistical significance of the composite means, consult the references cited below [Terray_etal] [Noreen].

If your data contains missing values, use `comp_composite_miss_3d` instead of `comp_composite_3d` to estimate the statistics of your composite analysis from your gappy dataset.

Finally, if the NetCDF variable is fourdimensional use `comp_composite_4d` instead of `comp_composite_3d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro.

2.5.4 Further Details

Usage

```
$ comp_composite_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -c=input_climatology_netcdf_file \
  -a=year1,year2,year3, ... , yearn \
  -m=input_mesh_mask_netcdf_file      (optional) \
  -g=grid_type                        (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                       (optional) \
  -y=lat1,lat2                       (optional) \
  -t=time1,time2                     (optional) \
  -s=type_of_statistics               (optional : comp, diff) \
  -alg=algorithm                      (optional : normal, simul) \
  -nb=number_of_shuffles              (optional) \
  -o=output_composite_netcdf_file     (optional) \
  -mi=missing_value                   (optional) \
  -nostd                              (optional) \
  -double                             (optional) \
  -bigfile                            (optional) \
  -hdf5                               (optional) \
  -cdf5                               (optional) \
  -tlimited                            (optional)
```

By default

-m= an `output_mesh_mask_netcdf_file` is not used

- g=** the *grid_type* is set to `n` which means that the 2-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- s=** the *type_of_statistics* is set to `comp`
- alg=** the *algorithm* is set to `normal`
- nb=** *number_of_shuffles* is set to `99`
- o=** the *output_composite_netcdf_file* is named `composite_netcdf_variable.nc`
- mi=** the *missing_value* attribute for the NetCDF variables in the output NetCDF file is set to `1.e+20`
- nostd** the composite fields of the input NetCDF variable are standardized. If **-nostd** is activated, the composite fields are not standardized
- double** the composite statistics are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a composite analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The *input_climatology_netcdf_file* specified with the **-c=** argument must contain the means and standard deviations for the finite parent population. The periodicity for the composite analysis is also deduced from the number of observations and periodicity in the *input_climatology_netcdf_file*. This *input_climatology_netcdf_file* must have been created by `comp_clim_3d` applied to the same *netcdf_variable* with an identical **-t=time1,time2** argument as used in `comp_composite_3d` in order to obtain correct statistics in the output NetCDF file *output_composite_netcdf_file*.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the climatology (in the *input_climatology_netcdf_file*) must agree.

- 3) The **-a=** argument lists the indices of the years (or seasons, months or days depending on the sampling of the observations in the *input_netcdf_file*), which must be included in the composite analysis (e.g., for computing the composite means). The indices of the years are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing).

The list may be specified in different formats:

- **-a=n1,n2,...nn** allows to select for years *n1*, *n2*, ... and *nn* in the *input_netcdf_file*
- **-a=n1:n2** allows to select for years *n1* to *n2* in the *input_netcdf_file*

The two forms of the **-a=** argument may be combined and repeated any number of times, but duplicate years are not allowed.

Remember also, when specifying the indices of the years in the **-a=** argument, that the “length” of a year in the *input_netcdf_file* is determined by the periodicity of the observations as deduced from the number of time observations and periodicity in the *input_climatology_netcdf_file*.

This periodicity will also determine how many composite means $\mathbf{M}(j)$ and statistics $\mathbf{U}(j)$ will be computed and stored in the *output_composite_netcdf_file* by `comp_composite_3d`.

- 4) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before computing the composite analysis. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the climatology (in the *input_climatology_netcdf_file*) and the mask (if an *input_mesh_mask_netcdf_file* is used) must agree.

Refer to [comp_clim_3d](#) or [comp_mask_3d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using `comp_composite_3d`.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.
- 6) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used in the composite analysis.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using `comp_composite_3d`.

- 7) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., $time2 - time1 + 1$) must be a whole multiple of the *periodicity* as deduced from the number of time observations in the *input_climatology_netcdf_file* if **-alg=simul** (see remark below).

It is also assumed that the selected time period matches exactly the time period used to compute the climatology in the *input_climatology_netcdf_file*. Moreover, the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.

- 8) The **-s=type_of_statistics** argument specified how the composite variable *netcdf_variable_composite* stored in the *output_composite_netcdf_file* is computed:
 - **-s=comp** means that the composites are computed as the standardized differences between the mean of the composite years and the overall mean (e.g., the mean of the parent finite population) for each grid-point.
 - **-s=diff** means that the composites are computed as the standardized differences between the mean of the composite years and the mean of the other years in the parent finite population for each grid-point (option useful when we consider only two groups in the composite analysis).

The standard deviations from the finite parent population are used for the standardization in both cases and are read from the *input_climatology_netcdf_file*.

Finally, note that this argument does not affect the other variables stored in the *output_composite_netcdf_file*.

- 9) The **-nostd** argument specifies that the composite variable *netcdf_variable_composite* in the *output_composite_netcdf_file* must not be standardized.

By default, the composites are standardized in the *output_composite_netcdf_file*.

- 10) The **-alg=** argument selects the method for computing critical probabilities associated with the composite mean $\mathbf{M}(j)$ and the $\mathbf{U}(j)$ statistic:

- If **-alg=normal**, a normal approximation is used to compute the critical probabilities associated with the composite mean $\mathbf{M}(j)$ and the $\mathbf{U}(j)$ statistic.
- If **-alg=simul**, the critical probabilities are estimated by a resampling method. If **-alg=simul** the selected time period (e.g., *time2* - *time1* + 1) must be a whole multiple of the periodicity.

- 11) The **-nb=number_of_shuffles** argument specifies the number of shuffles for the resampling procedure used to estimate critical probabilities if **-alg=simul**.

- 12) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_composite_netcdf_file*.

If the **-mi=** argument is not specified the *missing_value* attribute is set to $1.e+20$ for the NetCDF variables in the *output_composite_netcdf_file*.

- 13) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 18) It is assumed that the data has no missing values. If it is the case, use *comp_composite_miss_3d* instead of *comp_composite_3d*.

- 19) For more details on composite analysis and statistical testing in composite analysis, see

- “The Use of t values in Composite Analyses” by Brown, T.J., and Hall, B.L., Journal of climate, vol. 12, 2941-2944, 1999. doi: [10.1175/1520-0442\(1999\)012<2941:TUOTVI>2.0.CO;2](https://doi.org/10.1175/1520-0442(1999)012<2941:TUOTVI>2.0.CO;2)

- “Sea Surface Temperature associations with the Late Indian Summer Monsoon”, by Terray, P., Delecluse P., Labattu S., Terray L., Climate Dynamics, vol. 21, 593-618, 2003. doi: [10.1007/s00382-003-0354-0](https://doi.org/10.1007/s00382-003-0354-0)
- “Computer-intensive methods for testing hypotheses: an introduction”, by Noreen, E.W., Wiley and Sons, New York, USA, 1989. ISBN: [978-0-471-61136-3](https://www.wiley.com/9780471611363)
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](https://www.cambridge.org/9780521012300)

Outputs

`comp_composite_3d` creates an output NetCDF file that contains the composite statistics and critical probabilities associated with the composite means, $\mathbf{M}(j)$, taking into account eventually the periodicity of the data as determined by the `-c=input_climatology_netcdf_file` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the length of the spatial dimensions of the input NetCDF variable) and *periodicity* time observations (*periodicity* is the number of time observations in the `input_climatology_netcdf_file`):

- 1) `netcdf_variable_compmean(periodicity, nlat, nlon)` : the mean fields $\mathbf{M}(j)$ of the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the number of observations in the `input_climatology_netcdf_file` (specified with the help of the `-c=` argument).
- 2) `netcdf_variable_compstd(periodicity, nlat, nlon)` : the standard-deviation fields of the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the `-c=input_climatology_netcdf_file` argument. This corresponds to the standard-deviation of the years belonging to class $C(j)$.
- 3) `netcdf_variable_qstd(periodicity, nlat, nlon)` : the ratio between the standard deviations fields of the selected time observations to the standard deviations fields of all the time observations, taking into account the periodicity of the data as determined by the `-c=input_climatology_netcdf_file` argument. The standard deviations fields for all the observations are extracted from the `input_climatology_netcdf_file` specified in the `-c=` argument.
- 4) `netcdf_variable_composite(periodicity, nlat, nlon)` : the composite fields of the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the number of observations in the `input_climatology_netcdf_file` (specified with the help of the `-c=` argument).

How these composite fields are calculated is determined by the `-s=` and `-nostd` arguments.

- 5) `netcdf_variable_u(periodicity, nlat, nlon)` : the $\mathbf{U}(j)$ statistics for the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the number of observations in the `input_climatology_netcdf_file` (specified with the help of the `-c=` argument).

See above and the second publication cited above for more details on the definition of the $\mathbf{U}(j)$ statistic.

- 6) `netcdf_variable_prob(periodicity, nlat, nlon)` : the critical probabilities associated with the composite means $\mathbf{M}(j)$ and the $\mathbf{U}(j)$ statistics.

These critical probabilities are computed under the null hypothesis that the selected time observations (with the help of the `-a=` argument) come from the same finite population as the other observations in the `input_netcdf_file`. Small probabilities indicate a large departure from the null hypothesis H_0 .

The `-alg=algorithm` argument determines how these critical probabilities are computed.

- 7) *netcdf_variable_compnobs*(periodicity) : the number of observations used to compute the composite means and standard-deviations stored in the NetCDF variables *netcdf_variable_compmean* and *netcdf_variable_compst* defined above.

All these statistics are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Example

- 1) For computing a composite analysis from a tridimensional NetCDF variable *sosstsst* in the NetCDF file *ST7_1m_00101_20012_grid_T_sosstsst.nc*, assessing the significance of the results with a monte carlo simulation with 999 shuffles and, finally, storing the results in a NetCDF file named *composite_ST7_1m_sosstsst_grid_T.nc*, use the following command (note that the critical probabilities associated with the U statistics are estimated with the help of a resampling method using 999 surrogate time series since **-alg=simul** and **-nb=999** are specified) :

```
$ comp_composite_3d \  
-f=ST7_1m_00101_20012_sosstsst_grid_T.nc \  
-v=sosstsst \  
-c=clim_sosstsst_grid_T.nc \  
-a=10,11,20,55,143 \  
-g=t \  
-m=mesh_mask_ST7_grid_T.nc \  
-alg=simul \  
-nb=999 \  
-o=composite_ST7_1m_sosstsst_grid_T.nc
```

2.6 comp_composite_4d

2.6.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.6.2 Latest revision

30/04/2026

2.6.3 Purpose

Compute a composite analysis from a fourdimensional variable extracted from a NetCDF dataset and perform statistical tests on the differences between the composite mean and the overall mean (e.g., the mean of the parent finite population) for each cell of the 3-D grid-mesh associated with the fourdimensional NetCDF variable. The composite statistics and tests may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

Composite analysis is a classical statistical tool in climatology. Its purpose is to highlight the space-time evolution of a time series or a gridded dataset (for instance surface temperature) according to the variations of a given index time series (for instance an ENSO index).

For the sake of simplicity, in the following, we take “years” as our time samples but the approach would also be valid with days ,months,...

The first step of the method consists in defining groups of years according to the values of the index time series. The second step involves the description of each group of years with the help of the gridded dataset (or other time series). Usually, this description is obtained by computing composite means for each group of years. As the years used to estimate the composite means are restricted to those years belonging to each group, the resulting maps may be useful to describe the spatial-time variability associated with each group of years identified by the index time series.

While it is easy to compute composite means, assessing the statistical significance of these composite maps is a more difficult task. This is often done with the help of a classical Student’s two sample t-test, where one sample consists of the years belonging to one group and the second sample of the other years in the observed time series. In the usual context of statistical inference, this procedure is used to test the hypothesis of equal population means on the basis of two random samples independently drawn from two normal populations with a common variance, but possibly different means [vonStorch_Zwiers]. The assumptions of random selection and Gaussian distribution are essential for the validity of the test. As noted by [Brown_Hall], the use of the Student’s two sample t-test for statistical inference in composite analyses, as is traditionally used in the climate literature, is not valid since composite means are computed from groups of years which are not randomly selected, but rather by the value of the index time series. The assumption of Gaussian distribution is also difficult to verify and inappropriate here since the data distribution in each composite group is unknown even if the original data distribution is assumed to be Gaussian.

Therefore, we need an alternative procedure for statistical significance testing of the composite maps in order to overcome the drawbacks of the Student’s t-test.

To obtain more meaningful results, comp_composite_3d uses a resampling scheme [Noreen] or, alternatively, critical probabilities based on the theory of sampling in a finite population (e.g., the observed time observations) and a normal approximation, to estimate the statistical significance of the composite means [Terray_etal] .

The following approach to assess chance of occurrence of the composite maps is suggested.

Let $x(1), x(2), \dots, x(n)$ be the raw time series of one grid point X in the dataset observed during the n years included in the whole finite population, which is used in the composite analysis and

- **MEAN** = $\text{sum}(X(:))/n$
- **VAR** = $\text{sum}([X(:)-\text{MEAN}]^2)/n$
- **STD** = $\text{sqrt}(\text{VAR})$

the sample mean, variance and standard-deviation estimated from the n data values. Suppose now that the n years are classified by means of an index time series into k groups and let $C(1), C(2), \dots, C(k)$ be the k groups of years. In common applications k is less than 3. Finally, let $n(1), n(2), \dots, n(k)$ be the number of years in each group and $\mathbf{M}(1), \mathbf{M}(2), \dots, \mathbf{M}(k)$ be the overall mean computed from the years belonging to each group.

Suppose now that we want to assess if the $n(j)$ values for the grid point X observed during the years in class $C(j)$ are significantly different from the values observed on all the years.

This problem can be treated as the statistical testing of a null hypothesis. The null hypothesis **H0**, as usually stated, is that these $n(j)$ values were allocated in class $C(j)$ at random and without replacement among the finite population of the n years included in the composite analysis. In order to test this hypothesis, we may compare the actual mean $\mathbf{M}(j)$ with the expected value of the mean, assuming the null hypothesis **H0** is true. More precisely, it may be shown with the help of the theory of random sampling without replacement in a finite population, that **MEAN** is the expected value of the mean for the years in class $C(j)$ and that

$$\text{STD-ERROR}(j) = \text{sqrt}([n - n(j)] / [n(j) \cdot (n - 1)]) \cdot \text{STD}$$

is the standard-error of the mean for the years in class $C(j)$ if the null hypothesis **H0** is true. In order to assess the validity of this null hypothesis, we may then compute the following sample criterion for each grid-point X in the gridded dataset :

$$U(j) = [\mathbf{M}(j) - \text{MEAN}] / \text{STD-ERROR}(j)$$

In statistical terms, large absolute values of this **U** statistic indicate a strong departure from the null hypothesis. For example, $U(j) = 8$, means that the overall mean $M(j)$ in class $C(j)$ deviates from the finite population mean **MEAN** by over 8 standard-deviations under the null hypothesis of a random selection of years in class $C(j)$ from the finite population of the n years.

Finally, the **U** statistic may be used to compute a critical probability, that is to say, the probability that, under the null hypothesis of a random selection of the years in group $C(j)$ from the finite population, the **U** statistic takes values more discordant than the observed sampled criterion $U(j)$. More precisely, if we assume a two-sided test, this critical probability is defined by

$\text{PROB}(j) = P(\text{abs}(U) > \text{abs}(U(j)))$ if the null hypothesis **H0** is true

For this purpose, we need to find the null distribution of the **U** statistic. This null distribution may be estimated by means of simulation, as in bootstrapping techniques and (approximate) randomization tests [Noreen], or approximated since it is possible to demonstrate that **U** is approximately distributed as a Gaussian distribution with mean zero and standard-deviation unity if **H0** is true by the use of an extension of the central limit theorem.

In other words, the **U** statistic and its associated critical probability **PROB** estimated from the theory of random sampling in a finite population may be used to determine whether the null hypothesis **H0** is to be retained or rejected as in a classical statistical test.

The procedure outlined above may be applied to each grid-point in the gridded dataset and to each group of years defined in the composite analysis in order to assess the statistical significance of the composite maps. Finally, it should be noted that if there are only two groups in the composite analysis, this test procedure may also be used to test the difference between the means in the two groups. In other words, testing the significance of the mean in one group is the same as testing the mean in the other group, or the difference between the two group means in this case.

For more details and examples of use of the statistical test described above, and used in `comp_composite_4d` to assess the statistical significance of the composite means, consult the references cited below [Terray_etal] [Noreen].

If the NetCDF variable is tridimensional, use `comp_composite_3d` instead of `comp_composite_4d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro.

2.6.4 Further Details

Usage

```
$ comp_composite_4d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -c=input_climatology_netcdf_file \
  -a=year1,year2,year3,..., yearn \
  -m=input_mesh_mask_netcdf_file      (optional) \
  -g=grid_type                        (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                        (optional) \
  -y=lat1,lat2                        (optional) \
  -z=level1,level2                    (optional) \
  -t=time1,time2                      (optional) \
  -s=type_of_statistics                (optional : comp, diff) \
  -alg=algorithm                      (optional : normal, simul) \
  -o=output_composite_netcdf_file     (optional) \
  -nb=number_of_shuffles              (optional) \
  -mi=missing_value                   (optional) \
  -nostd                              (optional) \
  -double                             (optional) \
```

(continues on next page)

(continued from previous page)

-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- m=** an *output_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 3-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- s=** the *type_of_statistics* is set to *comp*
- alg=** the *algorithm* is set to *normal*
- nb=** *number_of_shuffles* is set to 99
- o=** the *output_composite_netcdf_file* is named *composite_netcdf_variable.nc*
- mi=** the *missing_value* attribute in the output NetCDF file is set to *1.e+20*
- nostd** the composite fields of the input NetCDF variable are standardized. If **-nostd** is activated, the composite fields are not standardized
- double** the composite statistics are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a composite analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The *input_climatology_netcdf_file* specified with the **-c=** argument must contain the means and standard deviations for the finite parent population. The periodicity for the composite analysis is also deduced from the number of observations and periodicity in the *input_climatology_netcdf_file*. This *input_climatology_netcdf_file* must have been created by *comp_clim_4d* applied to the same *netcdf_variable* with an identical **-t=time1,time2** argument as used in *comp_composite_4d* in order to obtain correct statistics in the output NetCDF file *output_composite_netcdf_file*.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the climatology (in the *input_climatology_netcdf_file*) must agree.

- 3) The **-a=** argument lists the indices of the years (or seasons, months or days depending on the sampling of the observations in the *input_netcdf_file*), which must be included in the composite analysis (e.g., for computing the composite means). The indices of the years are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing).

The list may be specified in different formats:

- **-a=n1,n2,...nn** allows to select for years *n1*, *n2*, ... and *nn* in the *input_netcdf_file*
- **-a=n1:n2** allows to select for years *n1* to *n2* in the *input_netcdf_file*

The two forms of the **-a=** argument may be combined and repeated any number of times, but duplicate years are not allowed.

Remember also, when specifying the indices of the years in the **-a=** argument, that the “length” of a year in the *input_netcdf_file* is determined by the periodicity of the observations as deduced from the number of time observations and periodicity in the *input_climatology_netcdf_file*.

This periodicity will also determine how many composite means **M(j)** and statistics **U(j)** will be computed and stored in the *output_composite_netcdf_file* by `comp_composite_4d`.

- 4) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this fourdimensional NetCDF variable as a rectangular matrix of observed variables before computing the composite analysis. By default, it is assumed that each cell in the 3-D grid-mesh associated with the input fourdimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the climatology (in the *input_climatology_netcdf_file*) and the mask (if an *input_mesh_mask_netcdf_file* is used) must agree.

Refer to `comp_clim_4d` or `comp_mask_4d` for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using `comp_composite_4d`.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f*, it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.
- 6) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* are used in the composite analysis.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to `comp_mask_4d` for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using `comp_composite_4d`.

- 7) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., *time2* - *time1* + 1) must be a whole multiple of the *periodicity* as deduced from the number of time observations in the *input_climatology_netcdf_file* if **-alg=simul** (see remark below).

It is also assumed that the selected time period matches exactly the time period used to compute the climatology in the *input_climatology_netcdf_file*. Moreover, the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.

- 8) The **-s=type_of_statistics** argument specified how the composite variable *netcdf_variable_composite* stored in the *output_composite_netcdf_file* is computed:

- **-s=comp** means that the composites are computed as the standardized differences between the mean of the composite years and the overall mean (e.g., the mean of the parent finite population) for each grid-point.
- **-s=diff** means that the composites are computed as the standardized differences between the mean of the composite years and the mean of the other years in the parent finite population for each grid-point (option useful when we consider only two groups in the composite analysis).

The standard deviations from the finite parent population are used for the standardization in both cases and are read from the *input_climatology_netcdf_file*.

Finally, note that this argument does not affect the other variables stored in the *output_composite_netcdf_file*.

- 9) The **-nostd** argument specifies that the composite variable *netcdf_variable_composite* in the *output_composite_netcdf_file* must not be standardized.

By default, the composites are standardized in the *output_composite_netcdf_file*.

- 10) The **-alg=** argument selects the method for computing critical probabilities associated with the composite mean $\mathbf{M}(j)$ and the $\mathbf{U}(j)$ statistic:

- If **-alg=normal**, a normal approximation is used to compute the critical probabilities associated with the composite mean $\mathbf{M}(j)$ and the $\mathbf{U}(j)$ statistic.
- If **-alg=simul**, the critical probabilities are estimated by a resampling method. If **-alg=simul** the selected time period (e.g., *time2* - *time1* + 1) must be a whole multiple of the periodicity.

- 11) The **-nb=number_of_shuffles** argument specifies the number of shuffles for the resampling procedure used to estimate critical probabilities if **-alg=simul**.

- 12) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_composite_netcdf_file*.

If the **-mi=** argument is not specified the *missing_value* attribute is set to $1.e+20$ for the NetCDF variables in the *output_composite_netcdf_file*.

- 13) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 18) It is assumed that the data has no missing values.
- 19) For more details on composite analysis and statistical testing in composite analysis, see
 - “The Use of t values in Composite Analyses” by Brown, T.J., and Hall, B.L., Journal of climate, vol. 12, 2941-2944, 1999. doi: [10.1175/1520-0442\(1999\)012<2941:TUOTVI>2.0.CO;2](https://doi.org/10.1175/1520-0442(1999)012<2941:TUOTVI>2.0.CO;2)
 - “Sea Surface Temperature associations with the Late Indian Summer Monsoon”, by Terray, P., Delecluse P., Labattu S., Terray L., Climate Dynamics, vol. 21, 593-618, 2003. doi: [10.1007/s00382-003-0354-0](https://doi.org/10.1007/s00382-003-0354-0)
 - “Computer-intensive methods for testing hypotheses: an introduction”, by Noreen, E.W., Wiley and Sons, New York, USA, 1989. ISBN: [978-0-471-61136-3](https://www.wiley.com/9780471611363)
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](https://www.cambridge.org/9780521012300)

Outputs

`comp_composite_4d` creates an output NetCDF file that contains the composite statistics and critical probabilities associated with the composite means, taking into account eventually the periodicity of the data as determined by the `-c=input_climatology_netcdf_file` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the length of the vertical and spatial dimensions of the input NetCDF variable) and *periodicity* time observations (*periodicity* is the number of time observations in the *input_climatology_netcdf_file*) :

- 1) `netcdf_variable_compmean(periodicity, nlev, nlat, nlon)` : the mean fields of the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the number of observations in the *input_climatology_netcdf_file* (specified with the help of the `-c=` argument).
- 2) `netcdf_variable_compstd(periodicity, nlev, nlat, nlon)` : the standard-deviation fields of the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the `-c=input_climatology_netcdf_file` argument.
- 3) `netcdf_variable_qstd(periodicity, nlev, nlat, nlon)` : the ratio between the standard deviations fields of the selected time observations to the standard deviations fields of all the time observations, taking into account the periodicity of the data as determined by the `-c=input_climatology_netcdf_file` argument. The standard deviations fields for all the observations are extracted from the *input_climatology_netcdf_file* specified in the `-c=` argument.
- 4) `netcdf_variable_composite(periodicity, nlev, nlat, nlon)` : the composite fields of the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the number of observations in the *input_climatology_netcdf_file* (specified with the help of the `-c=` argument).

How these composite fields are calculated is determined by the `-s=` and `-nostd` arguments.

- 5) `netcdf_variable_u(periodicity, nlev, nlat, nlon)` : the $U(j)$ statistics for the time observations selected with the help of the `-a=` argument, taking into account the periodicity of the data as determined by the number of observations in the *input_climatology_netcdf_file* (specified with the help of the `-c=` argument).

See above and the second publication cited above for more details on the definition of the $U(j)$ statistic.

- 6) *netcdf_variable_prob*(periodicity, nlev, nlat, nlon) : the critical probabilities associated with the U statistics.

These critical probabilities are computed under the null hypothesis that the selected time observations (with the help of the **-a=** argument) come from the same finite population as the other observations in the *input_netcdf_file*. Small probabilities indicate a large departure from the null hypothesis **H0**.

The **-alg=algorithm** argument determines how these critical probabilities are computed.

- 7) *netcdf_variable_compnobs*(periodicity) : the number of observations used to compute the composite means and standard-deviations stored in the NetCDF variables *netcdf_variable_compmean* and *netcdf_variable_compsd* defined above.

All these statistics are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Example

- 1) For computing a composite analysis from a fourdimensional NetCDF variable *votemper* in the NetCDF file *ST7_1m_00101_20012_votemper_grid_T.nc*, assessing the significance of the results with a monte carlo simulation with 999 shuffles and, finally, storing the results in a NetCDF file named *composite_ST7_1m_votemper_grid_T.nc*, use the following command (note that the critical probabilities associated with the U statistics are estimated with the help of a resampling method using 999 surrogate time series since **-alg=simul** and **-nb=999** are specified) :

```
$ comp_composite_4d \
-f=ST7_1m_00101_20012_T_votemper_grid_T.nc \
-v=votemper \
-c=clim_votemper_grid_T.nc \
-a=10,11,20,55,100 \
-g=t \
-m=mesh_mask_ST7_votemper_grid_T.nc \
-alg=simul \
-nb=999 \
-o=composite_ST7_1m_votemper_grid_T.nc
```

2.7 comp_composite_miss_3d

2.7.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.7.2 Latest revision

30/04/2026

2.7.3 Purpose

Compute a composite analysis from a tridimensional variable with missing values extracted from a NetCDF dataset and perform statistical tests on the differences between the composite mean and the overall mean (e.g., the mean of the parent finite population) for each cell of the 2-D grid-mesh associated with the tridimensional NetCDF variable. The composite statistics and their associated critical probabilities may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

The use of the classical Student's two sample t-test [[vonStorch_Zwiers](#)] for statistical inference in composite analyses, as is traditionally used in the climate literature, is not valid [[Brown_Hall](#)]. Thus, to obtain more meaningful results, `comp_composite_miss_3d` uses the theory of random sampling without replacement in the finite population of the observed time observations and a normal approximation to assess the statistical significance of the composite means [[Terray_etal](#)]. Note, however, that resampling techniques can not be used in `comp_composite_miss_3d` due to the presence of missing values in the data. This is in contrast with `comp_composite_3d` in which the user can choose between a resampling scheme and the normal approximation method to estimate the statistical significance of the composite means.

For a much more comprehensive description of the composite method and the statistical test used in `comp_composite_miss_3d` to assess the statistical significance of the composite means, consult the documentation of `comp_composite_3d`, which is more extensive, and the references cited below [[Brown_Hall](#)] [[Terray_etal](#)].

If your data does not contain missing values, use `comp_composite_3d` instead of `comp_composite_miss_3d` to estimate the statistics of the composite analysis from your dataset.

2.7.4 Further Details

Usage

```
$ comp_composite_miss_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-c=input_climatology_netcdf_file \
-a=year1,year2,year3, ... , yearn \
-m=input_mesh_mask_netcdf_file      (optional) \
-g=grid_type                        (optional : n, t, u, v, w, f) \
-x=lon1,lon2                        (optional) \
-y=lat1,lat2                        (optional) \
-t=time1,time2                      (optional) \
-s=type_of_statistics               (optional : comp, diff) \
-o=output_composite_netcdf_file     (optional) \
-mi=missing_value                   (optional) \
-nostd                              (optional) \
-double                             (optional) \
-bigfile                            (optional) \
-hdf5                               (optional) \
-cdf5                               (optional) \
-tlimited                            (optional)
```

By default

- m=** an `output_mesh_mask_netcdf_file` is not used
- g=** the `grid_type` is set to `n` which means that the 2-D grid-mesh associated with the input `netcdf_variable` is assumed to be regular or Gaussian

- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- s=** the *type_of_statistics* is set to `comp`
- o=** the *output_composite_netcdf_file* is named `composite_netcdf_variable.nc`
- mi=** the *missing_value* attribute in the output NetCDF file is set to `1.e+20`
- nostd** the composite fields of the input NetCDF variable are standardized. If **-nostd** is activated, the composite fields are not standardized
- double** the composite statistics are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=***netcdf_variable* argument specifies the NetCDF variable for which a composite analysis must be computed and the **-f=***input_netcdf_file* argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The *input_climatology_netcdf_file* specified with the **-c=** argument must contain the means and standard deviations for the parent population. The periodicity for the composite analysis is also deduced from the number of observations in the *input_climatology_netcdf_file*. This *input_climatology_netcdf_file* must have been created by *comp_clim_miss_3d* applied to the same *netcdf_variable* with an identical **-t=time1,time2** argument.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the climatology (in the *input_climatology_netcdf_file*) must agree.

- 3) The **-a=** argument lists the indices of the years (or seasons, months or days depending on the sampling of the observations in the *input_netcdf_file*), which must be included in the composite analysis (e.g., for computing the composite means). The indices of the years are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing).

The list may be specified in different formats:

- **-a=n1,n2,...nn** allows to select for years *n1*, *n2*, ... and *nn* in the *input_netcdf_file*
- **-a=n1:n2** allows to select for years *n1* to *n2* in the *input_netcdf_file*

The two forms of the **-a=** argument may be combined and repeated any number of times, but duplicate years are not allowed.

Note, however, that the number of years really used to compute the composite means may vary from one cell to another in the 2-D grid-mesh associated with the input tridimensional NetCDF variable because of the presence of missing values.

Remember also, when specifying the indices of the years in the **-a=** argument, that the “length” of a year in the *input_netcdf_file* is determined by the periodicity of the observations as deduced from the number of time observations and periodicity in the *input_climatology_netcdf_file*.

This periodicity will also determine how many composite means $\mathbf{M}(j)$ and statistics $\mathbf{U}(j)$ will be computed and stored in the *output_composite_netcdf_file* by `comp_composite_miss_3d`.

See the documentation of `comp_composite_3d` for more details on the definition of the $\mathbf{U}(j)$ statistic.

- 4) The optional argument **-m=***input_mesh_mask_netcdf_file* specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before computing the composite analysis. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g., some non-missing values are not present in each time series).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the climatology (in the *input_climatology_netcdf_file*) and the mask (if an *input_mesh_mask_netcdf_file* is used) must agree.

Refer to `comp_clim_miss_3d` or `comp_mask_3d` for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using `comp_composite_miss_3d`.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determined the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.
- 6) If the **-x=***lon1,lon2* and **-y=***lat1,lat2* arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used in the composite analysis.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where `nlon` is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to `comp_mask_3d` for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using `comp_composite_miss_3d`.

- 7) If the **-t=***time1,time2* argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

It is also assumed that the selected time period matches exactly the time period used to compute the climatology in the *input_climatology_netcdf_file*. Moreover, the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.

- 8) The **-s=***type_of_statistics* argument specified how the composite variable *netcdf_variable_composite* stored in the *output_composite_netcdf_file* is computed:
 - **-s=**`comp` means that the composites are computed as the standardized differences between the mean of the composite years and the overall mean (e.g., the mean of the parent finite population) for each grid-point.
 - **-s=**`diff` means that the composites are computed as the standardized differences between the mean of the composite years and the mean of the other years in the parent finite population for each grid-point.

The standard deviations from the parent population are used for the standardization in both cases and are read from the *input_climatology_netcdf_file*. Finally, note that this argument does not affect the other variables stored in the *output_composite_netcdf_file*.

- 9) The **-nostd** argument specifies that the composite variable *netcdf_variable_composite* in the *output_composite_netcdf_file* must not be standardized.

By default, the composites are standardized in the *output_composite_netcdf_file*.

- 10) It is assumed that the specified *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the value of this *missing_value* or *_FillValue* attribute.
- 11) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_composite_netcdf_file*.

If the **-mi=** argument is not specified the *missing_value* attribute is set to $1.e+20$ for the NetCDF variables in the *output_composite_netcdf_file*.
- 12) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 13) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 14) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 15) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 16) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 17) For more details on composite analysis and statistical testing in composite analysis, see
 - “The Use of t values in Composite Analyses” by Brown, T.J., and Hall, B.L., Journal of climate, vol. 12, 2941-2944, 1999. doi: [10.1175/1520-0442\(1999\)012<2941:TUOTVI>2.0.CO;2](https://doi.org/10.1175/1520-0442(1999)012<2941:TUOTVI>2.0.CO;2)
 - “Sea Surface Temperature associations with the Late Indian Summer Monsoon”, by Terray, P., Delecluse P., Labattu S., Terray L., Climate Dynamics, vol. 21, 593-618, 2003. doi: [10.1007/s00382-003-0354-0](https://doi.org/10.1007/s00382-003-0354-0)
 - “Computer-intensive methods for testing hypotheses: an introduction”, by Noreen, E.W., Wiley and Sons, New York, USA, 1989. ISBN: [978-0-471-61136-3](https://www.wiley.com/9780471611363)
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](https://www.cambridge.org/9780521012300)

Outputs

`comp_composite_miss_3d` creates an output NetCDF file that contains the composite statistics and critical probabilities associated with the composite means, taking into account the missing values and, eventually, the periodicity of the data as determined by the **-c=input_climatology_netcdf_file** argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, *nlat* and *nlon* are

the length of the spatial dimensions of the input NetCDF variable) and *periodicity* time observations (*periodicity* is the number of time observations in the *input_climatology_netcdf_file*) :

- 1) *netcdf_variable_compmean*(*periodicity*,*nlat*,*nlon*) : the mean fields $\mathbf{M}(j)$ of the time observations selected with the help of the **-a=** argument, taking into account the missing values and the periodicity of the data as determined by the number of observations in the *input_climatology_netcdf_file* (specified with the help of the **-c=** argument).
- 2) *netcdf_variable_compstd*(*periodicity*,*nlat*,*nlon*) : the standard-deviation fields of the time observations selected with the help of the **-a=** argument, taking into account the missing values and the periodicity of the data as determined by the **-c=input_climatology_netcdf_file** argument. This corresponds to the standard-deviation of the years belonging to class $C(j)$.
- 3) *netcdf_variable_qstd*(*periodicity*,*nlat*,*nlon*) : the ratio between the standard deviations fields of the selected time observations to the standard deviations fields of all the time observations, taking into account the periodicity of the data as determined by the **-c=input_climatology_netcdf_file** argument. The standard deviations fields for all the observations are extracted from the *input_climatology_netcdf_file* specified in the **-c=** argument.
- 4) *netcdf_variable_composite*(*periodicity*,*nlat*,*nlon*) : the composite fields of the time observations selected with the help of the **-a=** argument, taking into account the missing values and the periodicity of the data as determined by the number of observations in the *input_climatology_netcdf_file* (specified with the help of the **-c=** argument).

How these composite fields are calculated is determined by the **-s=** and **-nostd** arguments.

- 5) *netcdf_variable_u*(*periodicity*,*nlat*,*nlon*) : the $\mathbf{U}(j)$ statistics for the time observations selected with the help of the **-a=** argument, taking into account the missing values and the periodicity of the data as determined by the number of observations in the *input_climatology_netcdf_file* (specified with the help of the **-c=** argument).

See the documentation of [comp_composite_3d](#) for more details on the definition of the $\mathbf{U}(j)$ statistic.

- 6) *netcdf_variable_prob*(*periodicity*,*nlat*,*nlon*) : the critical probabilities associated with the \mathbf{U} statistics.

These critical probabilities are computed under the null hypothesis that the selected time observations (with the help of the **-a=** argument) come from the same population as the other observations in the *input_netcdf_file*. Small probabilities indicate a large departure from the null hypothesis $\mathbf{H0}$.

These critical probabilities are computed with a Gaussian approximation, see the documentation of [comp_composite_3d](#) for more details.

- 7) *netcdf_variable_compnobs*(*periodicity*,*nlat*,*nlon*) : the number of observations used to compute the composite means and standard-deviations for each grid cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable.

Note that some of these statistics and the associated critical probabilities may be missing for some grid-cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable depending on the pattern of missing values in this input NetCDF variable.

All these statistics are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Example

- 1) For computing a composite analysis from a tridimensional NetCDF variable *sst* in the NetCDF file *Hadisst_1m_195001_201512_sst.nc*, assessing the significance of the re-

sults with a Gaussian approximation and, finally, storing the results in a NetCDF file named `composite_Hadisst_1m_195001_201512_sst.nc`, use the following command :

```
$ comp_composite_miss_3d \
-f=Hadisst_1m_195001_201512_sst.nc \
-v=sst \
-c=clim_Hadisst_1m_195001_201512_sst.nc \
-a=10,11,20,55,143 \
-m=mesh_mask_Hadisst.nc \
-o=composite_Hadisst_1m_195001_201512_sst.nc
```

2.8 comp_cor_1d

2.8.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.8.2 Latest revision

29/04/2026

2.8.3 Purpose

Compute correlation and regression coefficients between an index time series and an unidimensional variable extracted from NetCDF datasets and perform statistical tests on these correlation coefficients.

If $X(:)$ and $Y(:)$ are the vectors of *ntime* observations corresponding, respectively, to the index time series and the input unidimensional NetCDF variable, the (Pearson) correlation coefficient of the two variables X and Y is defined by

$$\mathbf{COR} = \mathbf{COV}(X, Y) / [\mathbf{STD}(X) \cdot \mathbf{STD}(Y)]$$

where

- $\mathbf{COV}(X, Y)$ is the covariance between X and Y (e.g., `dot_product(X(:) - MEAN(X) , Y(:) - MEAN(Y))/ntime`)
- $\mathbf{MEAN}(X)$ and $\mathbf{MEAN}(Y)$ are the means of X and Y , respectively.
- $\mathbf{STD}(X)$ and $\mathbf{STD}(Y)$ are the standard deviations of X and Y , respectively.

From this definition, it follows that the correlation is a pure number invariant under changes of scale and origin of the variables X and Y . Furthermore, it can be shown that \mathbf{COR} cannot be less than -1 or greater than 1 .

The Pearson correlation coefficient can be used to assess if the relationship between two variables is more or less linear (e.g., if the two variables are “proportional” to each other). More precisely, $\text{abs}(\mathbf{COR}) = 1$ if and only if there exists a linear relationship between X and Y , e.g., if

$$X(i) = a \cdot Y(i) + b, \text{ for } i = 1 \text{ to } ntime, \text{ with } a > 0 \text{ if } \mathbf{COR} = 1 \text{ and } a < 0 \text{ if } \mathbf{COR} = -1.$$

If the correlation coefficient is high in absolute value, but not equal to 1 or -1 , it can also be “summarized” by a straight line (sloped upwards or downwards) which is called the regression line between X and Y . This line is also a least squares line, because it can be determined such that the sum of the squared distances of all the data points from the line is the lowest possible in the scatter plot of X and Y .

The coefficients a and b of the regression line are called, respectively, the regression and intercept coefficients for predicting the dependent variable X by the independent variable Y . These coefficients are computed by `comp_cor_1d`

(intercept coefficients are computed only if the argument *-intercept* is specified). Furthermore, the regression line for predicting the index time series *Y* by the *X* variable (e.g., the role of the two variables is interchanged) can also be estimated if the optional argument **-rg=reg2** is specified when calling `comp_cor_1d`.

As mentioned before, the correlation coefficient **COR** represents the linear relationship between two variables. If this correlation coefficient is squared, then the resulting value (the coefficient of determination) will represent the proportion of common variation (or shared variance) between the two variables (i.e., the “strength” of the relationship). More precisely, this value gives you the percentage of variance of the dependent variable *X* explained by the independent variable *Y*. In order to evaluate the correlation between variables, it is important to know this “strength” as well as the significance of the correlation which can be assessed with the help of statistical tests.

If *X* and *Y* are independently distributed, their covariance, and hence their correlation, is zero, but the converse is not generally true. However, in the case of *X* and *Y* follow a bivariate normal distribution, the nullity of the correlation coefficient implies the independence of the variables *X* and *Y*.

Furthermore, in the case of a random sample of *ntime* observation pairs from such bivariate normal population with a zero correlation coefficient, the distribution of the variate

$$T = \text{COR} \cdot \sqrt{(ntime - 2) / [1 - \text{COR} \cdot \text{COR}]}$$

has a Student-Fisher *t* distribution with *ntime* - 2 degrees of freedom (call it *t[ntime-2]* in what follows) [[vonStorch_Zwiers](#)].

Given the sample correlation **COR**, we can thus test the hypothesis of no correlation in the bivariate normal parent population with the help of the Student-Fisher *t* distribution [[vonStorch_Zwiers](#)]. Monte Carlo simulations suggest that this test remains valid if the couple *X* and *Y* does not follow a bivariate normal distribution and the number of observations is big enough (e.g., *ntime* > 30); but in this case it is not a test of the independence of the two variables *X* and *Y*.

If the correlation coefficient in the parent distribution is not assumed to be zero, the distribution of **COR** has a complicated form. In that case, for example for computing confidence intervals for the correlation coefficient or testing if the parent correlation is some number different of zero, it is better to use the monotonic transformation

$$Z = (1/2) \cdot \log([1 + \text{COR}] / [1 - \text{COR}])$$

, which is called the Fisher *z* transformation and is the inverse of the hyperbolic tangent function. The Fisher *z* transform produces an asymptotically normal variate with variance equal to $1/(ntime - 3)$ [[vonStorch_Zwiers](#)].

The critical probabilities associated with the correlation coefficients are estimated by

$$\text{PROB} = P(|t[ntime-2]| > |T|)$$

if the argument **-a=student** is specified when calling the procedure (this is the default value for this argument).

If your sample of *ntime* observation pairs cannot be assumed to be normal, **PROB** can also be estimated by permutation methods (e.g., by using the argument **-a=permute** when calling `comp_cor_1d`). More precisely, in the case of *ntime* independent and identically distributed observations from a bivariate population of unknown form, we may consider the permutation of *Y* coordinates to test for the hypothesis of no correlation in the parent population. Let *S*(*X*, *Y*) be the set of points obtained by permuting the coordinates of *Y*(:) in all *ntime*! possible ways. Then, there is no correlation between *X* and *Y* in each element of *S*(*X*, *Y*) since any permutation uniformly makes sets of uncorrelated data. Hence, by randomly permuting the order of the elements of the *Y*(:) vector and recomputing the correlation coefficient between *X*(:) and this permuted vector many times (as determined by the **-nb=number_of_shuffles** argument), we can estimate the permutation distribution of $\text{abs}(\text{COR})$, conditionally on the *Y*(:) vector, and compute critical probabilities **PROB** useful for testing the hypothesis of no correlation in the parent population (e.g., by counting the number of times the permuted values of $\text{abs}(\text{COR})$ exceed the magnitude of the original correlation in the observed bivariate sample).

If your sample of *ntime* observation pairs cannot be assumed to be a random bivariate sample and the observations are auto-correlated in time, bootstrap procedures both in the time or frequency domains are available for estimating the critical probabilities, **PROB**, associated with the correlations coefficients [[Davison_Hinkley](#)].

If the argument `-a=bootstrap` a blockwise bootstrap procedure in the time domain is used to estimate the critical probabilities, **PROB**. This is useful when the observations are serially correlated. In this algorithm, we consider the population of subsamples or overlapping blocks of length `bootstrap_block_length` formed from the time observations in the $Y(:)$ vector. These overlapping blocks form a finite set defined by

$$\mathbf{BLK}(i) = Y(i + 1 : i + \text{bootstrap_block_length}) \text{ for } i = 0 \text{ to } ntime - \text{bootstrap_block_length}$$

Blockwise bootstrap is then realized by resampling randomly the blocks $\mathbf{BLK}(i)$ and gluing them together to form a kind of surrogate time series of length `ntime`. Finally, the correlation coefficient between $X(:)$ and this surrogate time series is computed. This procedure is iterated many times, as determined by the `-nb=number_of_shuffles` argument, to estimate the bootstrap distribution of `abs(COR)` and compute critical probabilities **PROB** useful for testing the hypothesis of no correlation in the parent population (e.g., by counting the number of times the bootstrap values of `abs(COR)` exceed the magnitude of the original correlation in the observed sample). The `-bp=`, `-bs=` and `-bl=` arguments allow the user to determine the exact form of the blockwise bootstrap algorithm. The `-bp=` argument is particularly useful if your time series are cyclostationary since it forces all the blocks to start at specific observations which are the same day, month or season. The `-bl=` argument allows the user to choose the size of the blocks. See the remarks below for more details.

If the argument `-a=` is set to `theiler` or `scramble`, a frequency bootstrap procedure is used to generate independent surrogate time series with the same spectral characteristics as the original time series. The basic idea of these methods is to create simulated series by manipulating the Discrete Fourier Transform (DFT) of a given time series instead of resampling randomly blocks of this series in the time domain. There are many different ways for how to do these manipulations in the literature. Two of these methods are currently implemented in `comp_cor_1d`, the Theiler method [[Theiler_etal](#)] [[Ebisuzaki](#)] and the Davison and Hinkley method [[Davison_Hinkley](#)] [[Braun_Kulperger](#)]. Both methods assume that the input time series are stationary (e.g., they do not contain pure harmonic components such as a seasonal cycle or a well-defined trend).

The Theiler method (used when the argument `-a=` is set to `theiler` when calling `comp_cor_1d`) consists of randomly shifting the phases in the DFT of the $Y(:)$ vector and back-transforms it to obtain a bootstrap sample in the time domain. As an illustration, assuming that `ntime` is odd and the DFT of $Y(:)$ is the complex vector $Z(:)$, then define the complex vector $O(:)$ of `ntime` elements as

- $O(1) = 1$
- $O(k) = \exp(i \cdot 2 \cdot \pi \cdot U(k))$ for $k = 2$ to $\lfloor ntime + 1 \rfloor / 2$
- $O(k) = -O(ntime - k)$ for $k = \lfloor ntime + 1 \rfloor / 2 + 1$ to `ntime`

, where $i = \sqrt{-1}$ and the $U(k)$ are a random sample drawn from an uniform distribution on $[0, 1]$. Then, the Theiler surrogate series is obtained by multiplying element by element the complex vectors $Z(:)$ and $O(:)$ and, finally, taking the inverse DFT of this new complex vector to obtain a bootstrap sample in the time domain. By construction, this simulated series is real-valued, independent of the vector $X(:)$, but its sample mean and periodogram are identical to those of the vector $Y(:)$.

The Davison and Hinkley approach (used when the argument `-a=` is set to `scramble` when calling `comp_cor_1d`) amounts also to a randomization of the phases of the Fourier coefficients of the DFT of $Y(:)$, but also includes an additional step that modifies the amplitudes of these Fourier coefficients. Let, again, the vector $U(:)$ be a random sample of `ntime` observations drawn from an uniform distribution on $[0, 1]$, and, define, the complex vector $O(:)$ as

- $O(1) = 0$
- $O(k) = Z(k) \cdot \exp(i \cdot 2 \cdot \pi \cdot U(k))$ for $k = 2$ to `ntime`

and form the complex vector $A(:)$ as

- $A(1) = Z(1)$
- $A(k) = \sqrt{0.5} \cdot [O(k) + \text{conjg}(O(ntime - k))]$ for $k = 2$ to `ntime`

Then, the Davison and Hinkley surrogate series is obtained by taking the inverse DFT of the complex vector $A(:)$. Again, by construction, this simulated series is real-valued, independent of the vector $X(:)$, with a sample mean iden-

tical to that of the vector $Y(\cdot)$, but, now its periodogram will be different from that of the vector $Y(\cdot)$. Note, however, that the mean spectrum obtained by averaging the periodograms of many Davison and Hinkley surrogate series will tend on average to the periodogram of the vector $Y(\cdot)$.

Finally, Davison and Hinkley [Davison_Hinkley] also discuss how to generate surrogate datasets with non-Gaussian distributions. Their approach can be used both in the context of the Theiler and, Davison and Hinkley methods, and is used to derived critical probabilities if the argument **-a=** is set to `theiler2` or `scramble2`. These variations of the original methods are useful when the observations are serially correlated and have also an asymmetric marginal distribution.

For more details on both the Theiler or Davison and Hinkley surrogate methods, consult the references cited below.

By default, `comp_cor_1d` computes the sample correlation and regression coefficients, the associated critical probabilities for testing the nullity of the correlation coefficients and the z transforms of the correlation coefficients between the index time series and the time series associated with the input unidimensional NetCDF variable. The intercept coefficient of the regression line between X and Y is also computed if the optional argument **-intercept** is specified when calling `comp_cor_1d`.

Moreover, all these statistics may be computed by taking into account the periodicity of the input tridimensional NetCDF variable if you suspect that the time series are cyclostationary (by using the **-p=periodicity** argument when calling the procedure). All the results are finally stored in an output NetCDF dataset.

Finally, if your input NetCDF variable is three or fourdimensional use `comp_cor_3d` or `comp_cor_4d` instead of `comp_cor_1d`.

This procedure is parallelized if OpenMP is used. Moreover, this procedure computes the correlation and regression coefficients with only one pass through the data and an out-of-core strategy which is highly efficient.

2.8.4 Further Details

Usage

```
$ comp_cor_1d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -vi=index_netcdf_variable \
  -fi=input_index_netcdf_file          (optional) \
  -t=time1,time2                      (optional) \
  -p=periodicity                      (optional) \
  -nv=indice_for_2d_netcdf_variable   (optional) \
  -a=type_of_analysis                 (optional : student, permute, bootstrap, \
                                     theiler, theiler2, \
                                     scramble, scramble2) \

  -rg=type_of_regression               (optional : reg1, reg2) \
  -o=output_netcdf_file               (optional) \
  -ti=itime1,itime2                   (optional) \
  -pi=iperiodicity,istep               (optional) \
  -ni=indice_for_2d_index_netcdf_variable (optional) \
  -nb=number_of_shuffles               (optional) \
  -bp=bootstrap_periodicity            (optional) \
  -bs=bootstrap_season                 (optional) \
  -bl=bootstrap_block_length           (optional) \
  -sm=smoothing_factor                (optional) \
  -mi=missing_value                   (optional) \
  -regstd                             (optional) \
  -intercept                          (optional) \
  -double                             (optional) \
```

(continues on next page)

(continued from previous page)

-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- fi=** the same as the **-f=** argument
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1
- nv=** if the *netcdf_variable* is bidimensional, the first time series is used
- ti=** the whole time period associated with the *index_netcdf_variable*
- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- a=** *type_of_analysis* is set to `student`
- rg=** *type_of_regression* is set to `reg1`
- nb=** *number_of_shuffles* is set to 99
- bp=** this parameter is set 1
- bs=** this parameter is not used
- bl=** the *bootstrap_block_length* is set 1
- sm=** no smoothing is applied to the *index_netcdf_variable*
- mi=** the *missing_value* is set to `1.e+20` in the *output_netcdf_file*
- o=** *output_netcdf_file* name is set to `cor_netcdf_variable.index_netcdf_variable.nc`
- regstd** the regression coefficients are computed in units of the input NetCDF variables. If **-regstd** is activated, the regression coefficients are computed in units of the *netcdf_variable* by standard-deviation of the *index_netcdf_variable*
- intercept** the intercept coefficients of the regressions are not computed. If **-intercept** is activated, the intercept coefficients of the regressions are computed and stored in the *output_netcdf_file*
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a correlation analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.

- 2) The **-nv=** specifies the indice (e.g., an integer) for selecting the time series if the input NetCDF variable *netcdf_variable* specified in the **-v=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_netcdf_variable* to 1.
- 3) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., *time2 - time1 + 1*) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.

- 4) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc.

Note that the output NetCDF file will have *periodicity* time observations.

- 5) The **-vi=index_netcdf_variable** specifies a time series for the correlation analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.
- 6) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_index_netcdf_variable* to 1.
- 7) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 8) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute correlations with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.
- 9) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(\text{time2} - \text{time1} + 1) / \text{periodicity} = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$

otherwise, an error message will be issued and the program will stop.

- 10) The **-a=** argument selects the method for computing critical probabilities associated with the correlation coefficients.
 - If **-a=student**, a classical Student-Fisher t test is used.
 - If **-a=permute**, a permutation test is used.
 - If **-a=bootstrap**, a moving block bootstrap test is used.
 - If **-a=theiler**, a phase-scrambled bootstrap (Theiler method) test is used.
 - If **-a=theiler2**, a phase-scrambled bootstrap (Theiler method) test is used, but when phase-scrambling the index time series exact empirical margins are used instead of normal margins as in the *theiler* option.
 - If **-a=scramble**, a phase-scrambled bootstrap (Davison-Hinkley method) test is used.
 - If **-a=scramble2**, a phase-scrambled bootstrap (Davison-Hinkley method) test is used, but when phase-scrambling the index time series exact empirical margins are used instead of normal margins as in the *scramble* option.

- 11) The **-nb=number_of_shuffles** argument specifies the number of shuffles for the phase-scrambled, bootstrap or permutation tests if **-a=permute, bootstrap, theiler, theiler2, scramble or scramble2**.
- 12) The **-bp=bootstrap_periodicity** argument specifies that the indice, i , of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where j is a random positive integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. By default, *bootstrap_periodicity* is set to 1.
- 13) The **-bs=bootstrap_season** argument specifies that the input time series is a repetition of the same season for different years and *bootstrap_season* specifies the length of the season. *bootstrap_season* must be greater than zero and the length of the time series must be a multiple of *bootstrap_season*. If the optional argument *bootstrap_periodicity* is used, *bootstrap_season* must also be greater or equal to *bootstrap_periodicity*. By default, *bootstrap_season* is set to the length of the time series.
- 14) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater than zero and less than the length of the time series. If the optional argument *bootstrap_periodicity* is used, *bootstrap_block_length* must also be greater or equal to *bootstrap_periodicity*. Moreover, if the optional argument *bootstrap_season* is used, *bootstrap_block_length* must also be less than *bootstrap_season*. By default, *bootstrap_block_length* is set to 1 or to *bootstrap_periodicity* if this optional argument is used.
- 15) The **-rg=** argument selects the method for computing the regression coefficients:
 - If **-rg=reg1**, the coefficients of the regression equation for predicting the *netcdf_variable* by the *index_netcdf_variable* are computed. This is the default.
 - If **-rg=reg2**, the coefficients of the regression equation for predicting the *index_netcdf_variable* by the *netcdf_variable* are computed.
- 16) The **-intercept** argument species that the intercept coefficients of the regression equation must be computed and stored in the output NetCDF file. By default, the intercept coefficients are not computed.
- 17) The **-regstd** argument specifies that the regression coefficients of the regression equation must be expressed in terms of units of the input NetCDF variable by standard-deviation of the *index_netcdf_variable*. By default, the regression coefficients are expressed in units of the input NetCDF variables.
- 18) **-sm=smoothing_factor** means that the time series associated with the *index_netcdf_variable* (e.g., the **-vi=** argument) must be smoothed with a moving average of approximately $2 \cdot \text{smoothing_factor} + 1$ terms before computing the correlations with the *netcdf_variable* (e.g., the **-v=** argument). *smoothing_factor* must be a strictly positive integer (>zero).
- 19) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variables* in the *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1 \cdot e+20$.
- 20) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file. By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built

with the `_USE_NETCDF44` CPP macro.

- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) It is assumed that the data has no missing values.
- 25) For more details on correlation and regression analysis in the climate literature, see
 - “Statistical Analysis in Climate Research”, by H. von Storch and F.W. Zwiers, Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: [9780521012300](#)

For more details on frequency or time series bootstrap procedures, see

- “Testing for nonlinearity in time series: the method of surrogate data.” by Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., and Farmer, J.D. Physica D, vol. 58, 77-94, 1992. doi: [10.1016/0167-2789\(92\)90102-s](#)
- “A method to estimate the statistical significance of a correlation when the data are serially correlated”, by Ebisuzaki, W., Journal of climate, vol. 10, 2147-2153, 1997. doi: [10.1175/1520-0442\(1997\)010<2147:AMTETS>2.0.CO;2](#)
- “Properties of a fourier bootstrap method for time series”, by Braun, W.J., and Kulperger, R.J., Communications in Statistics - Theory and Methods, vol 26, 1329-1336, 1997. doi: [10.1080/03610929708831985](#)
- “Bootstrap methods and their application”, by Davison, A.C., and Hinkley, D.V., Cambridge Univesity press, Cambridge, UK, 1997. doi: [10.1017/CBO9780511802843](#)

Outputs

`comp_cor_1d` creates an output NetCDF file that contains the correlation and regression statistics and critical probabilities associated with these coefficients, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables and *periodicity* time observations, if `-rg=reg1`:

- 1) `netcdf_variable_index_netcdf_variable_cor(periodicity)` : the Pearson correlation coefficients between the `netcdf_variable` and `index_netcdf_variable` time series.
- 2) `netcdf_variable_index_netcdf_variable_prob(periodicity)` : the critical probabilities associated with two-sided tests of the correlation coefficients (e.g., the absolute value of the correlation is tested). These critical probabilities are computed under the null hypothesis that the corresponding correlation coefficients in the parent population are zero.
- 3) `netcdf_variable_index_netcdf_variable_z(periodicity)` : the Fisher z Transforms of the correlation coefficients.
- 4) `netcdf_variable_index_netcdf_variable_reg(periodicity)` : the regression coefficients for predicting the time series associated with the input NetCDF variable by the `index_netcdf_variable` time series.

By default, the regression coefficients are expressed in units of the input NetCDF variable `netcdf_variable` by unit of the `index_netcdf_variable` time series. However, if the `-regstd` argument is specified the regression coefficients are expressed in terms of units of the input NetCDF variable `netcdf_variable` by standard-deviation of the `index_netcdf_variable` time series. Finally, if `-rg=reg2` is specified the roles of the input NetCDF variables `netcdf_variable` and `index_netcdf_variable` are interchanged and the fitted regression models are for predicting the `index_netcdf_variable` by the time series of the input NetCDF variable `netcdf_variable`.

- 5) `netcdf_variable_index_netcdf_variable_int(periodicity)` : the intercept coefficients in the regression models for predicting the time series associated with the input NetCDF variable by the `index_netcdf_variable` time series.

This variable is stored only if the **-intercept** argument has been specified when calling `comp_cor_1d`. Finally, if **-rg=reg2** is specified the roles of the input NetCDF variables *netcdf_variable* and *index_netcdf_variable* are interchanged and the fitted regression models are for predicting the *index_netcdf_variable* by each time series of the 2-D grid-mesh associated with the input NetCDF variable *netcdf_variable*.

- 6) *netcdf_variable_index_netcdf_variable_nobs*(periodicity) : the number of observations used to compute the correlation and regression coefficients.

If **-rg=reg2**, the naming convention for the variables is reversed, the *index_netcdf_variable* will be listed first and the *netcdf_variable* will appear after. For example, the name of the NetCDF variable storing the correlation coefficient will be *index_netcdf_variable_netcdf_variable_cor* instead of *netcdf_variable_index_netcdf_variable_cor* if **-rg=reg2**.

Examples

- 1) For computing lead-lag correlations between a bimonthly SST time series stored in a NetCDF variable *sst_nino34* in the NetCDF file *sst_2m_sst_nino34_1950_2000.nc* and a JJAS rainfall time series for India stored in the NetCDF variable *precip_india* in the NetCDF file *iitm.precip_1950_2000.nc* and write the results in a NetCDF file named *cor_2m_sst_nino34_precip_india_1950_2000.nc*, use the following commands (note that the critical probabilities associated with the correlations are estimated with the help of the scramble method using 999 surrogate time series and cyclostationarity is assumed for the *sst_nino34* variable since **-p=6** is specified) :

```
$ comp_cor_1d \
-f=sst_2m_sst_nino34_1950_2000.nc \
-v=sst_nino34 \
-p=6 \
-fi=iitm.precip_1950_2000.nc \
-vi=precip_india \
-a=scramble \
-nb=999 \
-o=cor_2m_sst_nino34_precip_india_1950_2000.nc
```

2.9 comp_cor_3d

2.9.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.9.2 Latest revision

29/04/2026

2.9.3 Purpose

Compute correlation and regression coefficients between an index time series and a tridimensional variable extracted from NetCDF datasets and perform statistical tests on these correlation and regression coefficients.

The procedure first transforms the input tridimensional NetCDF variable as a *ntime* by *nv* rectangular matrix of observed variables stored columnwise (e.g., the selected cells of the 2-D grid-mesh associated with the tridimensional

NetCDF variable) and then computes measures of association between each of these variables, say X , and the input index time series, say Y .

If $X(:)$ and $Y(:)$ are vectors of $ntime$ observations, the (Pearson) correlation coefficient of the two variables X and Y is defined by

$$\mathbf{COR} = \mathbf{COV}(X, Y) / [\mathbf{STD}(X) \cdot \mathbf{STD}(Y)]$$

where

- $\mathbf{COV}(X, Y)$ is the covariance between X and Y (e.g., `dot_product(X(:) - MEAN(X) , Y(:) - MEAN(Y))/ntime`)
- $\mathbf{MEAN}(X)$ and $\mathbf{MEAN}(Y)$ are the means of X and Y , respectively, as computed by `comp_clim_3d`.
- $\mathbf{STD}(X)$ and $\mathbf{STD}(Y)$ are the standard deviations of X and Y , respectively, as computed by `comp_clim_3d`.

From this definition, it follows that the correlation is a pure number invariant under changes of scale and origin of the variables X and Y . Furthermore, it can be shown that \mathbf{COR} cannot be less than -1 or greater than 1 .

The Pearson correlation coefficient can be used to assess if the relationship between two variables is more or less linear (e.g., if the two variables are “proportional” to each other). More precisely, $\text{abs}(\mathbf{COR}) = 1$ if and only if there exists a linear relationship between X and Y , e.g., if

$$X(i) = a \cdot Y(i) + b, \text{ for } i = 1 \text{ to } ntime, \text{ with } a > 0 \text{ if } \mathbf{COR} = 1 \text{ and } a < 0 \text{ if } \mathbf{COR} = -1.$$

If the correlation coefficient is high in absolute value, but not equal to 1 or -1 , it can also be “summarized” by a straight line (sloped upwards or downwards), which is called the regression line between X and Y . This line is also a least squares line, because it can be determined such that the sum of the squared distances of all the data points from the line is the lowest possible in the scatter plot of X and Y .

The coefficients a and b of the regression line are called, respectively, the regression and intercept coefficients for predicting the dependent variable X by the independent variable Y . These coefficients are computed by `comp_cor_3d` (intercept coefficients are computed only if the argument `-intercept` is specified). Furthermore, the regression lines for predicting the index time series Y from each of the X variables from the 2-D grid-mesh associated with the tridimensional NetCDF variable can also be estimated if the optional argument `-rg=reg2` is specified.

As mentioned before, the correlation coefficient \mathbf{COR} represents the linear relationship between two variables. If this correlation coefficient is squared, then the resulting value (the coefficient of determination) will represent the proportion of common variation (or shared variance) between the two variables (i.e., the “strength” of the relationship). More precisely, this value gives you the percentage of variance of the dependent variable X explained by the independent variable Y . In order to evaluate the correlation between two variables, it is important to know this “strength” as well as the significance of the correlation which can be assessed with the help of statistical tests.

If X and Y are independently distributed, their covariance, and hence their correlation, is zero, but the converse is not generally true. However, in the case of X and Y follow a bivariate normal distribution, the nullity of the correlation coefficient implies the independence of the variables X and Y .

Furthermore, in the case of a random sample of $ntime$ observation pairs from such bivariate normal population with a zero correlation coefficient, the distribution of the variate

$$\mathbf{T} = \mathbf{COR} \cdot \sqrt{(ntime - 2) / (1 - \mathbf{COR} \cdot \mathbf{COR})}$$

has a Student-Fisher t distribution with $ntime - 2$ degrees of freedom (call it $t[ntime-2]$ in what follows) [[vonStorch_Zwiers](#)].

Given the sample correlation \mathbf{COR} , we can thus test the hypothesis of no correlation in the bivariate normal parent population with the help of the Student-Fisher t distribution [[vonStorch_Zwiers](#)]. Monte Carlo simulations suggest that this test remains valid if the couple X and Y does not follow a bivariate normal distribution and the number of observations is big enough (e.g., $ntime > 30$); but in this case it is not a test of the independence of the two variables X and Y .

If the correlation coefficient in the parent distribution is not assumed to be zero, the distribution of **COR** has a complicated form. In that case, for example for computing confidence intervals for the correlation coefficient or testing if the parent correlation is some number different of zero, it is better to use the monotonic transformation

$$Z = (1/2) \cdot \log([1 + \text{COR}] / [1 - \text{COR}])$$

, which is called the Fisher z transformation and is the inverse of the hyperbolic tangent function. The Fisher z transform produces an asymptotically normal variate with variance equal to $1/(n\text{time} - 3)$ [vonStorch_Zwiers].

The critical probabilities associated with the correlation coefficients are estimated by

$$\text{PROB} = P(\text{abs}(t[n\text{time}-2]) > \text{abs}(T))$$

if the argument **-a=student** is specified when calling the procedure (this is the default value for this argument).

If your sample of *n*time observation pairs cannot be assumed to be normal, **PROB** can also be estimated by permutation methods (e.g., by using the argument **-a=permute** when calling `comp_cor_3d`). More precisely, in the case of *n*time independent and identically distributed observations from a bivariate population of unknown form, we may consider the permutation of *Y* coordinates to test for the hypothesis of no correlation in the parent population. Let $S(X, Y)$ be the set of points obtained by permuting the coordinates of $Y(\cdot)$ in all *n*time! possible ways. Then, there is no correlation between *X* and *Y* in each element of $S(X, Y)$ since any permutation uniformly makes sets of uncorrelated data. Hence, by randomly permuting the order of the elements of the $Y(\cdot)$ vector and recomputing the correlation coefficient between $X(\cdot)$ and this permuted vector many times (as determined by the **-nb=number_of_shuffles** argument), we can estimate the permutation distribution of $\text{abs}(\text{COR})$, conditionally on the $Y(\cdot)$ vector, and compute critical probabilities **PROB** useful for testing the hypothesis of no correlation in the parent population (e.g., by counting the number of times the permuted values of $\text{abs}(\text{COR})$ exceed the magnitude of the original correlation in the observed bivariate sample).

If your sample of *n*time observation pairs cannot be assumed to be a random bivariate sample and the observations are auto-correlated in time, bootstrap procedures both in the time or frequency domains are available for estimating the critical probabilities, **PROB**, associated with the correlations coefficients [Davison_Hinkley].

If the argument **-a=bootstrap** a blockwise bootstrap procedure in the time domain is used to estimate the critical probabilities, **PROB**. This is useful when the observations are serially correlated. In this algorithm, we consider the population of subsamples or overlapping blocks of length *bootstrap_block_length* formed from the time observations in the $Y(\cdot)$ vector. These overlapping blocks form a finite set defined by

$$\text{BLK}(i) = Y(i + 1 : i + \text{bootstrap_block_length}) \text{ for } i = 0 \text{ to } n\text{time} - \text{bootstrap_block_length}$$

Blockwise bootstrap is then realized by resampling randomly a sufficient number of blocks $\text{BLK}(i)$ and gluing them together to form a kind of surrogate time series of length *n*time. Finally, the correlation coefficient between $X(\cdot)$ and this surrogate time series is computed. This procedure is iterated many times, as determined by the **-nb=number_of_shuffles** argument, to estimate the bootstrap distribution of $\text{abs}(\text{COR})$ and compute critical probabilities **PROB** useful for testing the hypothesis of no correlation in the parent population (e.g., by counting the number of times the bootstrap values of $\text{abs}(\text{COR})$ exceed the magnitude of the original correlation in the observed sample). The **-bp=**, **-bs=** and **-bl=** arguments allow the user to determine the exact form of the blockwise bootstrap algorithm. The **-bp=** argument is particularly useful if your time series are cyclostationary since it forces all the blocks to start at specific observations which are the same day, month or season. The **-bl=** argument allows the user to choose the size of the blocks. See the remarks below for more details.

If the argument **-a=** is set to `theiler` or `scramble`, a frequency bootstrap procedure is used to generate independent surrogate time series with the same spectral characteristics as the original time series. The basic idea of these methods is to create simulated series by manipulating the Discrete Fourier Transform (DFT) of a given time series instead of resampling randomly blocks of this series in the time domain. There are many different ways for how to do these manipulations in the literature. Two of these methods are currently implemented in `comp_cor_3d`, the Theiler method [Theiler_etal] [Ebisuzaki] and the Davison and Hinkley method [Davison_Hinkley] [Braun_Kulperger]. Both methods assume that the input time series are stationary (e.g., they do not contain pure harmonic components such as a seasonal cycle or a well-defined trend).

The Theiler method (used when the argument `-a=` is set to `theiler` when calling `comp_cor_3d`) consists of randomly shifting the phases in the DFT of the $Y(\cdot)$ vector and back-transforms it to obtain a bootstrap sample in the time domain. As an illustration, assuming that $ntime$ is odd and the DFT of $Y(\cdot)$ is the complex vector $Z(\cdot)$, then define the complex vector $O(\cdot)$ of $ntime$ elements as

- $O(1) = 1$
- $O(k) = \exp(i \cdot 2 \cdot \pi \cdot U(k))$ for $k = 2$ to $\lfloor ntime + 1 \rfloor / 2$
- $O(k) = -O(ntime - k)$ for $k = \lfloor ntime + 1 \rfloor / 2 + 1$ to $ntime$

, where $i = \sqrt{-1}$ and the $U(k)$ are a random sample drawn from an uniform distribution on $[0, 1]$. Then, the Theiler surrogate series is obtained by multiplying element by element the complex vectors $Z(\cdot)$ and $O(\cdot)$ and, finally, taking the inverse DFT of this new complex vector to obtain a bootstrap sample in the time domain. By construction, this simulated series is real-valued, independent of the vector $X(\cdot)$, but its sample mean and periodogram are identical to those of the vector $Y(\cdot)$.

The Davison and Hinkley approach (used when the argument `-a=` is set to `scramble` when calling `comp_cor_3d`) amounts also to a randomization of the phases of the Fourier coefficients of the DFT of $Y(\cdot)$, but also includes an additional step that modifies the amplitudes of these Fourier coefficients. Let, again, the vector $U(\cdot)$ be a random sample of $ntime$ observations drawn from an uniform distribution on $[0, 1]$, and, define, the complex vector $O(\cdot)$ as

- $O(1) = 0$
- $O(k) = Z(k) \cdot \exp(i \cdot 2 \cdot \pi \cdot U(k))$ for $k = 2$ to $ntime$

and form the complex vector $A(\cdot)$ as

- $A(1) = Z(1)$
- $A(k) = \sqrt{0.5} \cdot [O(k) + \text{conj}(O(ntime - k))]$ for $k = 2$ to $ntime$

Then, the Davison and Hinkley surrogate series is obtained by taking the inverse DFT of the complex vector $A(\cdot)$. Again, by construction, this simulated series is real-valued, independent of the vector $X(\cdot)$, with a sample mean identical to that of the vector $Y(\cdot)$, but, now its periodogram will be different from that of the vector $Y(\cdot)$. Note, however, that the mean spectrum obtained by averaging the periodograms of many Davison and Hinkley surrogate series will tend on average to the periodogram of the vector $Y(\cdot)$.

Finally, Davison and Hinkley [[Davison_Hinkley](#)] also discuss how to generate surrogate datasets with non-Gaussian distributions. Their approach can be used both in the context of the Theiler and, Davison and Hinkley methods, and is used to derived critical probabilities if the argument `-a=` is set to `theiler2` or `scramble2`. These variations of the original methods are useful when the observations are serially correlated and have also an asymmetric marginal distribution.

For more details on both the Theiler or Davison and Hinkley surrogate methods, consult the references cited below.

By default, `comp_cor_3d` computes the sample correlation and regression coefficients, the associated critical probabilities for testing the nullity of the correlation coefficients and the z transforms of the correlation coefficients between the index time series and each point in the time series of the 2-D grid-mesh associated with the input tridimensional NetCDF variable. The intercept coefficients of the regression lines between X and Y are also computed if the optional argument `-intercept` is specified when calling `comp_cor_3d`.

Moreover, all these statistics may be computed by taking into account the periodicity of the input tridimensional NetCDF variable if you suspect that the time series are cyclostationary (by using the `-p=periodicity` argument when calling the procedure). All the results are finally stored in an output NetCDF dataset, after repacking the statistics on the original 2-D grid of the input tridimensional NetCDF variable.

If your data contains missing values use `comp_cor_miss_3d` instead of `comp_cor_3d` to estimate correlation and regression coefficients from your gappy dataset. Finally, if the NetCDF variable is fourdimensional use `comp_cor_4d` instead of `comp_cor_3d`.

This procedure is parallelized if OpenMP is used. Moreover, this procedure computes the correlation and regression coefficients with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.9.4 Further Details

Usage

```
$ comp_cor_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-vi=index_netcdf_variable \
-fi=input_index_netcdf_file           (optional) \
-m=input_mesh_mask_netcdf_file       (optional) \
-g=grid_type                         (optional : n, t, u, v, w, f) \
-x=lon1,lon2                         (optional) \
-y=lat1,lat2                         (optional) \
-t=time1,time2                      (optional) \
-p=periodicity                      (optional) \
-a=type_of_analysis                  (optional : student, permute, bootstrap, \
                                     theiler, theiler2, \
                                     scramble, scramble2) \

-rg=type_of_regression               (optional : reg1, reg2) \
-o=output_netcdf_file               (optional) \
-ti=itime1,itime2                   (optional) \
-pi=iperiodicity,istep              (optional) \
-ni=indice_for_2d_index_netcdf_variable (optional) \
-nb=number_of_shuffles              (optional) \
-bp=bootstrap_periodicity           (optional) \
-bs=bootstrap_season                (optional) \
-bl=bootstrap_block_length          (optional) \
-sm=smoothing_factor                (optional) \
-mi=missing_value                   (optional) \
-regstd                             (optional) \
-intercept                          (optional) \
-double                             (optional) \
-bigfile                            (optional) \
-hdf5                               (optional) \
-cdf5                               (optional) \
-tlimited                            (optional)
```

By default

- fi=** the same as the **-f=** argument
- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1
- ti=** the whole time period associated with the *index_netcdf_variable*

- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- a=** *type_of_analysis* is set to `student`
- rg=** *type_of_regression* is set to `reg1`
- nb=** *number_of_shuffles* is set to 99
- bp=** this parameter is set 1
- bs=** this parameter is not used
- bl=** the *bootstrap_block_length* is set 1
- sm=** no smoothing is applied to the *index_netcdf_variable*
- mi=** the *missing_value* is set to `1.e+20` in the *output_netcdf_file*
- o=** *output_netcdf_file* name is set to `cor_netcdf_variable.index_netcdf_variable.nc`
- regstd** the regression coefficients are computed in units of the input NetCDF variables. If **-regstd** is activated, the regression coefficients are computed in units of the *netcdf_variable* by standard-deviation of the *index_netcdf_variable*
- intercept** the intercept coefficients of the regressions are not computed. If **-intercept** is activated, the intercept coefficients of the regressions are computed and stored in the *output_netcdf_file*
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a correlation analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before computing the correlation analysis. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_3d](#) or [comp_mask_3d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using `comp_cor_3d`.

- 3) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determined the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_cor_3d*.

- 5) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., $time2 - time1 + 1$) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.

- 6) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc.

Note that the output NetCDF file will have *periodicity* time observations.

- 7) The **-vi=index_netcdf_variable** specifies a time series for the correlation analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

- 8) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_index_netcdf_variable* to 1.

- 9) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

- 10) The **-pi=** argument gives the periodicity and select the time step for the *index_netcdf_variable*. For example, to compute correlations with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.

- 11) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(time2 - time1 + 1) / periodicity = \text{ceiling}((itime2 - itime1 - istep + 2) / iperiodicity),$$

otherwise, an error message will be issued and the program will stop.

- 12) The **-a=** argument selects the method for computing critical probabilities associated with the correlation coefficients:

- If **-a=student**, a classical Student-Fisher t test is used.
- If **-a=permute**, a permutation test is used.
- If **-a=bootstrap**, a moving block bootstrap test is used.
- If **-a=theiler**, a phase-scrambled bootstrap (Theiler method) test is used.

- If **-a=theiler2**, a phase-scrambled bootstrap (Theiler method) test is used, but when phase-scrambling the index time series exact empirical margins are used instead of normal margins as in the `theiler` option.
 - If **-a=scramble**, a phase-scrambled bootstrap (Davison-Hinkley method) test is used.
 - If **-a=scramble2**, a phase-scrambled bootstrap (Davison-Hinkley method) test is used, but when phase-scrambling the index time series exact empirical margins are used instead of normal margins as in the `scramble` option.
- 13) The **-nb=number_of_shuffles** argument specifies the number of shuffles for the phase-scrambled, bootstrap or permutation tests if **-a=permute,bootstrap,theiler,theiler2,scramble** or `scramble2`.
The default value is 99.
- 14) The **-bp=bootstrap_periodicity** argument specifies that the indice, i , of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where j is a random positive (or null) integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. By default, *bootstrap_periodicity* is set to 1.
- 15) The **-bs=bootstrap_season** argument specifies that the input time series is a repetition of the same season for different years and *bootstrap_season* specifies the length of the season. *bootstrap_season* must be greater than zero and the length of the time series must be a multiple of *bootstrap_season*. If the optional argument *bootstrap_periodicity* is used, *bootstrap_season* must also be greater or equal to *bootstrap_periodicity*. By default, *bootstrap_season* is set to the length of the time series.
- 16) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater than zero and less than the length of the time series. If the optional argument *bootstrap_periodicity* is used, *bootstrap_block_length* must also be greater or equal to *bootstrap_periodicity*. Moreover, if the optional argument *bootstrap_season* is used, *bootstrap_block_length* must also be less than *bootstrap_season*. By default, *bootstrap_block_length* is set to 1 or to *bootstrap_periodicity* if this optional argument is used.
- 17) The **-rg=** argument selects the method for computing the regression coefficients:
- If **-rg=reg1**, the coefficients of the regression equation for predicting the *netcdf_variable* by the *index_netcdf_variable* are computed. This is the default.
 - If **-rg=reg2**, the coefficients of the regression equation for predicting the *index_netcdf_variable* by the *netcdf_variable* are computed.
- 18) The **-intercept** argument specifies that the intercept coefficients of the regression equation must be computed and stored in the output NetCDF file. By default, the intercept coefficients are not computed.
- 19) The **-regstd** argument specifies that the regression coefficients of the regression equation must be expressed in terms of units of the input NetCDF variable by standard-deviation of the *index_netcdf_variable*. By default, the regression coefficients are expressed in units of the input NetCDF variables.
- 20) **-sm=smoothing_factor** means that the time series associated with the *index_netcdf_variable* (e.g., the **-vi=** argument) must be smoothed with a moving average of approximately $2 \cdot \text{smoothing_factor} + 1$ terms before computing the correlations with the *netcdf_variable* (e.g., the **-v=** argument). *smoothing_factor* must be a strictly positive integer (>zero).
- 21) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variables* in the *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1 \cdot e+20$.
- 22) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file. By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 23) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 24) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 25) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 26) It is assumed that the data has no missing values. If it is the case, use *comp_cor_miss_3d* instead of *comp_cor_3d*.

- 27) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 28) For more details on correlation and regression analysis in the climate literature, see

- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: [9780521012300](#)

For more details on frequency or time series bootstrap procedures, see

- “Testing for nonlinearity in time series: the method of surrogate data.” by Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., and Farmer, J.D. Physica D, vol. 58, 77-94, 1992. doi: [10.1016/0167-2789\(92\)90102-s](#)
- “A method to estimate the statistical significance of a correlation when the data are serially correlated”, by Ebisuzaki, W., Journal of climate, vol. 10, 2147-2153, 1997. doi: [10.1175/1520-0442\(1997\)010<2147:AMTETS>2.0.CO;2](#)
- “Properties of a fourier bootstrap method for time series”, by Braun, W.J., and Kulperger, R.J., Communications in Statistics - Theory and Methods, vol 26, 1329-1336, 1997. doi: [10.1080/03610929708831985](#)
- “Bootstrap methods and their application”, by Davison, A.C., and Hinkley, D.V., Cambridge Univesity press, Cambridge, UK, 1997. doi: [10.1017/CBO9780511802843](#)

Outputs

comp_cor_3d creates an output NetCDF file that contains the correlation and regression statistics and critical probabilities associated with these coefficients, taking into account eventually the periodicity of the data as determined by the **-p=periodicity** argument. If **-rg=reg1**, the output NetCDF dataset contains the following NetCDF variables (in the description below, *nlat* and *nlon* are the length of the spatial dimensions of the input NetCDF variable) and *periodicity* time observations :

- 1) *netcdf_variable_index_netcdf_variable_cor*(*periodicity*, *nlat*, *nlon*) : the Pearson correlation coefficients between each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable and the *index_netcdf_variable* time series.

- 2) *netcdf_variable_index_netcdf_variable_prob*(periodicity,nlat,nlon) : the critical probabilities associated with two-sided tests of the correlation coefficients (e.g., the absolute value of the correlation is tested). These critical probabilities are computed under the null hypothesis that the corresponding correlation coefficients in the parent population are zero.

The **-a=type_of_analysis** argument determines how these critical probabilities are computed.

- 3) *netcdf_variable_index_netcdf_variable_z*(periodicity,nlat,nlon) : the Fisher z Transforms of the correlation coefficients for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable and the *index_netcdf_variable* time series.
- 4) *netcdf_variable_index_netcdf_variable_reg*(periodicity,nlat,nlon) : the regression coefficients for predicting each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable by the *index_netcdf_variable* time series.

By default, the regression coefficients are expressed in units of the input NetCDF variable *netcdf_variable* by unit of the *index_netcdf_variable* time series. However, if the **-regstd** argument is specified the regression coefficients are expressed in terms of units of the input NetCDF variable *netcdf_variable* by standard-deviation of the *index_netcdf_variable* time series. Finally, if **-rg=reg2** is specified the roles of the input NetCDF variables *netcdf_variable* and *index_netcdf_variable* are interchanged and the fitted regression models are for predicting the *index_netcdf_variable* by each time series of the 2-D grid-mesh associated with the input NetCDF variable *netcdf_variable*.

- 5) *netcdf_variable_index_netcdf_variable_int*(periodicity,nlat,nlon) : the intercept coefficients in the regression models for predicting each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable by the *index_netcdf_variable* time series.

This variable is stored only if the **-intercept** argument has been specified when calling *comp_cor_3d*. Finally, if **-rg=reg2** is specified the roles of the input NetCDF variables *netcdf_variable* and *index_netcdf_variable* are interchanged and the fitted regression models are for predicting the *index_netcdf_variable* by each time series of the 2-D grid-mesh associated with the input NetCDF variable *netcdf_variable*.

- 6) *netcdf_variable_index_netcdf_variable_nobs*(periodicity) : the number of observations used to compute the correlation and regression coefficients.

All these statistics, excepted the *netcdf_variable_index_netcdf_variable_nobs* variable, are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

If **-rg=reg2**, the naming convention for the variables is reversed, the *index_netcdf_variable* will be listed first and the *netcdf_variable* will appear after. For example, the name of the NetCDF variable storing the correlation coefficient will be *index_netcdf_variable_netcdf_variable_cor* instead of *netcdf_variable_index_netcdf_variable_cor* if **-rg=reg2**.

Examples

- 1) For computing monthly lead correlations from a tridimensional NetCDF variable *sosstsst* in the NetCDF file *ST7_1m_00101_20012_grid_T_sosstsst.nc* and a December-January Nino34 SST index in the NetCDF file *ST7_sst_nino34_dj.nc* and store the results in a NetCDF file named *cor_ST7_1m_sosstsst_nino34_dj_grid_T.nc*, use the following command (note that the critical probabilities associated with the correlations are estimated with the help of the Theiler method using 999 surrogate time series and cyclostationarity is assumed for the *sosstsst* variable since **-p=12** is specified):

```
$ comp_cor_3d \
-f=ST7_1m_00101_20012_grid_T_sosstsst.nc \
-v=sosstsst \
-m=mesh_mask_ST7.nc \
-p=12 \
-fi=sst_nino34_dj.nc \
-vi=sosstsst \
-a=theiler \
-nb=999 \
-o=cor_ST7_1m_sosstsst_nino34_dj_grid_T.nc
```

2.10 comp_cor_4d

2.10.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.10.2 Latest revision

29/04/2026

2.10.3 Purpose

Compute correlation and regression coefficients between an index time series and a fourdimensional variable extracted from a NetCDF dataset and perform statistical tests on these correlation and regression coefficients.

The procedure first transforms the input fourdimensional NetCDF variable as a *ntime* by *nv* rectangular matrix of observed variables stored columnwise (e.g., the selected cells of the 3-D grid-mesh associated with the fourdimensional NetCDF variable) and then computes measures of association between each of these variables, say *X*, and the input index time series, say *Y*.

If *X*(:) and *Y*(:) are vectors of *ntime* observations, the (Pearson) correlation coefficient of the two variables *X* and *Y* is defined by

$$\mathbf{COR} = \mathbf{COV}(X, Y) / [\mathbf{STD}(X) \cdot \mathbf{STD}(Y)]$$

where

- $\mathbf{COV}(X, Y)$ is the covariance between *X* and *Y* (e.g., `dot_product(X(:) - MEAN(X) , Y(:) - MEAN(Y))/ntime`)
- $\mathbf{MEAN}(X)$ and $\mathbf{MEAN}(Y)$ are the means of *X* and *Y*, respectively, as computed by [comp_clim_4d](#).
- $\mathbf{STD}(X)$ and $\mathbf{STD}(Y)$ are the standard deviations of *X* and *Y*, respectively, as computed by [comp_clim_4d](#).

From this definition, it follows that the correlation is a pure number invariant under changes of scale and origin of the variables *X* and *Y*. Furthermore, it can be shown that **COR** cannot be less than -1 or greater than 1.

The Pearson correlation coefficient can be used to assess if the relationship between two variables is more or less linear (e.g., if the two variables are “proportional” to each other). More precisely, $\text{abs}(\mathbf{COR}) = 1$ if and only if there exists a linear relationship between *X* and *Y*, e.g., if

$$X(i) = a \cdot Y(i) + b, \text{ for } i = 1 \text{ to } ntime, \text{ with } a > 0 \text{ if } \mathbf{COR} = 1 \text{ and } a < 0 \text{ if } \mathbf{COR} = -1.$$

If the correlation coefficient is high in absolute value, but not equal to 1 or -1, it can also be “summarized” by a straight line (sloped upwards or downwards), which is called the regression line between *X* and *Y*. This line is also a

least squares line, because it can be determined such that the sum of the squared distances of all the data points from the line is the lowest possible in the scatter plot of X and Y .

The coefficients a and b of the regression line are called, respectively, the regression and intercept coefficients for predicting the dependent variable X by the independent variable Y . These coefficients are computed by `comp_cor_4d` (intercept coefficients are computed only if the argument `-intercept` is specified). Furthermore, the regression lines for predicting the index time series Y from each of the X variables from the 3-D grid-mesh associated with the fourdimensional NetCDF variable can also be estimated if the optional argument `-rg=reg2` is specified.

As mentioned before, the correlation coefficient **COR** represents the linear relationship between two variables. If this correlation coefficient is squared, then the resulting value (the coefficient of determination) will represent the proportion of common variation (or shared variance) between the two variables (i.e., the “strength” of the relationship). More precisely, this value gives you the percentage of variance of the dependent variable X explained by the independent variable Y . In order to evaluate the correlation between variables, it is important to know this “strength” as well as the significance of the correlation which can be assessed with the help of statistical tests.

If X and Y are independently distributed, their covariance, and hence their correlation, is zero, but the converse is not generally true. However, in the case of X and Y follow a bivariate normal distribution, the nullity of the correlation coefficient implies the independence of the variables X and Y .

Furthermore, in the case of a random sample of $ntime$ observation pairs from such bivariate normal population with a zero correlation coefficient, the distribution of the variate

$$T = \text{COR}.\text{sqrt}([ntime - 2] / [1 - \text{COR}.\text{COR}])$$

has a Student-Fisher t distribution with $ntime - 2$ degrees of freedom (call it $t[ntime-2]$ in what follows) [[vonStorch_Zwiers](#)].

Given the sample correlation **COR**, we can thus test the hypothesis of no correlation in the bivariate normal parent population with the help of the Student-Fisher t distribution [[vonStorch_Zwiers](#)]. Monte Carlo simulations suggest that this test remains valid if the couple X and Y does not follow a bivariate normal distribution and the number of observations is big enough (e.g., $ntime > 30$); but in this case it is not a test of the independence of the two variables X and Y .

If the correlation coefficient in the parent distribution is not assumed to be zero, the distribution of **COR** has a complicated form. In that case, for example for computing confidence intervals for the correlation coefficient or testing if the parent correlation is some number different of zero, it is better to use the monotonic transformation

$$Z = (1/2).\text{log}([1 + \text{COR}] / [1 - \text{COR}])$$

, which is called the Fisher z transformation and is the inverse of the hyperbolic tangent function. The Fisher z transform produces an asymptotically normal variate with variance equal to $1/(ntime - 3)$ [[vonStorch_Zwiers](#)].

The critical probabilities associated with the correlation coefficients are estimated by

$$\text{PROB} = P(\text{abs}(t[ntime-2]) > \text{abs}(T))$$

if the argument `-a=student` is specified when calling the procedure (this is the default value for this argument).

If your sample of $ntime$ observation pairs cannot be assumed to be normal, **PROB** can also be estimated by permutation methods (e.g., by using the argument `-a=permute` when calling `comp_cor_4d`). More precisely, in the case of $ntime$ independent and identically distributed observations from a bivariate population of unknown form, we may consider the permutation of Y coordinates to test for the hypothesis of no correlation in the parent population. Let $S(X, Y)$ be the set of points obtained by permuting the coordinates of $Y(:)$ in all $ntime!$ possible ways. Then, there is no correlation between X and Y in each element of $S(X, Y)$ since any permutation uniformly makes sets of uncorrelated data. Hence, by randomly permuting the order of the elements of the $Y(:)$ vector and recomputing the correlation coefficient between $X(:)$ and this permuted vector many times (as determined by the `-nb=number_of_shuffles` argument), we can estimate the permutation distribution of $\text{abs}(\text{COR})$, conditionally on the $Y(:)$ vector, and compute critical probabilities **PROB** useful for testing the hypothesis of no correlation in the parent population (e.g., by counting the number of times the permuted values of $\text{abs}(\text{COR})$ exceed the magnitude of the original correlation in the observed sample).

If your sample of *ntime* observation pairs cannot be assumed to be a random bivariate sample and the observations are auto-correlated in time, bootstrap procedures both in the time or frequency domains are available for estimating the critical probabilities, **PROB**, associated with the correlations coefficients [Davison_Hinkley] .

If the argument **-a=bootstrap** a blockwise bootstrap procedure in the time domain is used to estimate the critical probabilities, **PROB**. This is useful when the observations are serially correlated. In this algorithm, we consider the population of subsamples or overlapping blocks of length *bootstrap_block_length* formed from the time observations in the *Y(:)* vector. These overlapping blocks form a finite set defined by

BLK(*i*) = *Y*(*i* + 1 : *i* + *bootstrap_block_length*) for *i* = 0 to *ntime* - *bootstrap_block_length*

Blockwise bootstrap is then realized by resampling randomly the blocks **BLK**(*i*) and gluing them together to form a kind of surrogate time series of length *ntime*. Finally, the correlation coefficient between *X(:)* and this surrogate time series is computed. This procedure is iterated many times, as determined by the **-nb=number_of_shuffles** argument, to estimate the bootstrap distribution of **abs(COR)** and compute critical probabilities **PROB** useful for testing the hypothesis of no correlation in the parent population (e.g., by counting the number of times the bootstrap values of **abs(COR)** exceed the magnitude of the original correlation in the observed sample). The **-bp=**, **-bs=** and **-bl=** arguments allow the user to determine the exact form of the blockwise bootstrap algorithm. The **-bp=** argument is particularly useful if your time series are cyclostationary since it forces all the blocks to start at specific observations which are the same day, month or season. The **-bl=** argument allows the user to choose the size of the blocks. See the remarks below for more details.

If the argument **-a=** is set to *theiler* or *scramble*, a frequency bootstrap procedure is used to generate independent surrogate time series with the same spectral characteristics as the original time series. The basic idea of these methods is to create simulated series by manipulating the Discrete Fourier Transform (DFT) of a given time series instead of resampling randomly blocks of this series in the time domain. There are many different ways for how to do these manipulations in the literature. Two of these methods are currently implemented in *comp_cor_3d*, the Theiler method [Theiler_etal] [Ebisuzaki] and the Davison and Hinkley method [Davison_Hinkley] [Braun_Kulperger] . Both methods assume that the input time series are stationary (e.g., they do not contain pure harmonic components such as a seasonal cycle or a well-defined trend).

The Theiler method (used when the argument **-a=** is set to *theiler* when calling *comp_cor_4d*) consists of randomly shifting the phases in the DFT of the *Y(:)* vector and back-transforms it to obtain a bootstrap sample in the time domain. As an illustration, assuming that *ntime* is odd and the DFT of *Y(:)* is the complex vector *Z(:)*, then define the complex vector *O(:)* of *ntime* elements as

- $O(1) = 1$
- $O(k) = \exp(i \cdot 2 \cdot \pi \cdot U(k))$ for $k = 2$ to $[ntime + 1] / 2$
- $O(k) = -O(ntime - k)$ for $k = [ntime + 1] / 2 + 1$ to *ntime*

, where $i = \sqrt{-1}$ and the $U(k)$ are a random sample drawn from an uniform distribution on $[0, 1]$. Then, the Theiler surrogate series is obtained by multiplying element by element the complex vectors *Z(:)* and *O(:)* and, finally, taking the inverse DFT of this new complex vector to obtain a bootstrap sample in the time domain. By construction, this simulated series is real-valued, independent of the vector *X(:)*, but its sample mean and periodogram are identical to those of the vector *Y(:)*.

The Davison and Hinkley approach (used when the argument **-a=** is set to *scramble* when calling *comp_cor_4d*) amounts also to a randomization of the phases of the Fourier coefficients of the DFT of *Y(:)*, but also includes an additional step that modifies the amplitudes of these Fourier coefficients. Let, again, the vector *U(:)* be a random sample of *ntime* observations drawn from an uniform distribution on $[0, 1]$, and, define, the complex vector *O(:)* as

- $O(1) = 0$
- $O(k) = Z(k) \cdot \exp(i \cdot 2 \cdot \pi \cdot U(k))$ for $k = 2$ to *ntime*

and form the complex vector *A(:)* as

- $A(1) = Z(1)$

- $A(k) = \sqrt{0.5} \cdot [O(k) + \text{conj}(O(\text{ntime} - k))] \text{ for } k = 2 \text{ to } \text{ntime}$

Then, the Davison and Hinkley surrogate series is obtained by taking the inverse DFT of the complex vector $A(:)$. Again, by construction, this simulated series is real-valued, independent of the vector $X(:)$, with a sample mean identical to that of the vector $Y(:)$, but, now its periodogram will be different from that of the vector $Y(:)$. Note, however, that the mean spectrum obtained by averaging the periodograms of many Davison and Hinkley surrogate series will tend on average to the periodogram of the vector $Y(:)$.

Finally, Davison and Hinkley [Davison_Hinkley] also discuss how to generate surrogate datasets with non-Gaussian distributions. Their approach can be used both in the context of the Theiler and, Davison and Hinkley methods, and is used to derived critical probabilities if the argument **-a=** is set to `theiler2` or `scramble2`. These variations of the original methods are useful when the observations are serially correlated and have also an asymmetric marginal distribution.

For more details on both the Theiler or Davison and Hinkley surrogate methods, consult the references cited below.

By default, `comp_cor_4d` computes the sample correlation and regression coefficients, the associated critical probabilities for testing the nullity of the correlation coefficients and the z transforms of the correlation coefficients between the index time series and each point in the time series of the 3-D grid-mesh associated with the input fourdimensional NetCDF variable. The intercept coefficients of the regression lines between X and Y are also computed if the optional argument **-intercept** is specified when calling `comp_cor_4d`.

Moreover, these statistics may be computed by taking into account the periodicity of the input fourdimensional NetCDF variable if you suspect that the time series are cyclostationary (by using the **-p=periodicity** argument when calling the procedure). All the results are finally stored in an output NetCDF dataset, after repacking the statistics on the original 3-D grid of the input fourdimensional NetCDF variable.

Finally, if the NetCDF variable is tridimensional use `comp_cor_3d` instead of `comp_cor_4d`.

This procedure is parallelized if OpenMP is used. Moreover, this procedure computes the correlation and regression coefficients with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.10.4 Further Details

Usage

```
$ comp_cor_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -vi=index_netcdf_variable \
  -fi=input_index_netcdf_file           (optional) \
  -m=input_mesh_mask_netcdf_file       (optional) \
  -g=grid_type                         (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                         (optional) \
  -y=lat1,lat2                         (optional) \
  -z=level1,level2                     (optional) \
  -t=time1,time2                       (optional) \
  -p=periodicity                       (optional) \
  -a=type_of_analysis                  (optional : student, permute, bootstrap, \
                                         theiler, theiler2, \
                                         scramble, scramble2) \

  -rg=type_of_regression                (optional : reg1, reg2) \
  -o=output_netcdf_file                 (optional) \
  -ti=itime1,itime2                     (optional) \
  -pi=iperiodicity,istep                 (optional) \
  -ni=indice_for_2d_index_netcdf_variable (optional) \
  -nb=number_of_shuffles                 (optional) \
```

(continues on next page)

(continued from previous page)

-bp=bootstrap_periodicity	(optional) \
-bs=bootstrap_season	(optional) \
-bl=bootstrap_block_length	(optional) \
-sm=smoothing_factor	(optional) \
-mi=missing_value	(optional) \
-regstd	(optional) \
-intercept	(optional) \
-double	(optional) \
-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- fi=** the same as the **-f=** argument
- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 3-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1.
- ti=** the whole time period associated with the *index_netcdf_variable*
- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- a=** *type_of_analysis* is set to *student*
- rg=** *type_of_regression* is set to *reg1*
- nb=** *number_of_shuffles* is set to 99
- bp=** this parameter is set 1.
- bs=** this parameter is not used
- bl=** the *bootstrap_block_length* is set 1.
- sm=** no smoothing is applied to the *index_netcdf_variable*
- mi=** the *missing_value* is set to *1.e+20* in the *output_netcdf_file*
- o=** *output_netcdf_file* name is set to *cor_netcdf_variable.index_netcdf_variable.nc*
- regstd** the regression coefficients are computed in units of the input NetCDF variables. If **-regstd** is activated, the regression coefficients are computed in units of the *netcdf_variable* by standard-deviation of the *index_netcdf_variable*
- intercept** the intercept coefficients of the regressions are not computed. If **-intercept** is activated, the intercept coefficients of the regressions are computed and stored in the *output_netcdf_file*

- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a correlation analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this fourdimensional NetCDF variable as a rectangular matrix of observed variables before computing the correlation analysis. By default, it is assumed that each cell in the 3-D grid-mesh associated with the input fourdimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes and vertical resolution of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_4d](#) or [comp_mask_4d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_cor_4d*.

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determined the name of the *mesh_mask_variable* if an *input_mesh_mask_netcdf_file* is used.
- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_cor_4d*.

- 5) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., *time2 - time1 + 1*) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.

- 6) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc.

Note that the output NetCDF file will have *periodicity* time observations.

- 7) The **-vi=index_netcdf_variable** specifies a time series for the correlation analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the

NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

- 8) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_index_netcdf_variable* to 1.
- 9) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 10) The **-pi=** argument gives the periodicity and select the time step for the *index_netcdf_variable*. For example, to compute correlations with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.
- 11) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(\text{time2} - \text{time1} + 1) / \text{periodicity} = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$

otherwise, an error message will be issued and the program will stop.

- 12) The **-a=** argument selects the method for computing critical probabilities associated with the correlation coefficients:
 - If **-a=student**, a classical Student-Fisher t test is used.
 - If **-a=permute**, a permutation test is used.
 - If **-a=bootstrap**, a moving block bootstrap test is used.
 - If **-a=theiler**, a phase-scrambled bootstrap (Theiler method) test is used.
 - If **-a=theiler2**, a phase-scrambled bootstrap (Theiler method) test is used, but when phase-scrambling the index time series exact empirical margins are used instead of normal margins as in the `theiler` option.
 - If **-a=scramble**, a phase-scrambled bootstrap (Davison-Hinkley method) test is used.
 - If **-a=scramble2**, a phase-scrambled bootstrap (Davison-Hinkley method) test is used, but when phase-scrambling the index time series exact empirical margins are used instead of normal margins as in the `scramble` option.
- 13) The **-nb=number_of_shuffles** argument specifies the number of shuffles for the phase-scrambled, bootstrap or permutation tests if **-a=permute, bootstrap, theiler, theiler2, scramble** or `scramble2`.
The default value is 999.
- 14) The **-bp=bootstrap_periodicity** argument specifies that the indice, *i*, of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where *j* is a random positive integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. By default, *bootstrap_periodicity* is set to 1.
- 15) The **-bs=bootstrap_season** argument specifies that the input time series is a repetition of the same season for different years and *bootstrap_season* specifies the length of the season. *bootstrap_season* must be greater than zero and the length of the time series must be a multiple of *bootstrap_season*. If the optional argument *bootstrap_periodicity* is used, *bootstrap_season* must also be greater or equal to *bootstrap_periodicity*. By default, *bootstrap_season* is set to the length of the time series.

- 16) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater than zero and less than the length of the time series. If the optional argument *bootstrap_periodicity* is used, *bootstrap_block_length* must also be greater or equal to *bootstrap_periodicity*. Moreover, if the optional argument *bootstrap_season* is used, *bootstrap_block_length* must also be less than *bootstrap_season*. By default, *bootstrap_block_length* is set to 1 or to *bootstrap_periodicity* if this optional argument is used.
- 17) The **-rg=** argument selects the method for computing the regression coefficients:
 - If **-rg=reg1**, the coefficients of the regression equation for predicting the *netcdf_variable* by the *index_netcdf_variable* are computed. This is the default.
 - If **-rg=reg2**, the coefficients of the regression equation for predicting the *index_netcdf_variable* by the *netcdf_variable* are computed.
- 18) The **-intercept** argument specifies that the intercept coefficients of the regression equation must be computed and stored in the output NetCDF file. By default, the intercept coefficients are not computed.
- 19) The **-regstd** argument specifies that the regression coefficients of the regression equation must be expressed in terms of units of the input NetCDF variable by standard-deviation of the *index_netcdf_variable*. By default, the regression coefficients are expressed in units of the input NetCDF variables.
- 20) **-sm=smoothing_factor** means that the time series associated with the *index_netcdf_variable* (e.g., the **-vi=** argument) must be smoothed with a moving average of approximately $2 \cdot \text{smoothing_factor} + 1$ terms before computing the correlations with the *netcdf_variable* (e.g., the **-v=** argument). *smoothing_factor* must be a strictly positive integer (>zero).
- 21) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variables* in the *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1 \cdot e+20$.
- 22) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file. By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 23) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 24) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 25) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 26) It is assumed that the data has no missing values.
- 27) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

28) For more details on correlation and regression analysis in the climate literature, see

- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: [9780521012300](#)

For more details on frequency or time series bootstrap procedures, see

- “Testing for nonlinearity in time series: the method of surrogate data.” by Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., and Farmer, J.D. Physica D, vol. 58, 77-94, 1992. doi: [10.1016/0167-2789\(92\)90102-s](#)
- “A method to estimate the statistical significance of a correlation when the data are serially correlated”, by Ebisuzaki, W., Journal of climate, vol. 10, 2147-2153, 1997. doi: [10.1175/1520-0442\(1997\)010<2147:AMTETS>2.0.CO;2](#)
- “Properties of a fourier bootstrap method for time series”, by Braun, W.J., and Kulperger, R.J., Communications in Statistics - Theory and Methods, vol 26, 1329-1336, 1997. doi: [10.1080/03610929708831985](#)
- “Bootstrap methods and their application”, by Davison, A.C., and Hinkley, D.V., Cambridge Univesity press, Cambridge, UK, 1997. doi: [10.1017/CBO9780511802843](#)

Outputs

`comp_cor_4d` creates an output NetCDF file that contains the correlation and regression statistics and critical probabilities associated with these coefficients, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the length of the vertical and spatial dimensions of the input NetCDF variable) and *periodicity* time observations, if `-rg=reg1` :

- 1) `netcdf_variable_index_netcdf_variable_cor` (`periodicity`, `nlev`, `nlat`, `nlon`) : the Pearson correlation coefficients between each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable and the `index_netcdf_variable` time series.
- 2) `netcdf_variable_index_netcdf_variable_prob` (`periodicity`, `nlev`, `nlat`, `nlon`) : the critical probabilities associated with two-sided tests of the correlation coefficients (e.g., the absolute value of the correlation is tested). These critical probabilities are computed under the null hypothesis that the corresponding correlation coefficients in the parent population are zero.

The `-a=type_of_analysis` argument determines how these critical probabilities are computed.

- 3) `netcdf_variable_index_netcdf_variable_z` (`periodicity`, `nlev`, `nlat`, `nlon`) : the Fisher z Transforms of the correlation coefficients for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable and the `index_netcdf_variable` time series.
- 4) `netcdf_variable_index_netcdf_variable_reg` (`periodicity`, `nlev`, `nlat`, `nlon`) : the regression coefficients for predicting each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable by the `index_netcdf_variable` time series.

By default, the regression coefficients are expressed in units of the input NetCDF variable `netcdf_variable` by unit of the `index_netcdf_variable` time series. However, if the `-regstd` argument is specified the regression coefficients are expressed in terms of units of the input NetCDF variable `netcdf_variable` by standard-deviation of the `index_netcdf_variable` time series. Finally, if `-rg=reg2` is specified the roles of the input NetCDF variables `netcdf_variable` and `index_netcdf_variable` are interchanged and the fitted regression models are for predicting the `index_netcdf_variable` by each time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` .

- 5) `netcdf_variable_index_netcdf_variable_int` (`periodicity`, `nlev`, `nlat`, `nlon`) : the intercept coefficients in the regression models for predicting each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable by the `index_netcdf_variable` time series.

This variable is stored only if the **-intercept** argument has been specified when calling `comp_cor_4d`. Finally, if **-rg=reg2** is specified the roles of the input NetCDF variables *netcdf_variable* and *index_netcdf_variable* are interchanged and the fitted regression models are for predicting the *index_netcdf_variable* by each time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable*.

- 6) *netcdf_variable_index_netcdf_variable_nobs*(periodicity) : the number of observations used to compute the correlation and regression coefficients.

All these statistics, excepted the *netcdf_variable_index_netcdf_variable_nobs* variable, are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

If **-rg=reg2**, the naming convention for the variables is reversed, the *index_netcdf_variable* will be listed first and the *netcdf_variable* will appear after. For example, the name of the NetCDF variable storing the correlation coefficient will be *index_netcdf_variable_netcdf_variable_cor* instead of *netcdf_variable_index_netcdf_variable_cor* if **-rg=reg2**.

Examples

- 1) For computing monthly lead correlations from a fourdimensional NetCDF variable *votemper* in the NetCDF file *ST7_1m_00101_20012_grid_T_votemper.nc* and a December-January Nino34 SST index in the NetCDF file *ST7_sst_nino34_dj.nc* and store the results in a NetCDF file named *cor_ST7_1m_votemper_nino34_dj_grid_T.nc*, use the following commands (note that the critical probabilities associated with the correlations are estimated with the help of the Theiler method using 999 surrogate time series and cyclostationarity is assumed for the *sosstsst* variable since **-p=12** is specified) :

```
$ comp_cor_4d \  
-f=ST7_1m_00101_20012_grid_T_votemper.nc \  
-v=votemper \  
-m=mesh_mask_ST7.nc \  
-p=12 \  
-fi=sst_nino34_dj.nc \  
-vi=sosstsst \  
-a=theiler \  
-nb=999 \  
-o=cor_ST7_1m_votemper_nino34_dj_grid_T.nc
```

2.11 comp_cor_miss_3d

2.11.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.11.2 Latest revision

29/04/2026

2.11.3 Purpose

Compute correlation and regression coefficients between an index time series and a tridimensional variable extracted from a NetCDF dataset and perform statistical tests on these correlation coefficients. Missing values are allowed both in the input index time series and the input NetCDF tridimensional variable. However, if your data does not contain missing values excepted those associated with a constant land-sea mask use `comp_cor_3d` instead of `comp_cor_miss_3d` to estimate correlation and regression coefficients from your dataset.

As in `comp_cor_3d`, the procedure first transforms the input tridimensional NetCDF variable as a *ntime* by *nv* rectangular matrix of observed variables stored columnwise (e.g., the selected cells of the 2-D grid-mesh associated with the tridimensional NetCDF variable) and then computes measures of association between each of these variables, say *X*, and the input index time series, say *Y*. However since missing values are present, the number of observations used to compute the means and standard-deviations for each variable and the coefficients of correlation between each pair of variables *X* and *Y* may vary; This is an important difference with the statistics obtained from `comp_cor_3d`.

By default, `comp_cor_miss_3d` computes the sample correlation and regression coefficients, the associated critical probabilities for testing the nullity of the correlation coefficients and the *z* transforms of the correlation coefficients between the index time series and each point in the time series of the 2-D grid-mesh associated with the input tridimensional NetCDF variable. The intercept coefficients of the regression line between *X* and *Y* are also computed if the optional argument `-intercept` is specified when calling `comp_cor_miss_3d`. Moreover, all these statistics may be computed by taking into account the periodicity of the input tridimensional NetCDF variable if you suspect that the time series are cyclostationary (by using the `-p=periodicity` argument when calling the procedure). All the results are finally stored in an output NetCDF dataset, after repacking the statistics on the original 2-D grid of the input tridimensional NetCDF variable.

Refer to `comp_cor_3d`, for a basic definition of all these statistics, which is not repeated here. Refer to [vonStorch_Zwiers] for a general introduction on the correlation/regression coefficients and the *z* transform of the correlation coefficients and their use in climate analysis.

Due to the presence of missing values, two different methods are, however, available to estimate the correlation and regression coefficients (and the Fisher *z* transform) in `comp_cor_miss_3d`.

In the first method (used when the argument `-alg=` is set to `miss1` ; this is the default), `comp_cor_miss_3d` estimates first the means and standard-deviations for the index time series and each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable by using all the available observations for each time series. Since missing values are present, the number of observations used to compute the means and standard-deviations may then vary from one point to another in the 2-D grid-mesh associated with the input NetCDF variable (as well as for the index time series). In a second step, `comp_cor_miss_3d` estimates the correlation and regression coefficients using the previously computed univariate statistics and all valid pairs of observations for each couple of variables. From this definition, it follows that the correlation coefficients computed from this method may be greater than 1 or less than -1 in some cases when the number of missing values is very important. However, in such cases, the procedure adjusts the value of the correlation coefficients accordingly.

In the second method (used when the argument `-alg=` is set to `miss2`), `comp_cor_miss_3d` computes both the univariate and bivariate statistics from all valid pairs of observations for each couple of variables separately. From this definition, it follows that the estimated correlation coefficient cannot be less than -1 or greater than 1. However, the univariate statistics may be based on much fewer observations than in the first method (e.g., when `-alg=miss1`).

Finally, note that only one method is available for computing critical probabilities associated with the correlation coefficients and test the significance of the correlations, since the permutation and bootstrap methods cannot easily be implemented if missing values are present in the time series. This is in contrast to the variety of approaches available in `comp_cor_3d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the correlation and regression coefficients with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.11.4 Further Details

Usage

```
$ comp_cor_miss_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-vi=index_netcdf_variable \
-fi=input_index_netcdf_file           (optional) \
-m=input_mesh_mask_netcdf_file       (optional) \
-g=grid_type                         (optional : n, t, u, v, w, f) \
-x=lon1,lon2                         (optional) \
-y=lat1,lat2                         (optional) \
-t=time1,time2                       (optional) \
-p=periodicity                       (optional) \
-a=type_of_analysis                  (optional : student) \
-rg=type_of_regression               (optional : reg1, reg2) \
-o=output_netcdf_file                (optional) \
-ti=itime1,itime2                    (optional) \
-pi=iperiodicity,istep               (optional) \
-ni=indice_for_2d_index_netcdf_variable (optional) \
-mi=missing_value                    (optional) \
-alg=algorithm                       (optional : miss1, miss2) \
-regstd                              (optional) \
-intercept                           (optional) \
-double                              (optional) \
-bigfile                             (optional) \
-hdf5                                (optional) \
-cdf5                                 (optional) \
-tlimited                             (optional)
```

By default

- fi=** the same as the **-f=** argument
- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1
- ti=** the whole time period associated with the *index_netcdf_variable*
- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- a=** *type_of_analysis* is set to *student*
- rg=** *type_of_regression* is set to *reg1*
- mi=** the *missing_value* is set to *1.e+20* in the *output_netcdf_file*
- alg=** the method used to compute univariate and bivariate statistics is *miss1*

- o=** *output_netcdf_file* name is set to *cor_netcdf_variable.index_netcdf_variable.nc*
- regstd** the regression coefficients are computed in units of the input NetCDF variables. If **-regstd** is activated, the regression coefficients are computed in units of the *netcdf_variable* by standard-deviation of the *index_netcdf_variable*
- intercept** the intercept coefficients of the regressions are not computed. If **-intercept** is activated, the intercept coefficients of the regressions are computed and stored in the *output_netcdf_file*
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a correlation analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before computing the correlation analysis. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g., a time series with some valid data for at least some observations).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_miss_3d](#) or [comp_mask_3d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using [comp_cor_miss_3d](#).

- 3) If **-g=** is set to **t**, **u**, **v**, **w** or **f** it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determined the name of the *mesh_mask_variable* if an *input_mesh_mask_netcdf_file* is used.
- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_cor_miss_3d](#).

- 5) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., $time2 - time1 + 1$) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.

- 6) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc.

Note that the output NetCDF file will have *periodicity* time observations.

- 7) The **-vi=index_netcdf_variable** specifies a time series for the correlation analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.
- 8) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_index_netcdf_variable* to 1.
- 9) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 10) The **-pi=** argument gives the periodicity and select the time step for the *index_netcdf_variable*. For example, to compute correlations with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.
- 11) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(time2 - time1 + 1) / periodicity = \text{ceiling}((itime2 - itime1 - istep + 2) / iperiodicity),$$

otherwise, an error message will be issued and the program will stop.

- 12) The **-alg=** argument selects the method for computing the correlation coefficients:
- If **-alg=miss1**, the means and standard-deviations of the *netcdf_variable* and *index_netcdf_variable* are computed from all valid data. The correlation coefficients are based on these univariate statistics and on all valid pairs of observations.
 - If **-alg=miss2**, the univariate and bivariate statistics are computed from all valid pairs of observations for each couple of variables separately.
- 13) The **-a=** argument selects the method for computing critical probabilities associated with the correlation coefficients.
- If **-a=student**, a classical Student-Fisher t test is used.
- No other test options are included in this version of NCSTAT to test correlation coefficients from data with missing values, but this optional parameter is still present for later use.
- 14) The **-rg=** argument selects the method for computing the regression coefficients:
- If **-rg=reg1**, the coefficients of the regression equation for predicting the *netcdf_variable* by the *index_netcdf_variable* are computed. This is the default.
 - If **-rg=reg2**, the coefficients of the regression equation for predicting the *index_netcdf_variable* by the *netcdf_variable* are computed.
- 15) The **-intercept** argument specifies that the intercept coefficients of the regression equation must be computed and stored in the output NetCDF file. By default, the intercept coefficients are not computed.

- 16) The **-regstd** argument specifies that the regression coefficients of the regression equation must be expressed in terms of units of the input NetCDF variable *netcdf_variable* by standard-deviation of the *index_netcdf_variable*. By default, the regression coefficients are expressed in units of the input NetCDF variables.
- 17) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variables* in the *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to 1.e+20.
- 18) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file. By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 19) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 20) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 21) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 22) It is assumed that the specified *netcdf_variable* and *index_netcdf_variable* have a scalar missing or `_FillValue` attributes and that missing values in the data are identified by the values of these missing or `_FillValue` attributes.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on correlation and regression analysis in the climate literature, see
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: 9780521012300

Outputs

`comp_cor_miss_3d` creates an output NetCDF file that contains the correlation and regression statistics and critical probabilities associated with these coefficients, taking into account eventually the periodicity of the data as determined by the **-p=periodicity** argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, *nlat* and *nlon* are the length of the spatial dimensions of the input NetCDF variable) and *periodicity* time observations, if **-rg=reg1** :

- 1) *netcdf_variable_index_netcdf_variable_cor*(*periodicity*, *nlat*, *nlon*) : the Pearson correlation coefficients between each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable and the *index_netcdf_variable* time series.
- 2) *netcdf_variable_index_netcdf_variable_prob*(*periodicity*, *nlat*, *nlon*) : the critical probabilities associated with two-sided tests of the correlation coefficients (e.g., the absolute value of the

correlation is tested). These critical probabilities are computed under the null hypothesis that the corresponding correlation coefficients in the parent population are zero.

The `-a=type_of_analysis` argument determines how these critical probabilities are computed.

- 3) `netcdf_variable_index_netcdf_variable_z` (`periodicity,nlat,nlon`) : the Fisher z Transforms of the correlation coefficients for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable and the `index_netcdf_variable` time series.
- 4) `netcdf_variable_index_netcdf_variable_reg` (`periodicity,nlat,nlon`) : the regression coefficients for predicting each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable by the `index_netcdf_variable` time series.

By default, the regression coefficients are expressed in units of the input NetCDF variable `netcdf_variable` by unit of the `index_netcdf_variable` time series. However, if the `-regstd` argument is specified the regression coefficients are expressed in terms of units of the input NetCDF variable `netcdf_variable` by standard-deviation of the `index_netcdf_variable` time series. Finally, if `-rg=reg2` is specified the roles of the input NetCDF variables `netcdf_variable` and `index_netcdf_variable` are interchanged and the fitted regression models are for predicting the `index_netcdf_variable` by each time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable`.

- 5) `netcdf_variable_index_netcdf_variable_int` (`periodicity,nlat,nlon`) : the intercept coefficients in the regression models for predicting each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable by the `index_netcdf_variable` time series.

This variable is stored only if the `-intercept` argument has been specified when calling `comp_cor_miss_3d`. Finally, if `-rg=reg2` is specified the roles of the input NetCDF variables `netcdf_variable` and `index_netcdf_variable` are interchanged and the fitted regression models are for predicting the `index_netcdf_variable` by each time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable`.

- 6) `netcdf_variable_index_netcdf_variable_nobs` (`periodicity,nlat,nlon`) : the number of observations used to compute the correlation and regression coefficient for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

All these statistics are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

If `-rg=reg2`, the naming convention for the variables is reversed, the `index_netcdf_variable` will be listed first and the `netcdf_variable` will appear after. For example, the name of the NetCDF variable storing the correlation coefficient will be `index_netcdf_variable_netcdf_variable_cor` instead of `netcdf_variable_index_netcdf_variable_cor` if `-rg=reg2`.

Examples

- 1) For computing monthly lead correlations from a tridimensional NetCDF variable `sst` in the NetCDF file `HadISST2_sst.nc` and a December-January Nino34 SST index in the NetCDF file `HadISST2_sst_nino34_dj.nc` and store the results in a NetCDF file named `cor_HadISST2_1m_sst_nino34_dj.nc`, use the following commands (note that cyclostationarity is assumed for the `sst` variable since `-p=12` is specified) :

```
$ comp_cor_miss_3d \
-f=HadISST2_sst.nc \
-v=sst \
```

(continues on next page)

(continued from previous page)

```
-m=mesh_mask_HadISST2.nc \
-p=12 \
-fi=HadISST2_sst_nino34_dj.nc \
-vi=sst \
-o=cor_HadISST2_1m_sst_nino34_dj.nc
```

2.12 comp_cross_spectrum_1d

2.12.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.12.2 Latest revision

30/04/2026

2.12.3 Purpose

Compute Fast Fourier Transform (FFT) estimates of the power spectra and cross-power spectra between an index time series and an unidimensional variable readed from a NetCDF dataset. From the Power Spectral Density (PSD) and Cross-Spectral Density (CSD) estimates, this program also computes the phase and amplitude relationships between the signals, e.g., the phase of the cross-spectrum and the squared coherency estimate at each frequency. In output of `comp_cross_spectrum_1d`, both are given on the closed interval $[0., 1.]$, meaning that both are unitless.

In general terms, the phase gives the lag (negative values) or lead (positive values) of the index time series with respect to the variable time series at each frequency in radians. Here, the phase is given in fractions of a circle (e.g., on the closed interval $[0., 1.]$) instead of being in radians. In other words, for phase values less than 0.5 the index time series leads the variable time series and vice-versa for phase values greater than 0.5 .

The magnitude-squared-coherence can be used to identify significant frequency-domain correlation (at the Fourier frequencies) between the two time series. Phase estimates in the cross-spectrum are only useful where significant frequency domain correlation exists and you can neglect the cross spectrum when the squared coherency is small.

Spectral analysis amounts to estimate the power spectrum of a signal (e.g., a time series) from its time representation. The PSD estimates, which constitute the power spectrum, characterize the frequency content of the signal and decompose the variance of the signal into the different frequencies and allow to identify the periodicities present in the signal. In a similar way, cross-spectral analysis amounts to decompose the correlation between two signals into the different frequencies.

`comp_cross_spectrum_1d` is based on a non-parametric method, which divides the time series into segments (see the `-sl=` argument), applies FFT on each segment, computes the PSD and CSD estimates from the FFTs and, finally, averages these estimates from each segment. Such non-parametric method is based on the observed time series and does not require prior knowledge about the data or a model generating the data. This method is known as the Welch's averaged, modified periodogram method for estimating the power and cross-power spectra of different time series [Welch]. This Welch's Overlapped Segment Averaging (WOSA) method also provides a consistent average coherence estimator, which prevents obtaining a squared coherency estimate that is identically $1.$ for all frequencies.

At the user option, the different time series can be detrended before computing the PSD and CSD estimates (see the `-tr=` and `-tr2=` arguments) in order to ensure that the analyzed time series remain stationary.

In order to obtain stable PSD and CSD estimates (e.g., reduce the variance of the estimates) and minimize the loss of information, the real time series can also be segmented, and PSD and CSD estimates on, eventually overlapping,

segments can be averaged at the user option (this is known as the Welch's method as already noted above). The stability of the PSD and CSD estimates is increased by this averaging process because the Welch's method uses PSD and CSD estimates from different segments of the time series, which represent approximately uncorrelated estimates of the true PSD and CSD estimates. That is, the greater the number of segments, the more stable, the resulting PSD and CSD estimates as their variance is reduced by the averaging.

The user can also select different data windows for the computations of the PSD and CSD estimates (see the **-win=** argument) on each segment. The segments are multiplied by a data window (selected with the **-win=** argument) before performing the FFTs, so that the Welch's method used in `comp_cross_spectrum_1d` computes a modified periodogram on each segment and then averaged these estimates to produce the final PSD and CSD estimates. Because the segments usually overlap (see the **-nooverlap** argument), data values at the beginning and end of each segment tapered by the data window on one segment, occur away from the ends of adjacent segments. In this way, the overlap guards against the loss of information caused by windowing, but allows to increase the number of segments and, finally, PSD and CSD estimates on each segment can still be considered as uncorrelated.

Optionally, the PSD and CSD estimates may also be smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument).

Additionally, the spectral and cross-spectral analysis can be done separately for different segments of equal length of the selected time series if these time series are not continuous in time (see the **-p=** argument).

The PSD estimates are returned in units, which are the square of the data if the **-nonormpsd** argument is used (meaning that the sum of the PSD estimates is equal to the variance of the time series) or in PSD units otherwise (meaning that the total area under the power spectrum is equal to the variance of the time series).

Finally, again at the user option, statistical confidence interval and equivalent number of degrees of freedom of the PSD estimates and critical values for testing the null hypothesis that the squared coherency between the time series is zero can be computed (see the **-conflim=** argument).

All the final PSD and CSD estimates of the different time series and the associated statistics are stored in an output NetCDF dataset.

If the NetCDF variable is tridimensional use `comp_cross_spectrum_3d` or `comp_cross_spectrum2_3d` instead of `comp_cross_spectrum_1d`. If the time series has a seasonal (or diurnal) cycle, use `comp_stl_1d` in order to estimate and remove the harmonic components of the time series before using `comp_cross_spectrum_1d`.

Finally, use `comp_spectvar_1d` to compute variance or PSD estimates in a selected frequency band for the real time series after computing the power spectrum of the real time series.

For more information on spectral and cross-spectral analysis, see the references below [Welch] [Bloomfield] [Diggle].

2.12.4 Further Details

Usage

```
$ comp_cross_spectrum_1d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -t=time1,time2                                (optional) \
  -p=period_length                             (optional) \
  -o=output_netcdf_file                        (optional) \
  -vi=index_netcdf_variable                    (optional) \
  -ni=indice_for_2d_index_netcdf_variable      (optional) \
  -fi=input_index_netcdf_file                  (optional) \
  -ti=itime1,itime2                            (optional) \
  -pi=iperiodicity,istep                       (optional) \
  -sl=segment_length                          (optional : 16,32,64,128,...) \
```

(continues on next page)

(continued from previous page)

```

-tr=trend_removal                (optional : 0,1,2,3) \
-tr2=trend_removal_on_segment   (optional : 0,1,2,3) \
-win=window_choice              (optional : 1,2,3,4,5,6) \
-tap=tapered_percentage         (optional : 0.0 > 1.) \
-nsm=Daniell_filter_length      (optional) \
-pro=critical_probability       (optional : 0.0 > 1.) \
-mi=missing_value              (optional) \
-nonormpsd                      (optional) \
-nooverlap                      (optional) \
-conflim                        (optional) \
-double                        (optional) \
-hdf5                          (optional) \
-cdf5                          (optional) \
-tlimited                        (optional)

```

By default

- t=** the whole time period associated with the *netcdf_variable*
- p=** the *period_length* is set to `time2 - time1 + 1`, which means that the time series is considered as continuous with only one time interval
- o=** *output_netcdf_file* name is set to `cross_spectrum_netcdf_variable.index_netcdf_variable.nc` if *index_netcdf_variable* is specified with the **-vi=** argument and to `spectrum_netcdf_variable.nc` if the **-vi=** argument is not used
- vi=** a power spectrum analysis is performed
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- fi=** the same as the **-f=** argument
- ti=** the whole time period associated with the *index_netcdf_variable*
- pi=** this parameter is not used, which is equivalent to use `1, 1` for the **-pi=** argument
- sl=** the *segment_length* is set to `0`, which means that the time series (or each time interval if the **-p=** argument is used) is not segmented to stabilize the the PSD and CSD estimates
- tr=** the *trend_removal* is set to `1`, which means that the mean of the whole time series is removed before the spectral computations
- tr2=** the *trend_removal_on_segment* is set to `0`, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_choice* is set to `1`, which means that the Bartlett window is used in the spectral computations
- tap=** the *tapered_percentage* is set to `0.2`, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to `6` with the **-win=** argument), 20% of the data are tapered
- nsm=** the *Daniell_filter_length* is set to `0`, which means that the PSD and CSD estimates are not smoothed in the frequency domain by a moving average
- pro=** the *critical_probability* is set to `0.05`, which means that if a confidence interval for the power spectrum is requested with the **-conflim** argument, the lower and upper 95% confidence limit factors are computed. In addition, if the **-vi=** argument is used, the lower and upper $(1-0.05)*100\%$ confidence limit factors and the critical value for testing the null hypothesis that the squared coherency is zero at the $0.05*100\%$ significance level are computed

- mi=** the *missing_value* for the output NetCDF variables is set to $1.e+20$
- nonormpsd** the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series. If **-nonormpsd** is specified, the PSD estimates are returned in units which are the square of the data
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- conflim** normally, the equivalent number of degrees of freedom, the confidence limit factors of the PSD estimates and the critical value for testing the null hypothesis that the squared coherency is zero are not computed. However, if **-conflim** is specified, these statistics are computed and stored in the output NetCDF dataset
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the segment axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable (e.g., the real time series) for which a power or cross-power spectrum analysis must be performed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to estimate the power and cross-power spectrum.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

 $time2 - time1 + 1$ must be an even integer if the **-sl=** argument is not used and of any length if this argument is used.
- 3) If the **-p=** argument is specified, the PSD and CSD estimates are computed separately for each time period of length *period_length* (as determined by the value of the **-p=** argument).

The whole selected time period (e.g., $time2 - time1 + 1$) must be a multiple of the *period_length*. The *period_length* must also be of even length if the **-sl=** argument is not used.
- 4) The **-vi=index_netcdf_variable** specifies a time series for the cross-power spectrum analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

If the **-vi=index_netcdf_variable** is absent, a power spectrum analysis of the NetCDF variable *netcdf_variable* specified by the **-v=** argument is performed.
- 5) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the index time series if the *index_netcdf_variable* specified with the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_netcdf_variable* to 1.

- 6) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 7) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute a cross-spectral analysis with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.
- 8) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$\text{time2} - \text{time1} + 1 = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$
 otherwise, an error message will be issued and the program will stop.
- 9) The **-sl=segment_length** argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., $\text{time2} - \text{time1} + 1$) or the *period_length* if the **-p=** argument is used. Spectral computations are at $(\text{segment_length}/2) + 1$ Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies $(i-1) / \text{segment_length}$ for $i=1, 2, \dots, (\text{segment_length}/2) + 1$). By default, *segment_length* is set to 0, e.g., the time series is not segmented and *segment_length* is equal to $\text{time2} - \text{time1} + 1$ or *period_length* if the **-p=** argument is used.
- 10) The **-tr=** argument specifies pre-filtering processing of the real time series before estimating the PSD and CSD estimates of the time series. If:
 - **-tr=+1**, the mean of the time series is removed before computing the PSD and CSD estimates
 - **-tr=+2**, the drift from the time series is removed before computing the PSD and CSD estimates. The drift for each time series (say *tseries*) is estimated using the formula : $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
 - **-tr=+3**, the least-squares line from the time series is removed before computing the PSD and CSD estimates.

For other values of the **-tr=** argument, nothing is done before estimating the PSD and CSD estimates.

If the **-p=** argument is present, the pre-filtering processing is applied to each time interval of length *period_length*, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 1, e.g., the mean of the time series is removed before the spectrum computations.

- 11) The **-tr2=** argument specifies pre-filtering processing of the real time series before estimating the PSD and CSD estimates on each time segment. If:
 - **-tr2=+1**, the mean of the time segment is removed before computing the PSD and CSD estimates on the segment
 - **-tr2=+2**, the drift from the time segment is removed before computing the PSD and CSD estimates on the segment
 - **-tr2=+3**, the least-squares line from the time segment is removed before computing the PSD and CSD estimates on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the PSD and CSD estimates on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 12) The **-win=***window_choice* argument specifies the form of the data window used in the computations of the modified periodogram on each segment. If:

- **-win=1** The Bartlett window is used
- **-win=2** The square window is used
- **-win=3** The Welch window is used
- **-win=4** The Hann window is used
- **-win=5** The Hamming window is used
- **-win=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6. The default value for the *window_choice* is 1, e.g., the Bartlett window is used for the spectral computations on each segment.

- 13) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 14) The **-nsm=Daniell_filter_length** argument specified that the PSD and CSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average). On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD and CSD estimates are not smoothed in the frequency domain.

- 15) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, *comp_cross_spectrum_1d* segments the time series without any overlapping (in each time period if the **-p=** argument is used).

By default, *comp_cross_spectrum_1d* overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting (modified if the **-win=** argument is used) periodograms will be averaged together to obtain PSD and CSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 16) If the **-nonormpsd** argument is specified, the sum of the PSD estimates is equal to the variance of the time series. By default, the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series.

- 17) If the **-conflim** argument is specified, the equivalent number of degrees of freedom, the lower and upper $(1 - critical_probability) * 100\%$ confidence limit factors of the PSD estimates and the critical value for testing the null hypothesis that the squared coherency is zero at the $critical_probability * 100\%$ significance level are computed.

Multiply the PSD estimates by the lower and upper confidence limit factors to get the lower and upper limits of a $(1 - critical_probability) * 100\%$ confidence interval for the PSD estimates.

Squared coherency less than the computed critical value should be regarded as not significantly different from zero at the $critical_probability * 100\%$ significance level.

By default, these statistics are not computed.

- 18) The **-pro=critical_probability** argument specifies the critical probability which is used to determine the lower and upper confidence limit factors and the critical value for testing the null hypothesis that the squared coherency is zero.

critical_probability must be greater than 0. and less than 1.. The default value is 0.05.

- 19) The **-mi=missing_value** argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 20) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file. By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or

`-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 23) It is assumed that the different real time series have no missing values.

- 24) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 25) For more details on power and cross-power spectrum analysis and examples of use in the climate literature, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- “Fourier analysis of time series- An introduction”, by P. Bloomfield, John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
- “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
- “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
- “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_cross_spectrum_1d` creates an output NetCDF file that contains the PSD and CSD estimates of the time series associated with the input NetCDF variables. The output NetCDF dataset contains the following NetCDF variables (in the description below, *nfreq* is the number of frequencies where the PSD and CSD estimates are computed (as determined by the **-t=**, **-p=** and **-sl=** arguments) and *nperiod* is 1 if the **-p=** argument is not used or equal to $\text{ntime} / \text{period_length}$ if it is used, where $\text{ntime} = \text{time2} - \text{time1} + 1$):

- 1) `frequency(nfreq)` : the frequencies at which the PSD estimates are computed.
- 2) `periodicity(nfreq)` : the corresponding periodicities in number of time observations.
- 3) `netcdf_variable_spectrum(nperiod,nfreq)` : the corresponding PSD estimates for each time interval of the time series associated with the input NetCDF variable.
- 4) `index_netcdf_variable_spectrum(nperiod,nfreq)` : the corresponding PSD estimates for each time interval of the time series associated with the input index NetCDF variable.

This variable is stored only if the **-vi=** argument has been specified when calling `comp_cross_spectrum_1d`.

- 5) `index_netcdf_variable_netcdf_variable_squaredcoherency(nperiod,nfreq)` : the corresponding squared coherency estimates for each time interval of the time series associated with the input NetCDF variable and the time series associated with the input index NetCDF variable. The square coherency is given on the closed interval $[0., 1.]$.

This variable is stored only if the **-vi=** argument has been specified when calling `comp_cross_spectrum_1d`.

- 6) `index_netcdf_variable_netcdf_variable_phase(nperiod,nfreq)` : the corresponding phase estimates of the cross-spectrum for each time interval of the time series associated with the input NetCDF variable and the time series associated with the input index NetCDF variable. In general terms, the phase gives the lag (negative values) or lead (positive values) of the `index_netcdf_variable` time series with respect to the `netcdf_variable` time series at each frequency in radians.

Here, the phase is given in fractions of a circle (e.g., on the closed interval $[0., 1.]$) instead of being in radians. In other words, for phase values less than 0.5 the `index_netcdf_variable` leads the `netcdf_variable` and vice-versa for phase values greater than 0.5.

This variable is stored only if the **-vi=** argument has been specified when calling `comp_cross_spectrum_1d`.

- 7) `edof(nfreq)` : the equivalent number of degrees of freedom of the PSD and CSD estimates of the time series associated with the input NetCDF variable.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum_1d`.

- 8) `lower_factor(nfreq)` : the lower confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding lower confidence limit factors to get the lower limits of the $(1 - \text{critical_probability}) * 100\%$ confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum_1d`.

- 9) `upper_factor(nfreq)` : the upper confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding upper confidence limit factors to get the upper limits of the $(1 - \text{critical_probability}) * 100\%$ confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum_1d`.

- 10) `coher_test(nfreq)` : the critical values for the statistical test of the squared coherency estimates. Squared coherency less than these computed critical values for a given frequency should be regarded as not significantly different from zero at the $\text{critical_probability} * 100\%$ significance level.

This variable is stored only if the **-conflim** and **-vi=** arguments have been specified when calling `comp_cross_spectrum_1d`.

Examples

- 1) For computing the power and cross-power spectra of a real monthly time series from a unidimensional NetCDF variable called `sst_iod` extracted from the file `sst.monthly.iod.nc` with a monthly Nino34 SST index in a NetCDF variable called `sst_nino34` extracted from the file `sst.monthly.nino34.nc` and store the results in a NetCDF file named `cross_spectrum_iod_nino34.nc`, use the following command:

```
$ comp_cross_spectrum_1d \
-f=sst.monthly.iod.nc \
-v=sst_iod \
-fi=sst.monthly.nino34.nc \
-vi=sst_nino34 \
-sl=128 \
-o=cross_spectrum_iod_nino34.nc \
-nonormpsd \
-nooverlap
```

2.13 comp_cross_spectrum_3d

2.13.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.13.2 Latest revision

30/04/2026

2.13.3 Purpose

Compute Fast Fourier Transform (FFT) estimates of the power spectra and cross-power spectra between an index time series and a tridimensional variable readed from a NetCDF dataset. From the Power Spectral Density (PSD) and Cross-Spectral Density (CSD) estimates, this program also computes the amplitude and phase relationships between the signals, e.g., the phases of the multi-channel cross-spectrum and the squared coherency estimates at each frequency. In output of `comp_cross_spectrum_3d`, both are given on the closed interval $[0., 1.]$, meaning that both are unitless.

In general terms, the phase gives the lag (negative values) or lead (positive values) of the index time series with respect to the variable multi-channel time series at each frequency in radians. Here, the phase is given in fractions of a circle (e.g., on the closed interval $[0., 1.]$) instead of being in radians. In other words, for phase values less than 0.5 the index time series leads the variable multi-channel time series and vice-versa for phase values greater than 0.5.

The magnitude-squared-coherence can be used to identify significant frequency-domain correlation (at the Fourier frequencies) between the index time series and the multi-channel time series. Phase estimates in the cross-spectrum are only useful where significant frequency domain correlation exists and you can neglect the cross spectrum when the squared coherency is small.

Spectral analysis amounts to estimate the power spectrum of a signal (e.g., a time series) from its time representation. The PSD estimates, which constitute the power spectrum, characterize the frequency content of the signal and decompose the variance of the signal into the different frequencies and allow to identify the periodicities present in the signal. In a similar way, cross-spectral analysis amounts to decompose the correlation between two signals into the different frequencies.

`comp_cross_spectrum_3d` is based on a non-parametric method, which divides the time series into segments (see the `-sl=` argument), applies FFT on each segment, computes the PSD and CSD estimates from the FFTs and, finally,

averages these estimates from each segment. Such non-parametric method is based on the observed time series and does not require prior knowledge about the data or a model generating the data. This method is known as the Welch's averaged, modified periodogram method for estimating the power and cross-power spectra of different time series [Welch]. This Welch's Overlapped Segment Averaging (WOSA) method also provides a consistent average coherence estimator, which prevents obtaining a squared coherency estimate that is identically 1. for all frequencies.

At the user option, the different time series can be detrended before computing the PSD and CSD estimates (see the **-tr=** and **-tr2=** arguments) in order to ensure that the analyzed time series remain stationary.

In order to obtain stable PSD and CSD estimates (e.g., reduce the variance of the estimates) and minimize the loss of information, the real time series can also be segmented, and PSD and CSD estimates on, eventually overlapping, segments can be averaged at the user option (this is known as the Welch's method as already noted above). The stability of the PSD and CSD estimates is increased by this averaging process because the Welch's method uses PSD and CSD estimates from different segments of the time series, which represent approximately uncorrelated estimates of the true PSD and CSD estimates. That is, the greater the number of segments, the more stable, the resulting PSD and CSD estimates as their variance is reduced by the averaging.

The user can also select different data windows for the computations of the PSD and CSD estimates (see the **-win=** argument) on each segment. The segments are multiplied by a data window (selected with the **-win=** argument) before performing the FFTs, so that the Welch's method used in `comp_cross_spectrum_3d` computes a modified periodogram on each segment and then averaged these estimates to produce the final PSD and CSD estimates. Because the segments usually overlap (see the **-nooverlap** argument), data values at the beginning and end of each segment tapered by the data window on one segment, occur away from the ends of adjacent segments. In this way, the overlap guards against the loss of information caused by windowing, but allows to increase the number of segments and, finally, PSD and CSD estimates on each segment can still be considered as uncorrelated.

Optionally, the PSD and CSD estimates may also be smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument).

Additionally, the spectral and cross-spectral analysis can be done separately for different segments of equal length of the selected multi-channel time series if the time series are not continuous in time (see the **-p=** argument).

The PSD estimates are returned in units, which are the square of the data if the **-nonormpsd** argument is used (meaning that the sum of the PSD estimates is equal to the variance of the time series) or in PSD units otherwise (meaning that the total area under the power spectrum is equal to the variance of the time series).

Finally, again at the user option, statistical confidence interval and equivalent number of degrees of freedom of the PSD estimates and critical values for testing the null hypothesis that the squared coherency between the time series is zero can be computed (see the **-conflim=** argument).

All the final PSD and CSD estimates of the different time series and the associated statistics are stored in an output NetCDF dataset.

If the NetCDF variable is unidimensional use `comp_cross_spectrum_1d` instead of `comp_cross_spectrum_3d`. If the multi-channel time series has a seasonal (or diurnal) cycle, use `comp_stl_3d` in order to estimate and remove the harmonic components of the time series before using `comp_cross_spectrum_3d`.

`comp_cross_spectrum_3d` performs the same task as `comp_cross_spectrum2_3d`, but is more flexible (e.g., the **-tr=** and **-p=** arguments are not allowed in `comp_cross_spectrum2_3d`). However, for the analysis of very large NetCDF datasets, `comp_cross_spectrum2_3d` requires much less memory and will be faster if OpenMP parallelism is used.

Finally, use `comp_spectvar_3d` to compute variance or PSD estimates in a selected frequency band for the real multi-channel time series after computing the power spectra of the real multi-channel time series.

For more information on spectral and cross-spectral analysis, see the references below [Welch] [Bloomfield] [Diggle].

2.13.4 Further Details

Usage

```
$ comp_cross_spectrum_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file           (optional) \
  -g=grid_type                             (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                             (optional) \
  -y=lat1,lat2                             (optional) \
  -t=time1,time2                           (optional) \
  -p=period_length                         (optional) \
  -o=output_netcdf_file                   (optional) \
  -vi=index_netcdf_variable                (optional) \
  -ni=indice_for_2d_index_netcdf_variable (optional) \
  -fi=input_index_netcdf_file              (optional) \
  -ti=itime1,itime2                       (optional) \
  -pi=iperiodicity,istep                   (optional) \
  -sl=segment_length                       (optional : 16,32,64,128,...) \
  -tr=trend_removal                       (optional : 0,1,2,3) \
  -tr2=trend_removal_on_segment            (optional : 0,1,2,3) \
  -win=window_choice                      (optional : 1,2,3,4,5,6) \
  -tap=tapered_percentage                  (optional : 0.0 > 1.) \
  -nsm=Daniell_filter_length               (optional) \
  -pro=critical_probability                 (optional : 0.0 > 1.) \
  -mi=missing_value                       (optional) \
  -nonormpsd                              (optional) \
  -nooverlap                              (optional) \
  -conflim                                (optional) \
  -nospect                                (optional) \
  -double                                  (optional) \
  -bigfile                                 (optional) \
  -hdf5                                    (optional) \
  -cdf5                                    (optional) \
  -tlimited                                 (optional)
```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *period_length* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time interval
- o=** *output_netcdf_file* name is set to *cross_spectrum_netcdf_variable.index_netcdf_variable.nc* if *index_netcdf_variable* is specified with the **-vi=** argument and to *spectrum_netcdf_variable.nc* if the **-vi=** argument is not used
- vi=** a power spectrum analysis is performed
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- fi=** the same as the **-f=** argument

- ti=** the whole time period associated with the *index_netcdf_variable*
- pi=** this parameter is not used, which is equivalent to use 1, 1 for the **-pi=** argument
- sl=** the *segment_length* is set to 0, which means that the time series (or each time interval if the **-p=** argument is used) is not segmented to stabilize the PSD and CSD estimates
- tr=** the *trend_removal* is set to 1, which means that the mean of the whole time series is removed before the spectral computations
- tr2=** the *trend_removal_on_segment* is set to 0, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_choice* is set to 1, which means that the Bartlett window is used in the spectral computations
- tap=** the *tapered_percentage* is set to 0.2, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to 6 with the **-win=** argument), 20% of the data are tapered
- nsm=** the *Daniell_filter_length* is set to 0, which means that the PSD and CSD estimates are not smoothed in the frequency domain by a moving average
- pro=** the *critical_probability* is set to 0.05, which means that if a confidence interval for the power spectrum is requested with the **-conflim** argument, the lower and upper 95% confidence limit factors are computed. In addition, if the **-vi=** argument is used, the lower and upper $(1-0.05) \times 100\%$ confidence limit factors and the critical value for testing the null hypothesis that the squared coherency is zero at the $0.05 \times 100\%$ significance level are computed
- mi=** the *missing_value* for the output NetCDF variables is set to $1.e+20$
- nonormpsd** the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series. If **-nonormpsd** is specified, the PSD estimates are returned in units which are the square of the data
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- conflim** normally, the equivalent number of degrees of freedom, the confidence limit factors of the PSD estimates and the critical value for testing the null hypothesis that the squared coherency is zero are not computed. However, if **-conflim** is specified, these statistics are computed and stored in the output NetCDF dataset
- nospect** the PSD estimates are stored in the output NetCDF dataset
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the segment axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable (e.g., the multi-channel time series) for which a power or cross-power spectrum analysis must be performed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before computing the correlation analysis. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_3d](#) or [comp_mask_3d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_cross_spectrum_3d*.

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.
- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_cross_spectrum_3d*.

- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to estimate the power and cross-power spectrum.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

$time2 - time1 + 1$ must be an even integer if the **-sl=** argument is not used and of any length if this argument is used.

- 6) If the **-p=** argument is specified, the PSD and CSD estimates are computed separately for each time period of length *period_length* (as determined by the value of the **-p=** argument).

The whole selected time period (e.g., $time2 - time1 + 1$) must be a multiple of the *period_length*. The *period_length* must also be of even length if the **-sl=** argument is not used.

- 7) The **-vi=index_netcdf_variable** specifies a time series for the cross-power spectrum analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

If the **-vi=index_netcdf_variable** is absent, a power spectrum analysis of the NetCDF variable *netcdf_variable* specified by the **-v=** argument is performed.

- 8) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the index time series if the *index_netcdf_variable* specified with the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_netcdf_variable* to 1.

- 9) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 10) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute a cross-spectral analysis with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.
- 11) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$\text{time2} - \text{time1} + 1 = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$

otherwise, an error message will be issued and the program will stop.

- 12) The **-sl=segment_length** argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., $\text{time2} - \text{time1} + 1$) or the *period_length* if the **-p=** argument is used. Spectral computations are at $(\text{segment_length}/2) + 1$ Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies $(i-1) / \text{segment_length}$ for $i=1, 2, \dots, (\text{segment_length}/2) + 1$). By default, *segment_length* is set to 0, e.g., the time series is not segmented and *segment_length* is equal to $\text{time2} - \text{time1} + 1$ or *period_length* if the **-p=** argument is used.
- 13) The **-tr=** argument specifies pre-filtering processing of the real time series before estimating the PSD and CSD estimates of the time series. If:
- **-tr=+1**, the mean of the time series is removed before computing the PSD and CSD estimates
 - **-tr=+2**, the drift from the time series is removed before computing the PSD and CSD estimates. The drift for each time series (say *tseries*) is estimated using the formula : $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
 - **-tr=+3**, the least-squares line from the time series is removed before computing the PSD and CSD estimates.

For other values of the **-tr=** argument, nothing is done before estimating the PSD and CSD estimates.

If the **-p=** argument is present, the pre-filtering processing is applied to each time interval of length *period_length*, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 1, e.g., the mean of the time series is removed before the spectrum computations.

- 14) The **-tr2=** argument specifies pre-filtering processing of the real time series before estimating the PSD and CSD estimates on each time segment. If:
- **-tr2=+1**, the mean of the time segment is removed before computing the PSD and CSD estimates on the segment
 - **-tr2=+2**, the drift from the time segment is removed before computing the PSD and CSD estimates on the segment
 - **-tr2=+3**, the least-squares line from the time segment is removed before computing the PSD and CSD estimates on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the PSD and CSD estimates on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 15) The **-win=***window_choice* argument specifies the form of the data window used in the computations of the modified periodogram on each segment. If:

- **-win=1** The Bartlett window is used
- **-win=2** The square window is used
- **-win=3** The Welch window is used
- **-win=4** The Hann window is used
- **-win=5** The Hamming window is used
- **-win=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6. The default value for the *window_choice* is 1, e.g., the Bartlett window is used for the spectral computations on each segment.

- 16) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 17) The **-nsm=Daniell_filter_length** argument specified that the PSD and CSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average). On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD and CSD estimates are not smoothed in the frequency domain.

- 18) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, `comp_cross_spectrum_3d` segments the time series without any overlapping (in each time period if the **-p=** argument is used).

By default, `comp_cross_spectrum_3d` overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting (modified if the **-win=** argument is used) periodograms and cross-periodograms will be averaged together to obtain PSD and CSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 19) If the **-nonormpsd** argument is specified, the sum of the PSD estimates is equal to the variance of the time series. By default, the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series.

- 20) If the **-conflim** argument is specified, the equivalent number of degrees of freedom, the lower and upper $(1 - critical_probability) * 100\%$ confidence limit factors of the PSD estimates and the critical value for testing the null hypothesis that the squared coherency is zero at the $critical_probability * 100\%$ significance level are computed.

Multiply the PSD estimates by the lower and upper confidence limit factors to get the lower and upper limits of a $(1 - critical_probability) * 100\%$ confidence interval for the PSD estimates.

Squared coherency less than the computed critical value should be regarded as not significantly different from zero at the $critical_probability * 100\%$ significance level.

By default, these statistics are not computed.

- 21) The **-pro=critical_probability** argument specifies the critical probability which is used to determine the lower and upper confidence limit factors and the critical value for testing the null hypothesis that the squared coherency is zero.

critical_probability must be greater than 0. and less than 1.. The default value is 0.05.

- 22) The **-mi=missing_value** argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to 1.e+20.

- 23) If the **-nospect** argument is specified, the PSD estimates of the time series are not stored in the output netCDF file. By default, the PSD estimates are stored.

- 24) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 25) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 26) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 27) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 28) It is assumed that the different real time series have no missing values.

- 29) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 30) For more details on power and cross-power spectrum analysis and examples of use in the climate literature, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- “Fourier analysis of time series- An introduction”, by P. Bloomfield, John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
- “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
- “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
- “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_cross_spectrum_3d` creates an output NetCDF file that contains the PSD and CSD estimates of the time series associated with the input NetCDF variables. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable `netcdf_variable`, `nfreq` is the number of frequencies where the PSD and CSD estimates are computed (as determined by the `-t=`, `-p=` and `-sl=` arguments) and `nperiod` is 1 if the `-p=` argument is not used or equal to `ntime / period_length` if it is used, where `ntime = time2 - time1 + 1`):

- 1) `frequency(nfreq)` : the frequencies at which the PSD estimates are computed.
- 2) `periodicity(nfreq)` : the corresponding periodicities in number of time observations.
- 3) `netcdf_variable_spectrum(nperiod, nfreq, nlat, nlon)` : the corresponding PSD estimates for each time interval of the multi-channel time series associated with the input NetCDF variable.

This variable is stored only if the `-nospect` argument is absent when calling `comp_cross_spectrum_3d`.

- 4) `index_netcdf_variable_spectrum(nperiod, nfreq)` : the corresponding PSD estimates for each time interval of the time series associated with the input index NetCDF variable.

This variable is stored only if the `-vi=` argument has been specified when calling `comp_cross_spectrum_3d`.

- 5) `index_netcdf_variable_netcdf_variable_squaredcoherency(nperiod, nfreq, nlat, nlon)` : the corresponding squared coherency estimates for each time interval of the multi-channel time series associated with the input NetCDF variable and the time series associated with the input index NetCDF variable. The square coherency is given on the closed interval $[0, 1]$.

This variable is stored only if the `-vi=` argument has been specified when calling `comp_cross_spectrum_3d`.

- 6) `index_netcdf_variable_netcdf_variable_phase(nperiod, nfreq, nlat, nlon)` : the corresponding phase estimates of the cross-spectrum for each time interval of the multi-channel time series associated with the input NetCDF variable and the time series associated with the input index NetCDF variable. In general terms, the phase gives the lag (negative values) or lead (positive values) of the `index_netcdf_variable` time series with respect to the `netcdf_variable` multi-channel time series at each frequency in radians.

Here, the phase is given in fractions of a circle (e.g., on the closed interval $[0., 1.]$) instead of being in radians. In other words, for phase values less than 0.5 the `index_netcdf_variable` leads the `netcdf_variable` and vice-versa for phase values greater than 0.5.

This variable is stored only if the `-vi=` argument has been specified when calling `comp_cross_spectrum_3d`.

- 7) `edof(nfreq)` : the equivalent number of degrees of freedom of the PSD and CSD estimates of the time series associated with the input NetCDF variable.

This variable is stored only if the `-conflim` argument has been specified when calling `comp_cross_spectrum_3d`.

- 8) `lower_factor(nfreq)` : the lower confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding lower confidence limit factors to get the lower limits of the $(1 - critical_probability) * 100\%$ confidence intervals for the PSD estimates.

This variable is stored only if the `-conflim` argument has been specified when calling `comp_cross_spectrum_3d`.

- 9) `upper_factor(nfreq)` : the upper confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding upper confidence limit factors to get the upper limits of the $(1 - \text{critical_probability}) * 100$ % confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum_3d`.

- 10) `coher_test(nfreq)` : the critical values for the statistical test of the squared coherency estimates. Squared coherency less than these computed critical values for a given frequency should be regarded as not significantly different from zero at the $\text{critical_probability} * 100$ % significance level.

This variable is stored only if the **-conflim** and **-vi=** arguments have been specified when calling `comp_cross_spectrum_3d`.

All the PSD and CSD estimates involving the input NetCDF variable *netcdf_variable* are packed in tri- or fourdimensional variables whose first and second dimensions are exactly the same as those associated with this input NetCDF variable even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For computing the power and cross-power spectra of a real multi-channel monthly time series from a tridimensional NetCDF variable called `sst` extracted from the file `sst.monthly.nc` with a monthly Nino34 SST index in a NetCDF variable called `sst_nino34` extracted from the file `sst.monthly.nino34.nc` and store the results in a NetCDF file named `cross_spectrum_sst_nino34_sst.nc`, use the following command:

```
$ comp_cross_spectrum_3d \  
-f=sst.monthly.nc \  
-v=sst \  
-fi=sst.monthly.nino34.nc \  
-vi=sst_nino34 \  
-sl=128 \  
-o=cross_spectrum_sst_nino34_sst.nc \  
-nonormpsd \  
-nooverlap
```

2.14 comp_cross_spectrum2_3d

2.14.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.14.2 Latest revision

30/04/2026

2.14.3 Purpose

Compute Fast Fourier Transform (FFT) estimates of the power spectra and cross-power spectra between an index time series and a tridimensional variable readed from a NetCDF dataset. From the Power Spectral Density (PSD) and Cross-Spectral Density (CSD) estimates, this program also computes the amplitude and phase relationships between

the signals, e.g., the phases of the multi-channel cross-spectrum and the squared coherency estimates at each frequency. In output of `comp_cross_spectrum_3d`, both are given on the closed interval $[0., 1.]$, meaning that both are unitless.

In general terms, the phase gives the lag (negative values) or lead (positive values) of the index time series with respect to the variable multi-channel time series at each frequency in radians. Here, the phase is given in fractions of a circle (e.g., on the closed interval $[0., 1.]$) instead of being in radians. In other words, for phase values less than 0.5 the index time series leads the variable multi-channel time series and vice-versa for phase values greater than 0.5.

The magnitude-squared-coherence can be used to identify significant frequency-domain correlation (at the Fourier frequencies) between the index time series and the multi-channel time series. Phase estimates in the cross-spectrum are only useful where significant frequency domain correlation exists and you can neglect the cross spectrum when the squared coherency is small.

Spectral analysis amounts to estimate the power spectrum of a signal (e.g., a time series) from its time representation. The PSD estimates, which constitute the power spectrum, characterize the frequency content of the signal and decompose the variance of the signal into the different frequencies and allow to identify the periodicities present in the signal. In a similar way, cross-spectral analysis amounts to decompose the correlation between two signals into the different frequencies.

`comp_cross_spectrum2_3d` is based on a non-parametric method, which divides the time series into segments (see the **-sl=** argument), applies FFT on each segment, computes the PSD and CSD estimates from the FFTs and, finally, averages these estimates from each segment. Such non-parametric method is based on the observed time series and does not require prior knowledge about the data or a model generating the data. This method is known as the Welch's averaged, modified periodogram method for estimating the power and cross-power spectra of different time series [Welch]. This Welch's Overlapped Segment Averaging (WOSA) method also provides a consistent average coherence estimator, which prevents obtaining a squared coherency estimate that is identically 1. for all frequencies.

At the user option, the different segments of each time series can be detrended before computing the PSD and CSD estimates (see the **-tr2=** argument) in order to ensure that the analyzed time series remain stationary on each segment.

In order to obtain stable PSD and CSD estimates (e.g., reduce the variance of the estimates) and minimize the loss of information, the real time series can also be segmented, and PSD and CSD estimates on, eventually overlapping, segments can be averaged at the user option (this is known as the Welch's method as already noted above). The stability of the PSD and CSD estimates is increased by this averaging process because the Welch's method uses PSD and CSD estimates from different segments of the time series, which represent approximately uncorrelated estimates of the true PSD and CSD estimates. That is, the greater the number of segments, the more stable, the resulting PSD and CSD estimates as their variance is reduced by the averaging.

The user can also select different data windows for the computations of the PSD and CSD estimates (see the **-win=** argument) on each segment. The segments are multiplied by a data window (selected with the **-win=** argument) before performing the FFTs, so that the Welch's method used in `comp_cross_spectrum2_3d` computes a modified periodogram on each segment and then averaged these estimates to produce the final PSD and CSD estimates. Because the segments usually overlap (see the **-nooverlap** argument), data values at the beginning and end of each segment tapered by the data window on one segment, occur away from the ends of adjacent segments. In this way, the overlap guards against the loss of information caused by windowing, but allows to increase the number of segments and, finally, PSD and CSD estimates on each segment can still be considered as uncorrelated.

Optionally, the PSD and CSD estimates may also be smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument).

The PSD estimates are returned in units, which are the square of the data if the **-nonormpsd** argument is used (meaning that the sum of the PSD estimates is equal to the variance of the time series) or in PSD units otherwise (meaning that the total area under the power spectrum is equal to the variance of the time series).

Finally, again at the user option, statistical confidence interval and equivalent number of degrees of freedom of the PSD estimates and critical values for testing the null hypothesis that the squared coherency between the time series is zero can be computed (see the **-conflim=** argument).

All the final PSD and CSD estimates of the different time series and the associated statistics are stored in an output NetCDF dataset.

If the NetCDF variable is unidimensional use *comp_cross_spectrum_1d* instead of *comp_cross_spectrum2_3d*. If the multi-channel time series has a seasonal (or diurnal) cycle, use *comp_stl_3d* in order to estimate and remove the harmonic components of the time series before using *comp_cross_spectrum2_3d*.

comp_cross_spectrum2_3d performs the same task as *comp_cross_spectrum_3d*, but requires much less memory and will be faster than *comp_cross_spectrum_3d* if OpenMP parallelism is used. On the other hand, *comp_cross_spectrum2_3d* is slightly less flexible (e.g., the **-tr=** and **-p=** arguments are not allowed in *comp_cross_spectrum2_3d*).

Finally, use *comp_spectvar_3d* to compute variance or PSD estimates in a selected frequency band for the real multi-channel time series after computing the power spectra of the real multi-channel time series.

For more information on spectral and cross-spectral analysis, see the references below [Welch] [Bloomfield] [Diggle].

2.14.4 Further Details

Usage

```
$ comp_cross_spectrum2_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file           (optional) \
  -g=grid_type                             (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                             (optional) \
  -y=lat1,lat2                             (optional) \
  -t=time1,time2                           (optional) \
  -o=output_netcdf_file                   (optional) \
  -vi=index_netcdf_variable               (optional) \
  -ni=indice_for_2d_index_netcdf_variable (optional) \
  -fi=input_index_netcdf_file             (optional) \
  -ti=itime1,itime2                       (optional) \
  -pi=iperiodicity,istep                   (optional) \
  -sl=segment_length                      (optional : 16,32,64,128,...) \
  -tr2=trend_removal_on_segment            (optional : 0,1,2,3) \
  -win=window_choice                      (optional : 1,2,3,4,5,6) \
  -tap=tapered_percentage                  (optional : 0.0 > 1.) \
  -nsm=Daniell_filter_length               (optional) \
  -pro=critical_probability                (optional : 0.0 > 1.) \
  -mi=missing_value                       (optional) \
  -nonormpsd                              (optional) \
  -nooverlap                              (optional) \
  -conflim                                (optional) \
  -nospect                                (optional) \
  -double                                  (optional) \
  -bigfile                                 (optional) \
  -hdf5                                    (optional) \
  -cdf5                                    (optional) \
  -tlimited                                 (optional)
```

By default

-m= an *input_mesh_mask_netcdf_file* is not used

- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** *output_netcdf_file* name is set to *cross_spectrum_netcdf_variable.index_netcdf_variable.nc* if *index_netcdf_variable* is specified with the **-vi=** argument and to *spectrum_netcdf_variable.nc* if the **-vi=** argument is not used
- vi=** a power spectrum analysis is performed
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- fi=** the same as the **-f=** argument
- ti=** the whole time period associated with the *index_netcdf_variable*
- pi=** this parameter is not used, which is equivalent to use 1, 1 for the **-pi=** argument
- sl=** the *segment_length* is set to 0, which means that the time series is not segmented to stabilize the PSD and CSD estimates
- tr2=** the *trend_removal_on_segment* is set to 0, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_choice* is set to 1, which means that the Bartlett window is used in the spectral computations
- tap=** the *tapered_percentage* is set to 0.2, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to 6 with the **-win=** argument), 20% of the data are tapered
- nsm=** the *Daniell_filter_length* is set to 0, which means that the PSD and CSD estimates are not smoothed in the frequency domain by a moving average
- pro=** the *critical_probability* is set to 0.05, which means that if a confidence interval for the power spectrum is requested with the **-conflim** argument, the lower and upper 95% confidence limit factors are computed. In addition, if the **-vi=** argument is used, the lower and upper $(1-0.05) \times 100\%$ confidence limit factors and the critical value for testing the null hypothesis that the squared coherency is zero at the $0.05 \times 100\%$ significance level are computed
- mi=** the *missing_value* for the output NetCDF variables is set to $1.e+20$
- nonormpsd** the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series. If **-nonormpsd** is specified, the PSD estimates are returned in units which are the square of the data
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- conflim** normally, the equivalent number of degrees of freedom, the confidence limit factors of the PSD estimates and the critical value for testing the null hypothesis that the squared coherency is zero are not computed. However, if **-conflim** is specified, these statistics are computed and stored in the output NetCDF dataset
- nospect** the PSD estimates are stored in the output NetCDF dataset
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the frequency axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable (e.g., the multi-channel time series) for which a power or cross-power spectrum analysis must be performed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before computing the correlation analysis. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g. missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_3d](#) or [comp_mask_3d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using [comp_cross_spectrum2_3d](#).

- 3) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.
- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_cross_spectrum2_3d](#).

- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to estimate the power and cross-power spectrum.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

$time2 - time1 + 1$ must be an even integer if the **-sl=** argument is not used and of any length if this argument is used.

- 6) The **-vi=index_netcdf_variable** specifies a time series for the cross-power spectrum analysis. If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset which contains the *index_netcdf_variable* is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

If the **-vi=index_netcdf_variable** is absent, a power spectrum analysis of the NetCDF variable *netcdf_variable* specified by the **-v=** argument is performed.

- 7) The **-ni=** argument specifies the indice (e.g., an integer) for selecting the index time series if the *index_netcdf_variable* specified with the **-vi=** argument is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *indice_for_2d_netcdf_variable* to 1.
- 8) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 9) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute a cross-power analysis with the January monthly time series extracted from the *index_netcdf_variable* which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.
- 10) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$\text{time2} - \text{time1} + 1 = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$
 otherwise, an error message will be issued and the program will stop.
- 11) The **-sl=segment_length** argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., $\text{time2} - \text{time1} + 1$). Spectral computations are at $(\text{segment_length}/2) + 1$ Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies $(i-1) / \text{segment_length}$ for $i=1, 2, \dots, (\text{segment_length}/2) + 1$). By default, *segment_length* is set to 0, e.g., the time series is not segmented and *segment_length* is equal to $\text{time2} - \text{time1} + 1$.
- 12) The **-tr2=** argument specifies pre-filtering processing of the real time series before estimating the PSD and CSD estimates on each time segment. If:
 - **-tr2=+1**, the mean of the time segment is removed before computing the PSD and CSD estimates on the segment
 - **-tr2=+2**, the drift from the time segment is removed before computing the PSD and CSD estimates on the segment
 - **-tr2=+3**, the least-squares line from the time segment is removed before computing the PSD and CSD estimates on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the PSD and CSD estimates on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 13) The **-win=window_choice** argument specifies the form of the data window used in the computations of the modified periodogram on each segment. If:
 - **-win=1** The Bartlett window is used
 - **-win=2** The square window is used
 - **-win=3** The Welch window is used
 - **-win=4** The Hann window is used
 - **-win=5** The Hamming window is used
 - **-win=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6. The default value for the *window_choice* is 1, e.g., the Bartlett window is used for the spectral computations on each segment.

- 14) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 15) The **-nsm=Daniell_filter_length** argument specified that the PSD and CSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average). On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD and CSD estimates are not smoothed in the frequency domain.

- 16) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, *comp_cross_spectrum2_3d* segments the time series without any overlapping.

By default, *comp_cross_spectrum2_3d* overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting (modified if the **-win=** argument is used) periodograms and cross-periodograms will be averaged together to obtain PSD and CSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 17) If the **-nonormpsd** argument is specified, the sum of the PSD estimates is equal to the variance of the time series. By default, the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series.

- 18) If the **-conflim** argument is specified, the equivalent number of degrees of freedom, the lower and upper $(1 - critical_probability) * 100\%$ confidence limit factors of the PSD estimates and the critical value for testing the null hypothesis that the squared coherency is zero at the $critical_probability * 100\%$ significance level are computed.

Multiply the PSD estimates by the lower and upper confidence limit factors to get the lower and upper limits of a $(1 - critical_probability) * 100\%$ confidence interval for the PSD estimates.

Squared coherency less than the computed critical value should be regarded as not significantly different from zero at the $critical_probability * 100\%$ significance level.

By default, these statistics are not computed.

- 19) The **-pro=critical_probability** argument specifies the critical probability which is used to determine the lower and upper confidence limit factors and the critical value for testing the null hypothesis that the squared coherency is zero.

critical_probability must be greater than 0. and less than 1.. The default value is 0.05.

- 20) The **-mi=missing_value** argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 21) If the **-nospect** argument is specified, the PSD estimates of the time series are not stored in the output NetCDF file. By default, the PSD estimates are stored.

- 22) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 23) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 24) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 25) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 26) It is assumed that the different real time series have no missing values.
- 27) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 28) For more details on power and cross-power spectrum analysis and examples of use in the climate literature, see
- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
 - “Fourier analysis of time series- An introduction”, by P. Bloomfield, John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
 - “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
 - “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
 - “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
 - “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_cross_spectrum2_3d` creates an output NetCDF file that contains the PSD and CSD estimates of the time series associated with the input NetCDF variables. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable *netcdf_variable*, `nfreq` is the number of frequencies where the PSD and CSD estimates are computed (as determined by the **-t=** and **-sl=** arguments):

- 1) frequency (`nfreq`) : the frequencies at which the PSD estimates are computed.
- 2) periodicity (`nfreq`) : the corresponding periodicities in number of time observations.

- 3) `netcdf_variable_spectrum(nfreq, nlat, nlon)` : the corresponding PSD estimates for the multi-channel time series associated with the input NetCDF variable.

This variable is stored only if the **-nospect** argument is absent when calling `comp_cross_spectrum2_3d`.

- 4) `index_netcdf_variable_spectrum(nfreq)` : the corresponding PSD estimates for the time series associated with the input index NetCDF variable.

This variable is stored only if the **-vi=** argument has been specified when calling `comp_cross_spectrum2_3d`.

- 5) `index_netcdf_variable_netcdf_variable_squaredcoherency(nfreq, nlat, nlon)` : the corresponding squared coherency estimates for the multi-channel time series associated with the input NetCDF variable and the time series associated with the input index NetCDF variable. The square coherency is given on the closed interval $[0., 1.]$.

This variable is stored only if the **-vi=** argument has been specified when calling `comp_cross_spectrum2_3d`.

- 6) `index_netcdf_variable_netcdf_variable_phase(nfreq, nlat, nlon)` : the corresponding phase estimates of the cross-spectrum for the multi-channel time series associated with the input NetCDF variable and the time series associated with the input index NetCDF variable. In general terms, the phase gives the lag (negative values) or lead (positive values) of the `index_netcdf_variable` time series with respect to the `netcdf_variable` multi-channel time series at each frequency in radians.

Here, the phase is given in fractions of a circle (e.g., on the closed interval $[0., 1.]$) instead of being in radians. In other words, for phase values less than 0.5 the `index_netcdf_variable` leads the `netcdf_variable` and vice-versa for phase values greater than 0.5.

This variable is stored only if the **-vi=** argument has been specified when calling `comp_cross_spectrum2_3d`.

- 7) `edof(nfreq)` : the equivalent number of degrees of freedom of the PSD and CSD estimates of the time series associated with the input NetCDF variable.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum2_3d`.

- 8) `lower_factor(nfreq)` : the lower confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding lower confidence limit factors to get the lower limits of the $(1 - \text{critical_probability}) * 100\%$ confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum2_3d`.

- 9) `upper_factor(nfreq)` : the upper confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding upper confidence limit factors to get the upper limits of the $(1 - \text{critical_probability}) * 100\%$ confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_cross_spectrum2_3d`.

- 10) `coher_test(nfreq)` : the critical values for the statistical test of the squared coherency estimates. Squared coherency less than these computed critical values for a given frequency should be regarded as not significantly different from zero at the $\text{critical_probability} * 100\%$ significance level.

This variable is stored only if the **-conflim** and **-vi=** arguments have been specified when calling `comp_cross_spectrum2_3d`.

All the PSD and CSD estimates involving the input NetCDF variable `netcdf_variable` are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with

this input NetCDF variable even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For computing the power and cross-power spectra of a real multi-channel monthly time series from a tridimensional NetCDF variable called `sst` extracted from the file `sst.monthly.nc` with a monthly Nino34 SST index in a NetCDF variable called `sst_nino34` extracted from the file `sst.monthly.nino34.nc` and store the results in a NetCDF file named `cross_spectrum_sst_nino34_sst.nc`, use the following command:

```
$ comp_cross_spectrum2_3d \
  -f=sst.monthly.nc \
  -v=sst \
  -fi=sst.monthly.nino34.nc \
  -vi=sst_nino34 \
  -sl=128 \
  -o=cross_spectrum_sst_nino34_sst.nc \
  -nonormpsd \
  -nooverlap
```

2.15 comp_eof_3d

2.15.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.15.2 Latest revision

05/05/2026

2.15.3 Purpose

Compute an Empirical Orthogonal Function (EOF) analysis, also known as Principal Component Analysis (PCA) from a tridimensional variable extracted from a NetCDF dataset.

The procedure first transforms the input tridimensional NetCDF variable as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 2-D grid-mesh associated with the tridimensional NetCDF variable) and then performs an EOF analysis or PCA of this matrix [Jolliffe] [Jackson] .

The eigenvalues, eigenvectors and standardized Principal Components (PC) time series of the EOF analysis (or PCA) are computed by a full or partial Singular Value Decomposition (SVD) of the rectangular matrix of the observed variables [Jolliffe] [Bjornsson_Venegas] [Hannachi] [vonStorch_Zwiers]. These algorithms find directly square roots of eigenvalues (e.g., singular values of the data matrix **X**) and associated eigenvectors of the sums of squares and cross-products, covariance or correlation matrix between the observed variables (e.g., right singular vectors of **X**) without actually computing this symmetric semi-positive-definite matrix. The standardized PC time series, which are the left singular vectors of **X** are also directly obtained from the (partial) SVD of **X**.

Note that very fast randomized partial SVD algorithms [Halko_etal] [Li_etal] [Martinsson] can also be used, which will be highly efficient on huge datasets. See description of the `-alg=` argument for more details on the numerical algorithms available for performing the EOF analysis.

Optionally, the eigenvalues, eigenvectors and associated PC time series can be estimated with the help of a metric such that the results are weighted by the surface (or volume) associated with each cell in the grid-mesh associated with the input tridimensional NetCDF variable so that equal areas (or volumes) carry equal weights in the results of the EOF analysis (see the **-d=** argument description for more details).

At the user option, a moving block bootstrap approach [Efron] [Politis] can also be used to obtain confidence intervals for the explained variances by the PC time series and to test the stability of the computed leading EOFs [Jolliffe]. More precisely, statistics (e.g., confidence intervals, means, standard-deviations, skewness, ...) of explained variances, cosines of the angles between the pairwise observed and bootstrapped EOFs, as well as spectral and (scaled) Frobenius distances between the pairwise observed and bootstrapped leading EOF subspaces are computed and can be used to select the number of EOFs for further analysis. The **-nb=**, **-bl=**, **-bp=**, **-bm=**, **-bt=** and **-nobias** arguments allow the user to determine the exact form of the blockwise bootstrap algorithm which is used. At least one of the **-nb=**, **-bl=**, **-bp=**, **-bm=** and **-bt=** arguments must be present if you want to test the stability of the computed leading EOFs by this bootstrap approach and store its results in the output NetCDF dataset.

The notions of distance between two (EOF) subspaces of the same dimension used in the moving block bootstrap approach are based on the unique orthogonal projectors onto the ranges of these two (EOF) subspaces and the distances between the two square matrices representing these two orthogonal projectors as measured in some matrix norms [Golub_VanLoan]. Here, we use both the spectral (e.g., the 2-norm) and Frobenius matrix norms to quantify the distances between two (EOF) subspaces of the same dimension [Golub_VanLoan].

The spectral distance, **S_dist**, of two subspaces of the same dimension is simply equal to the sine of the largest principal angle between these two subspaces [Golub_VanLoan] and is always greater or equal to 0 and less or equal to 1. It is further possible to show that the spectral distance between two subspaces is equal to 0 if and only if the two subspaces are the same. On the other hand, if the spectral distance between two subspaces is equal to 1 then it exists at least one vector belonging to the first subspace, which is orthogonal to all vectors belonging to the second subspace and vice-versa.

On the other hand, it can be shown that the Frobenius distance, **F_dist**, between two subspaces of dimension n is equal to

$$\mathbf{F_dist} = \sqrt{2 \cdot \sum (\sin(PA(:n)))^2)}$$

where the elements of the n -vector $PA(:)$ are the n principal angles between the two subspaces of dimension n . The Frobenius distance is equal to 0 if the two subspaces are the same and is always less than $\sqrt{2 \cdot n}$ and this maximum is attained only when the two subspaces are orthogonal to each other. Since this upper bound is a function of the dimension n of the two subspaces, it is further convenient to define the “scaled” Frobenius distance of two subspaces of dimension n as the ratio between the Frobenius distance and its upper bound. In this way, the “scaled” Frobenius distance is always greater or equal to 0 and less or equal to 1, as the spectral distance, and can also be used to compare “scaled” Frobenius distances between pairs of EOF subspaces of different dimensions in order to determine the most stable EOF subspaces with the help of the moving block bootstrap method.

An output NetCDF dataset containing singular values (e.g., square roots of the eigenvalues, which are equal to the standard-deviations of the corresponding PC time series), percentages of variance explained by each PC time series, eigenvectors scaled by the corresponding singular values and standardized PC time series is created. The scaled eigenvectors are repacked as a tridimensional variable in the output NetCDF dataset. If the bootstrap approach is requested, the output NetCDF dataset will also contain confidence intervals for the explained variances by the PC time series as well as statistics for explained variances, cosines of the angles between the pairwise observed and bootstrapped EOFs and distances between the pairwise observed and bootstrapped leading EOF subspaces.

You should use EOF analysis if you are interested in summarizing data and/or detecting linear relationships between the observed variables [Jolliffe] [Jackson] [vonStorch_Zwiers]. EOF or PCA analysis can also be used to reduce the number of variables or the noise in a dataset before a regression, cluster or Maximum Covariance Analysis (MCA). More specifically, the first k standardized PC time series and associated eigenvectors (scaled by the corresponding singular values) give a least-squares solution to the model

$$\mathbf{X} = \mathbf{AB} + \mathbf{E}$$

where

- **X** is the *ntime* by *nv* matrix of observed variables
- **A** is the *ntime* by *k* matrix of the first *k* (standardized) PC time series
- **B** is the *k* by *nv* matrix of the first *k* (scaled) eigenvectors (stored rowwise)
- **E** is an *ntime* by *nv* matrix of residuals

and you want to minimize the squared Frobenius norm of **E** (e.g., the sum of all the squared elements of **E**).

Refer to `comp_invert_eof_3d`, if you want to compute such approximation of your dataset and to `comp_ortho_rot_eof_3d`, `comp_filt_rot_eof_3d` or `comp_loess_rot_eof_3d` if you want to perform an orthogonal rotation of all or selected of the computed standardized PC time series in order to simplify their physical interpretations [Jolliffe] [Jackson] [Wills_etal] .

Refer to `comp_svd_3d`, `comp_svd2_3d`, `comp_reg_3d` and `comp_reg_4d`, for more details on MCA and regression procedures available in NCSTAT, respectively.

If your data contains missing values use `comp_eof_miss_3d` instead of `comp_eof_3d` to estimate approximate (scaled) eigenvectors, (standardized) PC time series and, in addition, the missing values in your dataset.

Finally, if the NetCDF variable is fourdimensional use `comp_eof_4d` instead of `comp_eof_3d`.

This procedure is parallelized if OpenMP is used. Moreover, as noted above, this procedure may use (randomized) partial SVD algorithms, which are highly efficient on huge datasets if you are interested only in the few leading terms of the SVD of the data matrix **X**.

2.15.4 Further Details

Usage

```
$ comp_eof_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -r=resolution                (optional : r2, r4) \
  -b=nlon_orca, nlat_orca      (optional) \
  -x=lon1,lon2                 (optional) \
  -y=lat1,lat2                 (optional) \
  -t=time1,time2               (optional) \
  -a=type_of_analysis          (optional : scp, cov, cor) \
  -c=input_climatology_netcdf_file (optional) \
  -d=distance                  (optional : dist2, ident) \
  -alg=algorithm               (optional : svd, inviter, deflate, rsvd) \
  -n=number_of_eof             (optional) \
  -o=output_eof_netcdf_file    (optional) \
  -bm=bootstrap_method         (optional : normal, percentile) \
  -nb=number_of_shuffles       (optional) \
  -bp=bootstrap_periodicity    (optional) \
  -bl=bootstrap_block_length   (optional) \
  -bt=bootstrap_trend_length   (optional) \
  -pro=critical_probability    (optional : 0.0 > 1.) \
  -mi=missing_value            (optional) \
  -ortho                       (optional) \
  -explvar                     (optional) \
  -nobias                      (optional) \
  -double                      (optional) \
  -bigfile                     (optional) \
```

(continues on next page)

(continued from previous page)

-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- g=** the *grid_type* is set to `n` which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to `n`) the resolution is assumed to be `r2`
- b=** if **-g=** is not set to `n`, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to `scp`. This means that the eigenvectors and eigenvalues are computed from the sums of squares and cross-products matrix between the observed variables
- c=** an *input_climatology_netcdf_file* is not needed if the *type_of_analysis* is set to `scp`
- d=** the *distance* is set to `dist2`. This means that distances and scalar products in the EOF analysis are computed with the diagonal metric associated with the 2-D grid-mesh associated with the input NetCDF variable
- alg=** the *algorithm* option is set to `inviter`. This means that the EOF model is computed by a partial SVD analysis of the matrix of the observed variables using an inverse iteration algorithm
- n=** *number_of_eofs* is set to 10 and a 10-component EOF model is stored in the output NetCDF file *output_eof_netcdf_file*
- o=** the *output_eof_netcdf_file* is named `eof_netcdf_variable.nc`
- bm=** *bootstrap_method* is set to `normal`. This means that bootstrap confidence intervals of explained variances by the EOFs are based on asymptotic normality
- nb=** *number_of_shuffles* is set to 99. This means that 99 bootstrap samples are generated in the moving block bootstrap algorithm for computing the statistics of explained variance statistics and their sampling distributions as well as statistics related to the stability of the computed EOFs
- bp=** the time series are assumed to be stationary and *bootstrap_periodicity* is set to 1 in the moving block bootstrap procedure for testing the significance of the singular triplets. This means that the blocks in the bootstrap algorithm are not forced to begin at specific observations. Use this parameter if the time series are cyclostationary, see the remarks below for further details
- bl=** *bootstrap_block_length* is set to *bootstrap_periodicity*.2
- bt=** a stationary moving block bootstrap algorithm is used and the *bootstrap_trend_length* parameter has no effect
- pro=** the *critical_probability* is set to 0.05, which means that the bootstrap confidence intervals for the explained variance statistics are 95% confidence intervals
- mi=** the *missing_value* attribute in the output NetCDF file is set to 1.e+20

- ortho** the computed EOFs and associated PC time series are not automatically reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods, e.g., if **-alg=inviter** or **-alg=deflate**
- explvar** the **-n=** option specifies the number of eofs to be computed and stored in the output NetCDF file. If **-explvar** is activated, the **-n=** option specifies the minimum value of explained variance by the EOF model for selecting the order (e.g., the number of components) of this EOF model
- nobias** biased bootstrap confidence intervals for explained variances are computed. However, if **-nobias** is activated, unbiased bootstrap confidence intervals for explained variances are computed
- double** the results of the EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which an EOF analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to the *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix before computing the EOF analysis. The scale factors associated with the 2-D grid-mesh of this NetCDF variable (needed if **-d=dist2** is specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the geographical domain used in the EOF analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*) which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF analysis.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_eof_3d*.

- 4) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to estimate eigenvalues, eigenvectors and PC time series.
The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.
- 5) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask. If your dataset has missing values, use [comp_eof_miss_3d](#) instead of *comp_eof_3d*.

- 6) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

In this case, the duplicate points from the ORCA grid are removed before the EOF analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe. On output, the duplicate points are restored when writing the EOFs (e.g., the eigenvectors), if the geographical domain of the input NetCDF variable is the whole globe.

If **-g=** is set to `n`, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typedmask*, *e1grid_type* and *e2grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_3d* if the 2-D grid-mesh is regular or gaussian.

- 7) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (e.g., if the input NetCDF variable, *netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
- **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
 - **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.
- 8) If the input NetCDF variable is from an experiment with the NEMO model, but the resolution is not `r2` or `r4`, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 9) The **-a=** argument specifies if the observed variables are centered or standardized with an input climatology (specified with the **-c=** argument) before the EOF analysis. If:
- **-a=scp**, the EOF analysis is done on the raw data
 - **-a=cov**, the EOF analysis is done on the anomalies
 - **-a=cor**, the EOF analysis is done on the standardized anomalies.

By default, the **-a=** argument is set to `scp`.

- 10) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.
- 11) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 12) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 13) The **-d=distance** argument specifies the metric and scalar product used in the EOF analysis. If:
- **-d=dist2**, the EOF analysis is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident**, the EOF analysis is done with the identity metric, e.g., the Euclidean distance and the usual scalar product is used in the EOF analysis.

By default, the **-d=** argument is set to `dist2`.

- 14) The **-alg=algorithm** argument determines how eigenvalues, eigenvectors and PC time series are computed. If:
- **-alg=svd**, a full SVD of the data matrix is computed, even if you ask only for the leading eigenvectors
 - **-alg=inviter**, a partial SVD of the data matrix is computed by inverse iteration
 - **-alg=deflate**, a partial SVD of the data matrix is computed by a deflation technique

- **-alg=rsvd**, a partial SVD of the data matrix is computed by a randomized algorithm.

All algorithms are parallelized if OpenMP is used. The default is **-alg=inviter** since computing a partial SVD is generally much faster than computing a full SVD. But, **-alg=deflate** is generally as fast as **-alg=inviter**. Note that **-alg=rsvd** is generally much faster than all other options, but the computed eigenvectors and PC time series may be less accurate depending on the shape of the distribution of the singular values of the input matrix.

- 15) The **-n=number_of_eofs** argument specifies the number of components of the EOF model which must be stored (and also computed if **-alg=inviter**, **-alg=deflate** or **-alg=rsvd** is specified) in the output NetCDF file specified by the **-o=** argument. See also the description of the **-explvar** argument below.
- 16) If the **-explvar** argument is activated, the number specified with the **-n=** argument is not the number of requested EOFs, but the minimum of variance that these EOFs must describe. The number of EOFs is then determined by reference with the minimum of explained variance required by the **-n=** argument. Express this explained variance in percentage (e.g., with an integer from 0 to 100).

If **-explvar** is specified, the **-n=** argument must be less or equal to 100. Note also that, if a randomized partial SVD algorithm is used (e.g., if **-alg=rsvd** is specified), the number of computed EOFs is limited to 200 even if these 200 EOFs describe less variance than the minimum requested with the **-n=** argument.

- 17) If the **-ortho** argument is used, the computed EOFs and associated PC time series are always reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods. By default, they are only partially orthogonalized if the computed singular values are not well separated. Note that this argument has no effect if **-alg=svd** or **-alg=rsvd**.
- 18) If any of the **-nb=**, **-bl=**, **-bp=** and **-bm=** arguments is specified, a moving block bootstrap algorithm is used to test the stability of the computed EOFs and help to select the number of significant EOFs for further analysis.
- 19) The **-bm=bootstrap_method** argument specifies how bootstrap confidence intervals for explained variances are computed. If:
 - **-bm=normal**, bootstrap confidence intervals of explained variances by the EOFs are based on asymptotic normality [Efron] [Politis] . These standard intervals will be bias-corrected [Politis] if the **-nobias** argument is also specified.
 - **-bm=percentile**, bootstrap confidence intervals of explained variances by the EOFs are based on the percentile or bias-corrected percentile methods [Efron] . The bias-corrected percentile method will be used if the **-nobias** argument is also specified.

Note that the percentile method requires a larger number of shuffles (specified with the **-nb=** argument), especially if the *critical_probability* (specified with the **-pro=** argument) is small. More precisely, *number_of_shuffles.critical_probability.0.5* must be greater or equal to 1. in the case of the percentile method. If it is not the case, the procedure will exit with an error message. As an illustration, with the default *critical_probability* (e.g., 0.05), *number_of_shuffles* must be at least 40 for using the percentile method for estimating confidence intervals for explained variances by the EOFs.

By default, the **-bm=** argument is set to `normal`.

- 20) The **-nb=number_of_shuffles** argument specifies the number of shuffles used in the bootstrap algorithm (by default 99).
- 21) The **-bp=bootstrap_periodicity** argument specifies that the index, *i*, of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where *j* is a random positive integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. This parameter is useful if the time series are cyclostationary instead of stationary.

By default, *bootstrap_periodicity* is set to 1 so that the time series is assumed to be stationary (see the description of the **-bt=** argument if you want to relax this strong assumption in the bootstrap algorithm).

- 22) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater or equal to the *bootstrap_periodicity* and less than the length of

the time series. By default, *bootstrap_block_length* is set to *bootstrap_periodicity*.2. If **-a=cov** or **-a=cor** is specified, it is highly recommended to set *bootstrap_block_length* as a multiple of the periodicity in the data as this will take properly into account the cyclostationary of the analyzed time series.

- 23) The **-bt=bootstrap_trend_length** argument specifies that a non-stationary bootstrap algorithm must be used and sets the length of the trend smoother, *bootstrap_trend_length*, in this non-stationary bootstrap algorithm. The non-stationary bootstrap algorithm consists of two steps:

- First, trends are extracted from the time series by using a LOESS smoother [Cleveland][Cleveland_Devlin]. This LOESS smoother is similar to the one performed in *comp_trend_3d* and the **-bt=** argument here has the the same meaning as the **-nt=** argument in *comp_trend_3d*.
- Secondly, residuals from the estimated trends are computed and are used in a stationary moving block bootstrap algorithm. The final non-stationary bootstrapped time series are finally generated by adding the trend components to the bootstrapped residuals time series.

The value of *bootstrap_trend_length* should be an odd integer greater than or equal to 3. As *bootstrap_trend_length* increases the values of the trend component become smoother and the amplitude of the residuals increases, generating more variability in the bootstrapped time series.

- 24) The **-pro=critical_probability** argument specifies the critical probability which is used to determine the lower and upper bounds of the confidence intervals for explained variances by the EOFs if the bootstrap method is used. More precisely, Lower and upper $(1 - \text{critical_probability}) \cdot 100\%$ bootstrap confidence bounds for explained variances are computed based on asymptotic normality or the percentile method depending on the value of the **-bm=bootstrap_method** argument. *critical_probability* must be greater than 0. and less than 1.. The default value is 0.05, meaning that 95% confidence intervals are computed by default.

- 25) If **-nobias** is specified, unbiased bootstrap confidence intervals for explained variances are computed. If the **-nobias** argument is absent, biased bootstrap confidence intervals are estimated by default.

If **-bm=percentile** is specified, this means that the confidence intervals will be obtained by the bias-corrected percentile method instead of the percentile method.

- 26) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.

- 27) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 28) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 29) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 30) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 31) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 32) For more details on PCA and EOF analysis, the bootstrap approach, distances between subspaces, the LOESS smoother and randomized SVD algorithms, see
 - “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
 - “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
 - “Matrix Computations”, by Golub, G.H., and Van Loan, C., The Johns Hopkins University Press, Baltimore, MD., 4rd Edition, 756 pp., 2013. ISBN: 978-1-4214-0794-4
 - “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
 - “A primer for EOF analysis of climate data”, by Hannachi, A., Reading University, Reading, UK, 33pp, 2004. <http://www.met.reading.ac.uk/~han/Monitor/eofprimer.pdf>
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
 - “Nonparametric standard-errors and confidence intervals” By Efron, B., The Canadian Journal of Statistics, 9, 2, 139-172, (1981). <https://doi.org/10.2307/3314608>
 - “Computer-Intensive Methods in Statistical Analysis” By Politis, D.N., IEEE Signal Processing Magazine, 15, 1, 39-55, (January 1998). <https://ieeexplore.ieee.org/document/647042>
 - “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: 10.1080/01621459.1979.10481038
 - “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: 10.1080/01621459.1988.10478639
 - “Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions”, By Halko, N., Martinsson, P.G., and Tropp, J.A., SIAM Rev., 53, 217-288, 2011. <https://epubs.siam.org/doi/abs/10.1137/090771806>
 - “Algorithm 971: An implementation of a randomized algorithm for principal component analysis”, by Li, H., Linderman, G.C., Szlam, A., Stanton, K.P., Kluger, Y., and Tygert, M., ACM Trans. Math. Softw. 43, 3, Article 28 (January 2017). <https://pubmed.ncbi.nlm.nih.gov/28983138>
 - “Randomized methods for matrix computations”, by Martinsson, P.G., arXiv.1607.01649, 2019. <https://arxiv.org/abs/1607.01649>
 - “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>

Outputs

`comp_eof_3d` creates an output NetCDF file that contains the standardized PC time series, scaled eigenvectors and the (square roots of) eigenvalues of the EOF analysis. The number of PC time series, eigenvectors and eigenvalues stored in the output NetCDF dataset is determined by the `-n=number_of_eofs` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_eof(number_of_eofs, nlat, nlon)` : the selected eigenvectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix between the observed variables. The eigenvectors are sorted by descending order of the associated eigenvalues. The eigenvectors are scaled such that they give the scalar products (`-a=scp`), covariances (`-a=cov`) or correlations (`-a=cor`) between the original observed variables and the associated PC time series.

The eigenvectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values. If this is a problem, you can use `comp_norm_3d` for restricting the geographical domain in the input dataset before using `comp_eof_3d`.

- 2) `netcdf_variable_pc(ntime, number_of_eofs)` : the PC time series sorted by descending order of the eigenvalues (e.g., the squares of the singular values of the input matrix of observed variables).

The PC time series are always standardized to unit variance.

- 3) `netcdf_variable_sing(number_of_eofs)` : the singular values of the input data matrix in decreasing order, up to a constant scaling factor (equals to the square root of $1/ntime$) and taking into account the effect of the metric used in the EOF analysis.

More precisely, these singular values are the square roots of the eigenvalues of the sums of squares and cross-products (`-a=scp`) or covariance (`-a=cov`) or correlation (`-a=cor`) matrix between the observed variables.

These eigenvalues (e.g., the squares of the singular values) are equal to the (raw) variance described by the PC time series.

- 4) `netcdf_variable_var(number_of_eofs)` : the proportions of variance explained by each PC time series (given between 0. and 1. with 1. corresponding to 100%).

Furthermore, if one of the `-nb=`, `-bl=`, `-bp=` or `-bm=` arguments is specified when calling `comp_eof_3d`, the following NetCDF variables are computed and also stored in the output NetCDF file:

- 1) `netcdf_variable_var_boot_lower_bound(number_of_eofs)` : lower (1 - critical_probability)“100“% bootstrap confidence bounds for explained variances. By default, *critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the `-pro=critical_probability` argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the `-nobias=` option is used.
- 2) `netcdf_variable_var_boot_upper_bound(number_of_eofs)` : upper (1 - critical_probability)“100“% bootstrap confidence bounds for explained variances. By default, *critical_probability is set to 0.05, such that 95%\`% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the `**pro=**\`critical_probability` argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the `-nobias=` option is used.

- 3) *netcdf_variable_var_boot_mean* (*number_of_eofs*) : the mean of bootstrapped explained variances.
- 4) *netcdf_variable_var_boot_var* (*number_of_eofs*) : the variance of bootstrapped explained variances.
- 5) *netcdf_variable_var_boot_std* (*number_of_eofs*) : the standard-deviation of bootstrapped explained variances.
- 6) *netcdf_variable_var_boot_skew* (*number_of_eofs*) : the skewness of bootstrapped explained variances.
- 7) *netcdf_variable_var_boot_kurt* (*number_of_eofs*) : the kurtosis of bootstrapped explained variances.
- 8) *netcdf_variable_var_boot_min* (*number_of_eofs*) : the minimum of bootstrapped explained variances.
- 9) *netcdf_variable_var_boot_max* (*number_of_eofs*) : the maximum of bootstrapped explained variances.
- 10) *netcdf_variable_cos_vec_boot_mean* (*number_of_eofs*) : the mean of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 11) *netcdf_variable_cos_vec_boot_var* (*number_of_eofs*) : the variance of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 12) *netcdf_variable_cos_vec_boot_std* (*number_of_eofs*) : the standard-deviation of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 13) *netcdf_variable_cos_vec_boot_skew* (*number_of_eofs*) : the skewness of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 14) *netcdf_variable_cos_vec_boot_kurt* (*number_of_eofs*) : the kurtosis of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 15) *netcdf_variable_cos_vec_boot_min* (*number_of_eofs*) : the minimum of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 16) *netcdf_variable_cos_vec_boot_max* (*number_of_eofs*) : the maximum of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 17) *netcdf_variable_spect_dist_subspace_boot_mean* (*number_of_eofs*) : the mean of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 18) *netcdf_variable_spect_dist_subspace_boot_var* (*number_of_eofs*) : the variance of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 19) *netcdf_variable_spect_dist_subspace_boot_std* (*number_of_eofs*) : the standard-deviation of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 20) *netcdf_variable_spect_dist_subspace_boot_skew* (*number_of_eofs*) : the skewness of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 21) *netcdf_variable_spect_dist_subspace_boot_kurt* (*number_of_eofs*) : the kurtosis of the spectral distances between the pairwise observed and bootstrapped EOF subspaces.
- 22) *netcdf_variable_spect_dist_subspace_boot_min* (*number_of_eofs*) : the minimum of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 23) *netcdf_variable_spect_dist_subspace_boot_max* (*number_of_eofs*) : the maximum of the spectral distances between pairwise observed and bootstrapped EOF subspaces.

- 24) *netcdf_variable_frob_dist_subspace_boot_mean*(number_of_eofs) : the mean of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 25) *netcdf_variable_frob_dist_subspace_boot_var*(number_of_eofs) : the variance of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 26) *netcdf_variable_frob_dist_subspace_boot_std*(number_of_eofs) : the standard-deviation of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 27) *netcdf_variable_frob_dist_subspace_boot_skew*(number_of_eofs) : the skewness of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 28) *netcdf_variable_frob_dist_subspace_boot_kurt*(number_of_eofs) : the kurtosis of the (scaled) Frobenius distances between the pairwise observed and bootstrapped EOF subspaces.
- 29) *netcdf_variable_frob_dist_subspace_boot_min*(number_of_eofs) : the minimum of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 30) *netcdf_variable_frob_dist_subspace_boot_max*(number_of_eofs) : the maximum of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.

Examples

- 1) For computing an EOF analysis from the NetCDF file `HadISST1_2m_197902_200501_sst.nc`, which includes a NetCDF variable `sst`, and store the results in a NetCDF file named `eof_HadISST1_2m_197902_200501_sst_oi.nc`, use the following command (the analysis is done on the raw data without removing the annual cycle) :

```
$ comp_eof_3d \  
-f=HadISST1_2m_197902_200501_sst.nc \  
-v=sst \  
-m=mask_HadISST1_sst.nc \  
-g=n \  
-a=scp \  
-n=20 \  
-o=eof_HadISST1_2m_197902_200501_sst_oi.nc
```

- 2) For computing an EOF analysis from the NetCDF file `ST7_1m_0101_20012_grid_T_sosstsst.nc`, which includes a NetCDF variable `sosstsst` from a numerical simulation with the ORCA R2 configuration of the NEMO ocean model, and store the results in a NetCDF file named `eof_sosstsst.nc`, use the following command (the analysis is done on the data after removing a climatology) :

```
$ comp_eof_3d \  
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \  
-v=sosstsst \  
-c=clim_ST7_1m_0101_20012_grid_T_sosstsst.nc \  
-a=cov \  
-m=meshmask.orca2.nc \  
-g=t
```

2.16 comp_eof_4d

2.16.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.16.2 Latest revision

05/05/2026

2.16.3 Purpose

Compute an Empirical Orthogonal Function (EOF) analysis, also known as Principal Component Analysis (PCA) from a fourdimensional variable extracted from a NetCDF dataset.

The procedure first transforms the input fourdimensional NetCDF variable as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 3-D grid-mesh associated with the fourdimensional NetCDF variable) and then performs an EOF analysis or PCA of this matrix [Jolliffe] [Jackson] .

The eigenvalues, eigenvectors and standardized Principal Components (PC) time series of the EOF analysis (or PCA) are computed by a full or partial Singular Value Decomposition (SVD) of the rectangular matrix of the observed variables [Jolliffe] [Bjornsson_Venegas] [Hannachi] [vonStorch_Zwiers]. These algorithms find directly square roots of eigenvalues (e.g., singular values of the data matrix **X**) and associated eigenvectors of the sums of squares and cross-products, covariance or correlation matrix between the observed variables (e.g., right singular vectors of **X**) without actually computing this symmetric semi-positive-definite matrix. The standardized PC time series, which are the left singular vectors of **X** are also directly obtained from the (partial) SVD of **X**.

Note that very fast randomized partial SVD algorithms [Halko_etal] [Li_etal] [Martinsson] can also be used, which will be highly efficient on huge datasets. See description of the **-alg=** argument for more details on the numerical algorithms available for performing the EOF analysis.

Optionally, the eigenvalues, eigenvectors and associated PC time series can be estimated with the help of a metric such that the results are weighted by the surface (or volume) associated with each cell in the grid-mesh associated with the input fourdimensional NetCDF variable so that equal areas (or volumes) carry equal weights in the results of the EOF analysis (see the **-d=** argument description for more details).

At the user option, a moving block bootstrap approach [Efron] [Politis] can also be used to obtain confidence intervals for the explained variances by the PC time series and to test the stability of the computed leading EOFs [Jolliffe]. More precisely, statistics (e.g., confidence intervals, means, standard-deviations, skewness, ...) of explained variances, cosines of the angles between the pairwise observed and bootstrapped EOFs, as well as spectral and (scaled) Frobenius distances between the pairwise observed and bootstrapped leading EOF subspaces are computed and can be used to select the number of EOFs for further analysis. The **-nb=**, **-bl=**, **-bp=**, **-bm=**, **-bt=** and **-nobias** arguments allow the user to determine the exact form of the blockwise bootstrap algorithm which is used. One of the **-nb=**, **-bl=**, **-bp=**, **-bm=** and **-bt=** arguments must be present if you want to test the stability of the computed leading EOFs by this bootstrap approach and store its results in the output NetCDF dataset.

The notions of distance between two (EOF) subspaces of the same dimension used in the moving block bootstrap approach are based on the unique orthogonal projectors onto the ranges of these two (EOF) subspaces and the distances between the two square matrices representing these two orthogonal projectors as measured in some matrix norms [Golub_VanLoan]. Here, we use both the spectral (e.g., the 2-norm) and Frobenius matrix norms to quantify the distances between two (EOF) subspaces of the same dimension [Golub_VanLoan].

The spectral distance, **S_dist**, of two subspaces of the same dimension is simply equal to the sine of the largest principal angle between these two subspaces [Golub_VanLoan] and is always greater or equal to 0 and less or equal to 1. It is further possible to show that the spectral distance between two subspaces is equal to 0 if and only if the two subspaces are the same. On the other hand, if the spectral distance between two subspaces is equal to 1 then it exists at least one vector belonging to the first subspace, which is orthogonal to all vectors belonging to the second subspace and vice-versa.

On the other hand, it can be shown that the Frobenius distance, **F_dist**, between two subspaces of dimension *n* is equal to

$$\mathbf{F_dist} = \sqrt{2 \cdot \sum (\sin(PA(:n)))^2)}$$

where the elements of the n -vector $PA(:)$ are the n principal angles between the two subspaces of dimension n . The Frobenius distance is equal to 0 if the two subspaces are the same and is always less than $\sqrt{2 \cdot n}$ and this maximum is attained only when the two subspaces are orthogonal to each other. Since this upper bound is a function of the dimension n of the two subspaces, it is further convenient to define the “scaled” Frobenius distance of two subspaces of dimension n as the ratio between the Frobenius distance and its upper bound. In this way, the “scaled” Frobenius distance is always greater or equal to 0 and less or equal to 1, as the spectral distance, and can also be used to compare “scaled” Frobenius distances between pairs of EOF subspaces of different dimensions in order to determine the most stable EOF subspaces with the help of the moving block bootstrap method.

An output NetCDF dataset containing singular values (e.g., square roots of the eigenvalues, which are equal to the standard-deviations of the corresponding PC time series), percentages of variance explained by each PC time series, eigenvectors scaled by the corresponding singular values and standardized PC time series is created. The scaled eigenvectors are repacked as a fourdimensional variable in the output NetCDF dataset. If the bootstrap approach is requested, the output NetCDF dataset will also contain confidence intervals for the explained variances by the PC time series as well as statistics for explained variances, cosines of the angles between the pairwise observed and bootstrapped EOFs and distances between the pairwise observed and bootstrapped leading EOF subspaces.

You should use EOF analysis if you are interested in summarizing data and/or detecting linear relationships between the observed variables [Jolliffe] [Jackson] [vonStorch_Zwiers]. EOF or PCA analysis can also be used to reduce the number of variables or the noise in a dataset before a regression, cluster or Maximum Covariance Analysis (MCA). More specifically, the first k standardized PC time series and associated eigenvectors (scaled by the corresponding singular values) give a least-squares solution to the model

$$\mathbf{X} = \mathbf{AB} + \mathbf{E}$$

where

- \mathbf{X} is the $ntime$ by nv matrix of observed variables
- \mathbf{A} is the $ntime$ by k matrix of the first k (standardized) PC time series
- \mathbf{B} is the k by nv matrix of the first k (scaled) eigenvectors (stored rowwise)
- \mathbf{E} is an $ntime$ by nv matrix of residuals

and you want to minimize the squared Frobenius norm of \mathbf{E} (e.g., the sum of all the squared elements of \mathbf{E}).

Refer to `comp_invert_eof_4d`, if you want to compute such approximation of your dataset and to `comp_ortho_rot_eof_4d`, `comp_filt_rot_eof_4d` or `comp_loess_rot_eof_4d` if you want to perform an orthogonal rotation of all or selected of the computed standardized PC time series in order to simplify their physical interpretations [Jolliffe] [Jackson] [Wills_etal].

Refer to `comp_svd_3d`, `comp_svd2_3d`, `comp_reg_3d` and `comp_reg_4d` for more details on MCA and regression procedures available in NCSTAT, respectively.

Finally, if the NetCDF variable is tridimensional use `comp_eof_3d` instead of `comp_eof_4d`.

This procedure is parallelized if OpenMP is used. Moreover, as noted above, this procedure may use (randomized) partial SVD algorithms, which are highly efficient on huge datasets if you are interested only in the few leading terms of the SVD of the data matrix \mathbf{X} .

2.16.4 Further Details

Usage

```
$ comp_eof_4d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file \
```

(continues on next page)

(continued from previous page)

```

-g=grid_type                (optional : n, t, u, v, w, f) \
-r=resolution                (optional : r2, r4) \
-b=nlon_orca, nlat_orca, nlevel_orca (optional) \
-x=lon1,lon2                (optional) \
-y=lat1,lat2                (optional) \
-z=level1,level2            (optional) \
-t=time1,time2              (optional) \
-a=type_of_analysis         (optional : scp, cov, cor) \
-c=input_climatology_netcdf_file (optional) \
-d=distance                 (optional : dist2, dist3, ident) \
-alg=algorithm              (optional : svd, inviter, deflate, rsvd) \
-n=number_of_eofs           (optional) \
-o=output_eof_netcdf_file   (optional) \
-bm=bootstrap_method        (optional : normal, percentile) \
-nb=number_of_shuffles      (optional) \
-bp=bootstrap_periodicity   (optional) \
-bl=bootstrap_block_length  (optional) \
-bt=bootstrap_trend_length  (optional) \
-pro=critical_probability   (optional : 0.0 > 1.) \
-mi=missing_value          (optional) \
-ortho                      (optional) \
-explvar                    (optional) \
-nobias                     (optional) \
-double                     (optional) \
-bigfile                    (optional) \
-hdf5                       (optional) \
-cdf5                       (optional) \
-tlimited                    (optional)

```

By default

- g=** the *grid_type* is set to *n* which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO ocean model (e.g., if **-n=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-n=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *scp*. This means that the eigenvectors and eigenvalues are computed from the sums of squares and cross-products matrix between the observed variables
- c=** an *input_climatology_netcdf_file* is not needed if the *type_of_analysis* is set to *scp*
- d=** the *distance* is set to *dist3*. This means that distances and scalar products in the EOF analysis are computed with the diagonal metric associated with the 3-D grid-mesh associated with the input NetCDF variable

- alg=** the *algorithm* option is set to `inviter`. This means that the EOF model is computed by a partial SVD analysis of the matrix of the observed variables using an inverse iteration algorithm
- n=** *number_of_eofs* is set to 10 and a 10-component EOF model is stored in the output NetCDF file *output_eof_netcdf_file*
- o=** the *output_eof_netcdf_file* is named `eof_netcdf_variable.nc`
- bm=** *bootstrap_method* is set to `normal`. This means that bootstrap confidence intervals of explained variances by the EOFs are based on asymptotic normality
- nb=** *number_of_shuffles* is set to 99. This means that 99 bootstrap samples are generated in the moving block bootstrap algorithm for computing the statistics of explained variance statistics and their sampling distributions as well as statistics related to the stability of the computed EOFs
- bp=** the time series are assumed to be stationary and *bootstrap_periodicity* is set to 1 in the moving block bootstrap procedure for testing the significance of the singular triplets. This means that the blocks in the bootstrap algorithm are not forced to begin at specific observations. Use this parameter if the time series are cyclostationary, see the remarks below for further details
- bl=** *bootstrap_block_length* is set to *bootstrap_periodicity*.2
- bt=** a stationary moving block bootstrap algorithm is used and the *bootstrap_trend_length* parameter has no effect
- pro=** the *critical_probability* is set to 0.05, which means that the bootstrap confidence intervals for the explained variance statistics are 95% confidence intervals
- mi=** the *missing_value* attribute in the output NetCDF file is set to 1.e+20
- ortho** the computed EOFs and associated PC time series are not automatically reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods, e.g., if **-alg=inviter** or **-alg=deflate**
- explvar** the **-n=** option specifies the number of eofs to be computed and stored in the output NetCDF file. If **-explvar** is activated, the **-n=** option specifies the minimum value of explained variance by the EOF model for selecting the order (e.g., the number of components) of this EOF model
- nobias** biased bootstrap confidence intervals for explained variances are computed. However, if **-nobias** is activated, unbiased bootstrap confidence intervals for explained variances are computed
- double** the results of the EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which an EOF analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.

- 2) The argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to the *netcdf_variable* for transforming this fourdimensional NetCDF variable as a rectangular matrix before computing the EOF analysis. The scale factors associated with the 3-D grid-mesh of this NetCDF variable (needed if **-d=dist2** or **dist3** are specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.
- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the geographical domain used in the EOF analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typedmask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit*, *lat2_Northern_limit*, *level1_First_level* and *level2_Last_level*) which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used in the EOF analysis.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_eof_4d*.

- 4) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to estimate eigenvalues, eigenvectors and PC time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 5) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

In this case, the duplicate points from the ORCA grid are removed before the EOF analysis, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variable covers the whole globe. On output, the duplicate points are restored when writing the EOFs (e.g., the eigenvectors), if the geographical domain of the input NetCDF variable is the whole globe.

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typedmask*, *e1grid_type*, *e2grid_type* and *e3grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by [comp_clim_4d](#) if the 3-D grid-mesh is regular or gaussian.

- 7) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variable is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
 - **-r=r2**, the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
 - **-r=r4**, the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.
- 8) If the input NetCDF variable is from an experiment with the NEMO model, but the resolution is not *r2* or *r4*, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 9) The **-a=** argument specifies if the observed variables are centered or standardized with an input climatology (specified with the **-c=** argument) before the EOF analysis. If:
 - **-a=scp**, the EOF analysis is done on the raw data
 - **-a=cov**, the EOF analysis is done on the anomalies
 - **-a=cor**, the EOF analysis is done on the standardized anomalies.

By default, the **-a=** argument is set to `scp`.

- 10) The `input_climatology_netcdf_file` specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.
- 11) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (`time1` if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 12) The geographical shapes of the `netcdf_variable` (in the `input_netcdf_file`), the mask (in the `input_mesh_mask_netcdf_file`), the scale factors (in the `input_mesh_mask_netcdf_file`), and the climatology (in the `input_climatology_netcdf_file`) must agree.
- 13) The **-d=distance** argument specifies the metric and scalar product used in the EOF analysis. If:
 - **-d=dist2**, the EOF analysis is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid point is weighted accordingly to the surface associated with it)
 - **-d=dist3**, the EOF analysis is done with the diagonal distance associated with the whole 3D grid-mesh (e.g., each grid point is weighted accordingly to the volume or weight associated with it)
 - **-d=ident**, the EOF analysis is done with the identity metric, e.g., the Euclidean distance and the usual scalar product are used in the EOF analysis.

By default, the **-d=** argument is set to `dist3`.

- 14) The **-alg=algorithm** argument determines how eigenvalues, eigenvectors and PC time series are computed. If:
 - **-alg=svd**, a full SVD of the data matrix is computed, even if you ask only for the leading eigenvectors
 - **-alg=inviter**, a partial SVD of the data matrix is computed by inverse iteration
 - **-alg=deflate**, a partial SVD of the data matrix is computed by a deflation technique
 - **-alg=rsvd**, a partial SVD of the data matrix is computed by a randomized algorithm.

All algorithms are parallelized if OpenMP is used. The default is **-alg=inviter** since computing a partial SVD is generally much faster than computing a full SVD. But, **-alg=deflate** is generally as fast as **-alg=inviter**. Note that **-alg=rsvd** is generally much faster than all other options, but the computed eigenvectors and PC time series may be less accurate depending on the shape of the distribution of the singular values of the input matrix.

- 15) The **-n=number_of_eofs** argument specifies the number of components of the EOF model which must be stored (and also computed if **-alg=inviter**, **-alg=deflate** or **-alg=rsvd** is specified) in the output NetCDF file specified by the **-o=** argument. See also the description of the **-explvar** argument below.
- 16) If the **-explvar** argument is activated, the number specified with the **-n=** argument is not the number of requested EOFs, but the minimum of variance that these EOFs must describe. The number of EOFs is then determined by reference with the minimum of explained variance required by the **-n=** argument. Express this explained variance in percentage (e.g., with an integer from 0 to 100).

If **-explvar** is specified, the **-n=** argument must be less or equal to 100. Note also that, if a randomized partial SVD algorithm is used (e.g., if **-alg=rsvd** is specified), the number of computed EOFs is limited to 200 even if these 200 EOFs describe less variance than the minimum requested with the **-n=** argument.

- 17) If the **-ortho** argument is used, the computed EOFs and associated PC time series are always reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods.

By default, they are only partially orthogonalized if the computed singular values are not well separated. Note that this argument has no effect if **-alg=svd** or **-alg=rsvd**.

- 18) If any of the **-nb=**, **-bl=**, **-bp=** and **-bm=** arguments is specified, a moving block bootstrap algorithm is used to test the stability of the computed EOFs and help to select the number of significant EOFs for further analysis.
- 19) The **-bm=bootstrap_method** argument specifies how bootstrap confidence intervals for explained variances are computed. If:

- **-bm=normal**, bootstrap confidence intervals of explained variances by the EOFs are based on asymptotic normality [Efron] [Politis] . These standard intervals will be bias-corrected [Politis] if the **-nobias** argument is also specified.
- **-bm=percentile**, bootstrap confidence intervals of explained variances by the EOFs are based on the percentile or bias-corrected percentile methods [Efron] . The bias-corrected percentile method will be used if the **-nobias** argument is also specified.

Note that the percentile method requires a larger number of shuffles (specified with the **-nb=** argument), especially if the *critical_probability* (specified with the **-pro=** argument) is small. More precisely, *number_of_shuffles.critical_probability.0.5* must be greater or equal to 1 . in the case of the percentile method. If it is not the case, the procedure will exit with an error message. As an illustration, with the default *critical_probability* (e.g., 0.05), *number_of_shuffles* must be at least 40 for using the percentile method for estimating confidence intervals for explained variances by the EOFs.

By default, the **-bm=** argument is set to `normal`.

- 20) The **-nb=number_of_shuffles** argument specifies the number of shuffles used in the bootstrap algorithm (by default 99).
- 21) The **-bp=bootstrap_periodicity** argument specifies that the index, *i*, of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where *j* is a random positive integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. This parameter is useful if the time series are cyclostationary instead of stationary.

By default, *bootstrap_periodicity* is set to 1 so that the time series is assumed to be stationary (see the description of the **-bt=** argument if you want to relax this strong assumption in the bootstrap algorithm).

- 22) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater or equal to the *bootstrap_periodicity* and less than the length of the time series. By default, *bootstrap_block_length* is set to *bootstrap_periodicity*.2. If **-a=cov** or **-a=cor** is specified, it is highly recommended to set *bootstrap_block_length* as a multiple of the periodicity in the data as this will take properly into account the cyclostationary of the analyzed time series.
- 23) The **-bt=bootstrap_trend_length** argument specifies that a non-stationary bootstrap algorithm must be used and sets the length of the trend smoother, *bootstrap_trend_length*, in this non-stationary bootstrap algorithm. The non-stationary bootstrap algorithm consists of two steps:
- First, trends are extracted from the time series by using a LOESS smoother [Cleveland] [Cleveland_Devlin]. This LOESS smoother is similar to the one performed in *comp_trend_4d* and the **-bt=** argument here has the the same meaning as the **-nt=** argument in *comp_trend_4d*.
 - Secondly, residuals from the estimated trends are computed and are used in a stationary moving block bootstrap algorithm. The final non-stationary bootstrapped time series are finally generated by adding the trend components to the bootstrapped residuals time series.

The value of *bootstrap_trend_length* should be an odd integer greater than or equal to 3. As *bootstrap_trend_length* increases the values of the trend component become smoother and the amplitude of the residuals increases, generating more variability in the bootstrapped time series.

- 24) The **-pro=critical_probability** argument specifies the critical probability which is used to determine the lower and upper bounds of the confidence intervals for explained variances by the EOFs if the bootstrap method is used. More precisely, Lower and upper $(1 - \text{critical_probability}) \cdot 100\%$ bootstrap confidence bounds for explained variances are computed based on asymptotic normality or the percentile method depending on the value of the **-bm=****bootstrap_method* argument. *critical_probability* must be greater than 0 . and less than 1 . . The default value is 0.05, meaning that 95% confidence intervals are computed by default.
- 25) If **-nobias** is specified, unbiased bootstrap confidence intervals for explained variances are computed. If the **-nobias** argument is absent, biased bootstrap confidence intervals are estimated by default.

If **-bm=percentile** is specified, this means that the confidence intervals will be obtained by the bias-corrected percentile method instead of the percentile method.

- 26) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified *missing_value* is set to `1.e+20`.

- 27) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 28) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 29) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 30) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 31) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 32) For more details on PCA and EOF analysis, the bootstrap approach, distances between subspaces, the LOESS smoother and randomized SVD algorithms, see

- “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
- “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
- “Matrix Computations”, by Golub, G.H., and Van Loan, C., The Johns Hopkins University Press, Baltimore, MD., 4rd Edition, 756 pp., 2013. ISBN: ‘978-1-4214-0794-4’_
- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsug.utexas.edu/fu/files/EOFSVD.pdf>
- “A primer for EOF analysis of climate data”, by Hannachi, A., Reading University, Reading, UK, 33pp, 2004. <http://www.met.reading.ac.uk/~han/Monitor/eofprimer.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300

- “Nonparametric standard-errors and confidence intervals” By Efron, B., The Canadian Journal of Statistics, 9, 2, 139-172, (1981). <https://doi.org/10.2307/3314608>
- “Computer-Intensive Methods in Statistical Analysis” By Politis, D.N., IEEE Signal Processing Magazine, 15, 1, 39-55, (January 1998). <https://ieeexplore.ieee.org/document/647042>
- “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: [10.1080/01621459.1979.10481038](https://doi.org/10.1080/01621459.1979.10481038)
- “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: [10.1080/01621459.1988.10478639](https://doi.org/10.1080/01621459.1988.10478639)
- “Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions”, By Halko, N., Martinsson, P.G., and Tropp, J.A., SIAM Rev., 53, 217-288, 2011. <https://epubs.siam.org/doi/abs/10.1137/090771806>
- “Algorithm 971: An implementation of a randomized algorithm for principal component analysis”, by Li, H., Linderman, G.C., Szlam, A., Stanton, K.P., Kluger, Y., and Tygert, M., ACM Trans. Math. Softw. 43, 3, Article 28 (January 2017). <https://pubmed.ncbi.nlm.nih.gov/28983138>
- “Randomized methods for matrix computations”, by Martinsson, P.G., arXiv.1607.01649, 2019. <https://arxiv.org/abs/1607.01649>
- “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>

Outputs

`comp_eof_4d` creates an output NetCDF file that contains the standardized PC time series, the scaled eigenvectors and the (square roots of) eigenvalues of the EOF analysis. The number of PC time series, eigenvectors and eigenvalues stored in the output NetCDF dataset is determined by the `-n=number_of_eofs` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_eof(number_of_eofs, nlev, nlat, nlon)` : the selected eigenvectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix between the observed variables. The eigenvectors are sorted by descending order of the associated eigenvalues. The eigenvectors are scaled such that they give the scalar products (`-a=scp`), covariances (`-a=cov`) or correlations (`-a=cor`) between the original observed variables and the associated PC time series.

The eigenvectors are packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values. If this is a problem, you can use `comp_norm_4d` for restricting the geographical domain in the input dataset before using `comp_eof_4d`.

- 2) `netcdf_variable_pc(ntime, number_of_eofs)` : the PC time series sorted by descending order of the eigenvalues (e.g., the squares of the singular values of the input matrix of observed variables).

The PC time series are always standardized to unit variance.

- 3) `netcdf_variable_sing(number_of_eofs)` : the singular values of the input data matrix in decreasing order, up to a constant scaling factor (equals to the square root of $1/ntime$) and taking

into account the effect of the metric used in the EOF analysis. More precisely, these singular values are the square roots of the eigenvalues of the sums of squares and cross-products (**-a=scp**) or covariance (**-a=cov**) or correlation (**-a=cor**) matrix between the observed variables.

These eigenvalues (e.g., the squares of the singular values) are equal to the (raw) variance described by the PC time series.

- 4) `netcdf_variable_var (number_of_eofs)` : the proportions of variance explained by each PC time series (given between 0. and 1. with 1. corresponding to 100%).

Furthermore, if one of the **-nb=**, **-bl=**, **-bp=** or **-bm=** arguments is specified when calling `comp_eof_4d`, the following NetCDF variables are computed and also stored in the output NetCDF file:

- 1) `netcdf_variable_var_boot_lower_bound (number_of_eofs)` : lower (1 - *critical_probability*)“100“% bootstrap confidence bounds for explained variances. By default, **critical_probability* is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.
- 2) `netcdf_variable_var_boot_upper_bound (number_of_eofs)` : upper (1 - *critical_probability*)“100“% bootstrap confidence bounds for explained variances. By default, **critical_probability* is set to 0.05, such that 95%\`% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the ****pro=**\`critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.
- 3) `netcdf_variable_var_boot_mean (number_of_eofs)` : the mean of bootstrapped explained variances.
- 4) `netcdf_variable_var_boot_var (number_of_eofs)` : the variance of bootstrapped explained variances.
- 5) `netcdf_variable_var_boot_std (number_of_eofs)` : the standard-deviation of bootstrapped explained variances.
- 6) `netcdf_variable_var_boot_skew (number_of_eofs)` : the skewness of bootstrapped explained variances.
- 7) `netcdf_variable_var_boot_kurt (number_of_eofs)` : the kurtosis of bootstrapped explained variances.
- 8) `netcdf_variable_var_boot_min (number_of_eofs)` : the minimum of bootstrapped explained variances.
- 9) `netcdf_variable_var_boot_max (number_of_eofs)` : the maximum of bootstrapped explained variances.
- 10) `netcdf_variable_cos_vec_boot_mean (number_of_eofs)` : the mean of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 11) `netcdf_variable_cos_vec_boot_var (number_of_eofs)` : the variance of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 12) `netcdf_variable_cos_vec_boot_std (number_of_eofs)` : the standard-deviation of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 13) `netcdf_variable_cos_vec_boot_skew (number_of_eofs)` : the skewness of the cosines of the angles between pairwise observed and bootstrapped EOFs.

- 14) *netcdf_variable_cos_vec_boot_kurt* (*number_of_eofs*) : the kurtosis of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 15) *netcdf_variable_cos_vec_boot_min* (*number_of_eofs*) : the minimum of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 16) *netcdf_variable_cos_vec_boot_max* (*number_of_eofs*) : the maximum of the cosines of the angles between pairwise observed and bootstrapped EOFs.
- 17) *netcdf_variable_spect_dist_subspace_boot_mean* (*number_of_eofs*) : the mean of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 18) *netcdf_variable_spect_dist_subspace_boot_var* (*number_of_eofs*) : the variance of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 19) *netcdf_variable_spect_dist_subspace_boot_std* (*number_of_eofs*) : the standard-deviation of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 20) *netcdf_variable_spect_dist_subspace_boot_skew* (*number_of_eofs*) : the skewness of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 21) *netcdf_variable_spect_dist_subspace_boot_kurt* (*number_of_eofs*) : the kurtosis of the spectral distances between the pairwise observed and bootstrapped EOF subspaces.
- 22) *netcdf_variable_spect_dist_subspace_boot_min* (*number_of_eofs*) : the minimum of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 23) *netcdf_variable_spect_dist_subspace_boot_max* (*number_of_eofs*) : the maximum of the spectral distances between pairwise observed and bootstrapped EOF subspaces.
- 24) *netcdf_variable_frob_dist_subspace_boot_mean* (*number_of_eofs*) : the mean of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 25) *netcdf_variable_frob_dist_subspace_boot_var* (*number_of_eofs*) : the variance of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 26) *netcdf_variable_frob_dist_subspace_boot_std* (*number_of_eofs*) : the standard-deviation of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 27) *netcdf_variable_frob_dist_subspace_boot_skew* (*number_of_eofs*) : the skewness of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 28) *netcdf_variable_frob_dist_subspace_boot_kurt* (*number_of_eofs*) : the kurtosis of the (scaled) Frobenius distances between the pairwise observed and bootstrapped EOF subspaces.
- 29) *netcdf_variable_frob_dist_subspace_boot_min* (*number_of_eofs*) : the minimum of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.
- 30) *netcdf_variable_frob_dist_subspace_boot_max* (*number_of_eofs*) : the maximum of the (scaled) Frobenius distances between pairwise observed and bootstrapped EOF subspaces.

Examples

- 1) For computing an EOF analysis from the NetCDF file `z.1970_2002.apm.nc`, which includes a NetCDF variable `z`, and store the results in a NetCDF file named `eof_era40_1m_z850_1979_2001.nc`, use the following command (the analysis is done on the centered data and only the level 21, which corresponds to 850 hPa, is considered in the analysis) :

```
$ comp_eof_4d \
  -f=z.1970_2002.apm.nc \
  -v=z \
```

(continues on next page)

(continued from previous page)

```
-z=21,21 \
-m=mask_era40_z.nc \
-g=n \
-c=clim_era40_lm_z_1979_2001.nc \
-d=dist2 \
-a=cov \
-n=10 \
-o=eof_era40_lm_z850_1979_2001.nc
```

- 2) For computing an EOF analysis from the NetCDF file `ST7_lm_0101_20012_grid_T_votemper.nc`, which includes a NetCDF variable `votemper` from a numerical simulation with the ORCA R2 configuration of the NEMO ocean model, and store the results in a NetCDF file named `eof_votemper.nc`, use the following command (the analysis is done on the standardized data and all the depths/levels are included in the analysis) :

```
$ comp_eof_4d \
-f=ST7_lm_0101_20012_grid_T_votemper.nc \
-v=votemper \
-c=clim_ST7_lm_0101_20012_grid_T_votemper.nc \
-a=cor \
-d=dist3 \
-m=meshmask.orca2.nc \
-g=t
```

2.17 comp_eof_miss_3d

2.17.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.17.2 Latest revision

30/04/2026

2.17.3 Purpose

Compute an Empirical Orthogonal Function (EOF) analysis , also known as Principal Component Analysis (PCA) from a tridimensional variable with missing values extracted from a NetCDF dataset.

The procedure first transforms the input tridimensional NetCDF variable as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 2-D grid-mesh associated with the tridimensional NetCDF variable) and then performs an EOF analysis of this rectangular matrix. Since missing values are present, there are irregularly distributed gaps in the matrix **X** and the procedure used here is different than the one used in [comp_eof_3d](#).

A first basic assumption of the estimation procedures used in `comp_eof_miss_3d` is that each row and column of the matrix, **X**, of the observed variables have at least one non-missing entry.

Two alternative strategies are available in `comp_eof_miss_3d` for estimating the first *k* eigenvectors and principal component time series of an EOF model with gappy data :

- 1) If the argument **-alg=** is set to `var`, `comp_eof_miss_3d` first computes an estimate of the matrix product (transpose(**X**).**X**)/*ntime* (as long as every pair of observed variables has at least one nonmissing entry). The elements of this matrix estimate are calculated from all valid observations for every pair of observed variables. If fewer

than one valid observation is present for some pair of variables, the procedure prints an error message and stops. In a second step, the first k eigenvectors, B (B is a k by nv matrix with eigenvectors stored rowwise) and eigenvalues of this matrix estimate are computed by inverse iteration. These eigenvectors are preliminary estimates of the first k eigenvectors of the matrix $(\text{transpose}(\mathbf{X})\mathbf{X})/ntime$. Note that because the matrix product estimate (derived from the data matrix \mathbf{X} with missing values) is not necessarily positive definite some of these eigenvalues may be negative. In a third step, `comp_eof_miss_3d` obtains preliminary least-squares estimates, A (A is a $ntime$ by k matrix), of the first k principal component scores by regressing the observations onto the eigenvectors, B , using only the valid entries in each observation. Note that these preliminary estimates of the first k principal component scores are not necessarily jointly uncorrelated if missing values are present.

- 2) If the argument `-alg=` is set to `obs`, `comp_eof_miss_3d` first computes an estimate of the matrix product $(\mathbf{X}\text{transpose}(\mathbf{X}))/nv$ (as long as every pair of observations has at least one nonmissing entry). The elements of this matrix estimate are calculated from all valid observed variables for every pair of observations. If fewer than one valid variable is present for some pair of observations, the procedure prints an error message and stops. In a second step, the first k eigenvectors, A (A is a $ntime$ by k matrix with eigenvectors stored columnwise) and eigenvalues of this matrix estimate are computed by inverse iteration. These eigenvectors are preliminary estimates of the first k (standardized) principal component scores. Note that because the matrix product estimate derived from the data matrix \mathbf{X} with missing values is not necessarily positive definite some of the eigenvalues may be negative. In a third step, `comp_eof_miss_3d` obtains preliminary least-squares estimates, B (B is a k by nv matrix), of the first k eigenvectors of the EOF model by regressing the variables onto the preliminary estimates of the principal component scores, B , using only the valid entries in each variable. Note that these preliminary estimates of the first k eigenvectors are not necessarily orthogonal if missing values are present.

After these alternative preliminary steps, it is an easy task to obtain suitable orthonormalization of the A and B matrices of the estimated k -component model similar to the traditional ones in the restricted k EOF model by computing the Singular Value Decomposition (SVD) of the AB matrix product. Note that there is no need to compute the AB product in this final computation since the SVD of this matrix product can be easily deduced from the two smallest SVD (or QR decomposition) of A and B , respectively. Let

$$AB = \mathbf{U}\mathbf{S}\mathbf{V}$$

where

- \mathbf{U} is a $ntime$ by k matrix with orthonormal columns (the left singular vectors stored columnwise)
- \mathbf{S} is a square k by k matrix with nonnegative elements on its principal diagonal and zeros elsewhere (the diagonal elements of \mathbf{S} are the singular values of AB)
- \mathbf{V} is a k by nv matrix with orthonormal rows (the right singular vectors stored rowwise)

Note that this SVD has no more than k terms with a singular value distinct from zero since AB is of rank less or equal to k . With these notations, the first k unstandardized “principal components” of the k -EOF model are just $\mathbf{A}=\mathbf{U}\mathbf{S}$ and the first k “eigenvectors” of this model may be defined as $\mathbf{B}=\mathbf{V}$ (up to a constant scaling factor).

With these final estimates of the EOF model with gappy data, observed that the k eigenvectors stored rowwise in \mathbf{B} are orthogonal and that the Principal Component (PC) time series stored columnwise in \mathbf{A} are jointly uncorrelated as for the traditional EOF model without missing values in the data as computed by `comp_eof_3d`.

More sophisticated methods are available in the literature for estimating PCA/EOF models with missing values [Terraya] [Terrayb] [Terrayd] [Schneider] and some of them will be implemented in future NCSTAT releases.

An output NetCDF dataset containing singular values (which are equal to the standard-deviations of the corresponding PC time series), estimates of percentages of variance explained by each PC time series, eigenvectors \mathbf{B} scaled by the corresponding singular values and standardized PC time series \mathbf{A} is created. The scaled eigenvectors are repacked as a tridimensional variable in the output NetCDF dataset.

Optionally, `comp_eof_miss_3d` may also compute estimates of the missing values in the data matrix, \mathbf{X} , of observed variables with the help of the estimated k -EOF model \mathbf{AB} and stored the resulting fill matrix in an output NetCDF file. This fill matrix is also repacked as a tridimensional variable in a second output NetCDF dataset which may be used for further analyses in NCSTAT procedures.

Refer to *comp_ortho_rot_eof_3d*, *comp_filt_rot_eof_3d* or *comp_loess_rot_eof_3d* if you want to perform an orthogonal rotation of all or selected of the computed standardized PC time series in order to simplify their physical interpretations [Jolliffe] [Wills_etal] .

This procedure is parallelized if OpenMP is used.

2.17.4 Further Details

Usage

```
$ comp_eof_miss_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-g=grid_type                (optional : n, t, u, v, w, f) \
-r=resolution                (optional : r2, r4) \
-b=nlon_orca, nlat_orca      (optional) \
-x=lon1,lon2                 (optional) \
-y=lat1,lat2                 (optional) \
-t=time1,time2               (optional) \
-a=type_of_analysis          (optional : scp, cov, cor) \
-c=input_climatology_netcdf_file (optional) \
-d=distance                  (optional : dist2, ident) \
-alg=algorithm                (optional : var, obs) \
-n=number_of_eofs            (optional) \
-o=output_eof_netcdf_file    (optional) \
-fo=output_netcdf_file       (optional) \
-mi=missing_value            (optional) \
-use_eps=tolerance           (optional) \
-comp_min_norm                (optional) \
-ortho                        (optional) \
-double                       (optional) \
-bigfile                      (optional) \
-hdf5                        (optional) \
-cdf5                         (optional) \
-tlimited                      (optional)
```

By default

- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO ocean model (e.g., if **-n=** argument is not set to *n*), the resolution is assumed to be *r2*
- b=** if **-n=** is not set to *n*, the dimensions of the grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *scp*. This means that the eigenvectors and eigenvalues are computed from the sums of squares and cross-products matrix between the observed variables
- c=** an *input_climatology_netcdf_file* is not needed if the *type_of_analysis* is set to *scp*, but must be specified if the *type_of_analysis* is set to *cov* or *cor*

- d=** the *distance* is set to `dist2`. This means that distances and scalar products in the EOF analysis are computed with the diagonal metric associated with the 2-D grid-mesh
- alg=** the *algorithm* option is set to `var`. This means that eigenvectors are first estimated in the space domain in the preliminary steps of the procedure
- n=** *number_of_eofs* is set to 1 and a 1-component EOF model is computed and stored in the output NetCDF file *output_eof_netcdf_file*
- o=** the *output_eof_netcdf_file* is named `eof_netcdf_variable.nc`
- fo=** an *output_netcdf_file* with a copy of the input *netcdf_variable* in which missing values are replaced by estimates computed from the selected EOF model is not created
- mi=** the *missing_value* attribute in the output NetCDF files is set to `1.e+20`
- use_eps=** the numerical rank is determined when solving each regression problem
- comp_min_norm** a minimal 2-norm solution is not computed when solving deficient linear least-squares problems
- ortho** the computed EOFs and associated PC time series are not automatically reorthogonalized
- double** the results of the EOF analysis are stored as single-precision floating point numbers in the output NetCDF files. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** NetCDF classical format file are created. If **-bigfile** is activated, the output NetCDF files are 64-bit offset format files
- hdf5** NetCDF classical format file are created. If **-hdf5** is activated, the output NetCDF files are NetCDF-4/HDF5 format files
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which an EOF analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to the *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix before computing the EOF analysis. The scale factors associated with the 2-D grid-mesh of this NetCDF variable (needed if **-d=dist2** is specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the geographical domain used in the EOF analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typedmask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*) which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF analysis.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_eof_miss_3d*.

- 4) If the **-t=***time1,time2* argument is missing the whole time period associated with the *netcdf_variable* is used to estimate eigenvalues, eigenvectors and PC time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF files will have `ntime = time2 - time1 + 1` observations.

- 5) It is assumed that the specified input *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the values of this *missing_value* or *_FillValue* attribute.
- 6) If the **-g=** argument is set to `t`, `u`, `v`, `w` or `f` it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

In this case, the duplicate points from the ORCA grid are removed before the EOF analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe. On output, the duplicate points are restored when writing the EOFs (e.g., the eigenvectors), if the geographical domain of the input NetCDF variable is the whole globe.

If the **-g=** argument is set to `n`, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typedmask*, *e1grid_type* and *e2grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_miss_3d* if the 2-D grid-mesh is regular or gaussian.

- 7) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (e.g., if the input NetCDF variable, *netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
 - **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.
- 8) If the input NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 9) The **-a=** argument specifies if the observed variables are centered or standardized with an input climatology (specified with the **-c=** argument) before the EOF analysis. If:
 - **-a=scp**, the EOF analysis is done on the raw data
 - **-a=cov**, the EOF analysis is done on the anomalies
 - **-a=cor**, the EOF analysis is done on the standardized anomalies.

By default, the **-a=** argument is set to `scp`.

- 10) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.
- 11) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 12) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 13) The **-n=***number_of_eofs* argument specifies the order of the EOF model which must be estimated from the gappy data and stored in the output NetCDF file specified by the **-o=** argument.

If the **-n=** argument is not specified, a 1-component EOF model is estimated.

14) The **-d=distance** argument specifies the metric and scalar product used in the EOF analysis. If:

- **-d=dist2**, the EOF analysis is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid point is weighted accordingly to the surface associated with it)
- **-d=ident**, the EOF analysis is done with the identity metric, e.g., the Euclidean distance and the usual scalar product are used in the EOF analysis.

By default, the **-d=** argument is set to `dist2`.

15) The **-alg=** argument specifies how the preliminary estimates of the eigenvectors and associated principal components are calculated. If:

- **-alg=var**, eigenvectors are estimated in the space domain first and time coefficients are obtained through a regression analysis in a second step
- **-alg=obs**, eigenvectors are estimated in the time domain first and space coefficients are obtained through a regression analysis in a second step.

In both cases, the final results are normalized with a partial SVD analysis as described above. Both algorithms are also parallelized if OpenMP is used.

The default algorithm is **-alg=var**.

16) **-use_eps=tolerance** is a real value used to determine the effective rank of the coefficient matrix for each regression problem solved in `comp_eof_miss_3d`. Tolerance must be set to the relative precision of the elements of the input data matrix. If each element is correct to, say, 5 digits then **-use_eps=0.00001** should be used.

Tolerance must not be greater or equal to 1 or less or equal than 0.

If the **-use_eps=** argument is absent, the numerical rank is determined for each regression problem solved in `comp_eof_miss_3d`.

17) The **-comp_min_norm** argument specifies that a minimal 2-norm solution must be computed for solving deficient linear least-squares problems in the regression steps of the procedure.

By default, a minimal 2-norm solution is not computed when solving deficient linear least-squares problems in the regression steps of the procedure.

18) The **-fo=output_netcdf_file** argument specifies that the missing values must be estimated from the computed EOF model (and climatology) and that the full data matrix repacked as a tridimensional variable must be stored in the NetCDF file `output_netcdf_file` on output of the procedure.

19) If the **-ortho** argument is used, the computed EOFs and associated PC time series are always reorthogonalized. By default, they are only partially orthogonalized if the computed eigenvalues are not well separated.

20) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the output netcdf files will be 64-bit offset format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the output netcdf files will be NetCDF-4/HDF5 format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the output netcdf files will be CDF5 format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 23) The **-mi=missing_value** argument specifies the missing value indicator (e.g., the *missing_value* attribute) associated with the NetCDF variables in the output NetCDF files.

If the **-mi=** argument is not specified the *missing_value* attribute is set to `1.e+20` in the output NetCDF files.

- 24) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF files.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 25) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 26) For more details on EOF analysis with missing values in the climate literature, see

- “Space/Time structure of monsoons interannual variability”, by Terray, P., Journal of Climate, Vol. 8, 2595-2619, 1995. doi: [10.1175/1520-0442\(1995\)008<2595:STSOMI>2.0.CO;2](https://doi.org/10.1175/1520-0442(1995)008<2595:STSOMI>2.0.CO;2)
- “Detecting Climatic Signals from Ship’s Datasets”, by Terray, P., Proceedings of the International Workshop on Digitization and Preparation of Historical Surface Marine Data and Metadata, 15-17 September. 1997, Toledo, Spain; 83-88 p.; H.F. Diaz and S.D. Woodruff, Eds., WMO/TD-No.957, MMROA Report No. 43, 1999. http://icoads.noaa.gov/mmroa43_toledo.pdf
- “Application of Weighted Empirical Orthogonal Function Analysis to ship’s datasets”, by Terray, P., Compte-Rendu de la IVème journée Statistique IPSL (Classification et Analyse spatiale). NAI n°23. pp. 11-28. 2002. ISSN 1626-8334. https://www.lmd.polytechnique.fr/nai/nai_23.pdf
- “Variable projection framework for the reduced-rank matrix approximation problem by weighted least-squares”, by Terray, P., arXiv:2505.03347, 220 pp., 2025. doi: [10.48550/arXiv.2505.03347](https://doi.org/10.48550/arXiv.2505.03347)
- “Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values”, by Schneider, T., Journal of Climate, Vol. 14, 853-871, 2001. doi: [10.1175/1520-0442\(2001\)014<0853:AOICDE>2.0.CO;2](https://doi.org/10.1175/1520-0442(2001)014<0853:AOICDE>2.0.CO;2)
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: [9780521012300](https://www.isbn-international.org/number/9780521012300)
- “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 2nd Ed, 2002. ISBN: [978-0-387-22440-4](https://www.springer.com/gp/book/9780387954424). <https://www.springer.com/gp/book/9780387954424>
- “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>

Outputs

`comp_eof_miss_3d` creates an output NetCDF file that contains the standardized PC time series, the scaled eigenvectors and the (square roots of) eigenvalues of the *number_of_eofs*-EOF model estimated from a dataset with missing values. The number of PC time series, eigenvectors and eigenvalues computed and stored in the output NetCDF dataset is determined by the **-n=number_of_eofs** argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_eof(number_of_eofs, nlat, nlon)` : the approximations of the selected eigenvectors of the sums of squares and cross-products (**-a=scp**), covariance (**-a=cov**) or correlation (**-a=cor**) matrix between the observed variables. The eigenvectors are sorted by descending order of the associated eigenvalues. The eigenvectors are scaled such that they give estimates of the scalar products (**-a=scp**), covariances (**-a=cov**) or correlations (**-a=cor**) between the original observed variables and the associated PC time series.

The eigenvectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values. If this is a problem, you can use `comp_norm_miss_3d` for restricting the geographical domain in the input dataset before using `comp_eof_miss_3d`.

- 2) `netcdf_variable_pc(ntime, number_of_eofs)` : the PC time series sorted by descending order of the eigenvalues.

The PC time series are always standardized to unit variance.

- 3) `netcdf_variable_sing(number_of_eofs)` : the singular values of the AB matrix product in decreasing order. Up to a constant scaling factor (equal to the square root of $1/ntime$), these singular values are estimates of the square roots of the eigenvalues of the sums of squares and cross-products (**-a=scp**) or covariance (**-a=cov**) or correlation (**-a=cor**) matrix between the observed variables.

These singular values are also estimates of the standard-deviations associated with the PC time series.

- 4) `netcdf_variable_var(number_of_eofs)` : estimates of the proportion of variance explained by each PC time series computed as the ratios between the estimates of variance described by the PC time series and the sum of the variances of the observed variables computed from non-missing data. These ratios are given, normally, between 0 . and 1 . with 1 . corresponding to 100%.

Optionally, `comp_eof_miss_3d` can also create an output NetCDF file that contains a copy of the input NetCDF variable with missing values filled with the help of the computed `number_of_eofs`-EOF model adjusted to the gappy data. You must use the **-fo=output_netcdf_file** argument in order to create this second output dataset. This optional output NetCDF dataset contains the following NetCDF variable (in the description below, `nlat` and `nlon` are the length of the dimensions of the input NetCDF variable):

- 1) `netcdf_variable(ntime, nlat, nlon)` : a copy of the input NetCDF variable with missing values filled with estimates computed with the help of the selected EOF model.

Examples

- 1) For estimating a 1-EOF model from the NetCDF file `hadcrut2v.nc`, which includes a NetCDF variable `temanom`, and store the results in a NetCDF file named `eof1_hadcrut2v.nc`, use the following command (the analysis is done on the raw data) :

```
$ comp_eof_miss_3d \
-f=hadcrut2v.nc \
-v=temanom \
-m=mesh_mask_hadcrut2v.nc \
-g=n \
-a=scp \
-n=1 \
-o=eof1_hadcrut2v.nc
```

- 2) For estimating a 3-EOF model from the NetCDF file `hadcrut2v.nc`, which includes a NetCDF variable `temanom`, and store the results in a NetCDF file named `eof3_hadcrut2v.nc` and, in addition, recon-

structuring a fill dataset with the help of this 3-EOF model approximation of your gappy data, use the following command :

```
$ comp_eof_miss_3d \  
-f=hadcrut2v.nc \  
-v=temanom \  
-m=mesh_mask_hadcrut2v.nc \  
-g=n \  
-a=scp \  
-n=3 \  
-o=eof3_hadcrut2v.nc \  
-fo=hadcrut2v_eof3.nc
```

2.18 comp_fftfiler_1d

2.18.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.18.2 Latest revision

30/04/2026

2.18.3 Purpose

Filter a real time series in a selected frequency band by windowed filtering [[Iacobucci_Noullez](#)] . The time series is extracted from a uni- or bidimensional variable readed from a NetCDF dataset and can also be detrended before windowed filtering at the user option.

The windowed filter is applied to the time series in the frequency domain [[Iacobucci_Noullez](#)] . The filter is obtained by convolving a raised-cosine window with the ideal rectangular filter response function. This filter is stationary and symmetric, therefore, it induces no phase-shift and is a good candidate for extracting frequency-defined series components from short-length time series.

Applying the filter in the frequency domain assumes implicitly that the time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series. The selected ideal filter gain vector (in the frequency domain) is convolved with the Fourier transform of a raised-cosine window to get the final filter frequency response. This filter frequency response is then convolved with the Fourier transform of the time series and the resulting sequence is inverted back in the time domain to obtain the final filtered time series. The specific form of the raised-cosine window can be controlled by the user with the help of the **-win=** argument, described below.

Additionally, the filtering can be done separately for different segments of equal length of the selected time series if this time series is not continuous in time (see the **-p=** argument).

This procedure returns the filtered real time series in a NetCDF dataset. If the NetCDF variable is trid or fourdimensional use *comp_fftfiler_3d* or *comp_fftfiler_4d*, respectively, instead of *comp_fftfiler_1d*. If the time series has a seasonal (or diurnal) cycle, use *comp_stl_1d* in order to estimate and remove the harmonic components of the time series before using *comp_fftfiler_1d*.

Finally, use *comp_spectvar_1d* to compute variance estimates in the selected frequency band for the real time series before or after filtering the real time series.

2.18.4 Further Details

Usage

```
$ comp_fftfiler_1d \
-f=input_netcdf_file \
-v=netcdf_variable \
-t=time1,time2 (optional) \
-o=output_netcdf_file (optional) \
-ni=index_for_2d_netcdf_variable (optional) \
-p=period_length (optional) \
-pl=minimum_period (optional) \
-ph=maximum_period (optional) \
-tr=trend_removal (optional : 0, 1, 2, 3, -1, -2, -3) \
-win=window_choice (optional : 0.5 > 1.) \
-mi=missing_value (optional) \
-double (optional) \
-hdf5 (optional) \
-cdf5 (optional) \
-tlimited (optional)
```

By default

- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- ni=** if the *netcdf_variable* is bidimensional, the first time series is used
- p=** the *period_length* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** *trend_removal* is set to 0, which means that no detrending is done before filtering
- win=0.54** the Hamming window is convolved with the ideal filter response
- mi=** the *missing_value* for the output variable is equal to $1.e+20$
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- limited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-limited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a windowed filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.

- 2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $ntime = time2 - time1 + 1$ time observations.

- 3) The **-ni=index_for_2d_netcdf_variable** argument specifies the index for selecting the time series if the *netcdf_variable* is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *index_for_2d_netcdf_variable* to 1.

- 4) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *period_length* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *period_length*.

- 5) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 1.

- 6) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 1 and less than the length of the time series or the *period_length* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 7) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the time series.
- 8) Setting **-pl=PL**, **-ph=PH** and $PH \ll PL$ is also allowed and performs band rejection of periods between *PH* and *PL*. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.
- 9) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.

- 10) The **-tr=** argument specifies pre- and post-filtering processing of the time series. If:

- **-tr=+/-1**, the mean of the time series is removed before time filtering
- **-tr=+/-2**, the drift from the time series is removed before time filtering. The drift for the time series is estimated using the formula: $drift = (tseries(ntime) - tseries(1)) / (ntime - 1)$
- **-tr=+/-3**, the least-squares line from the time series is removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the mean, drift or least-squares line are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 11) The **-win=** argument controls the form of the window, which will be convolved with the ideal filter response. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., the “ideal” filter).

The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1..

- 12) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 13) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 14) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 16) If the time series has a seasonal or diurnal cycle, use *comp_stl_1d* to remove the pure harmonic components from the time series before filtering.
- 17) It is assumed that the data has no missing values.
- 18) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 19) For more details on windowed filtering, see

- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_fftfiltfilter_1d` creates an output NetCDF file that contains the filtered time series estimated from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, *ntime* is the selected number of time observations) :

- 1) *netcdf_variable_filt(ntime)* : the filtered time series for the time series associated with the input NetCDF variable.

Examples

- 1) For windowed filtering a real monthly time series between 18 and 30 months (e.g., biennial time scale) from a NetCDF variable called `sst` extracted from the file `sst.monthly.nino34.nc`, which includes a monthly time series, and store the results in the NetCDF file `qbo_sst_nino34.nc`, use the following command :

```
$ comp_fftfiler_1d \
-f=sst.monthly.nino34.nc \
-v=sst \
-pl=18 \
-ph=30 \
-o=qbo_sst_nino34.nc
```

2.19 comp_fftfiler_3d

2.19.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.19.2 Latest revision

30/04/2026

2.19.3 Purpose

Filter a real multi-channel time series in a selected frequency band by windowed filtering [[Iacobucci_Noullez](#)] . The multi-channel time series is extracted from a tridimensional variable readed from a NetCDF dataset and can also be detrended before windowed filtering at the user option.

The windowed filter is applied to the time series in the frequency domain [[Iacobucci_Noullez](#)] . The filter is obtained by convolving a raised-cosine window with the ideal rectangular filter response function. This filter is stationary and symmetric, therefore, it induces no phase-shift and is a good candidate for extracting frequency-defined series components from short-length time series.

Applying the filter in the frequency domain assumes implicitly that the multi-channel time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series. The selected ideal filter gain vector (in the frequency domain) is multiplied with the Fourier transform of a raised-cosine window to get the final filter frequency response. This filter frequency response is then multiplied with the Fourier transforms of the multi-channel time series and the resulting sequences are inverted back in the time domain to obtain the final filtered multi-channel time series. The specific form of the raised-cosine window can be controlled by the user with the help of the **-win=** argument, described below.

Additionally, the filtering can be done separately for different segments of equal length of the selected multi-channel time series if the multi-channel time series are not continuous in time (see the **-p=** argument).

This procedure returns the filtered multi-channel time series in a NetCDF dataset. If the NetCDF variable is one or fourdimensional use *comp_fftfiler_1d* or *comp_fftfiler_4d*, respectively, instead of *comp_fftfiler_3d*. If the time series has a seasonal (or diurnal) cycle, use *comp_stl_3d* in order to estimate and remove the harmonic components of the time series before using *comp_fftfiler_3d*.

Finally, use *comp_spectvar_3d* to compute variance estimates in the selected frequency band for the real multi-channel time series before or after filtering the real multi-channel time series.

2.19.4 Further Details

Usage

```
$ comp_fftfiler_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file           (optional) \
-g=grid_type                             (optional : n, t, u, v, w, f) \
-x=lon1,lon2                             (optional) \
-y=lat1,lat2                             (optional) \
-t=time1,time2                           (optional) \
-o=output_netcdf_file                    (optional) \
-p=period_length                         (optional) \
-pl=minimum_period                       (optional) \
-ph=maximum_period                       (optional) \
-tr=trend_removal                        (optional : 0, 1, 2, 3, -1, -2, -3) \
-win=window_choice                       (optional : 0.5 > 1.) \
-mi=missing_value                        (optional) \
-ngp=number_of_grid_points                (optional) \
-double                                 (optional) \
-bigfile                                 (optional) \
-hdf5                                    (optional) \
-cdf5                                    (optional) \
-tlimited                                 (optional)
```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- p=** the *period_length* is set to *time2 - time1 + 1*, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** *trend_removal* is set to 0, which means that no detrending is done before filtering
- win=0.54** the Hamming window is convolved with the ideal filter response
- mi=** the *missing_value* for the output variable is equal to *1.e+20*
- ngp=** the *number_of_grid_points* is set to the number of points in the selected domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

-cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a windowed filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.

- 2) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF mesh_mask variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon + lon1 + 1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_fftfilt_3d*.

- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 6) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *period_length* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *period_length*.

- 7) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 1.

- 8) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 1 and less than the length of the time series or the *period_length* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the time series.
- 10) Setting **-pl=PL**, **-ph=PH** and $PH \ll PL$ is also allowed and performs band rejection of periods between *PH* and *PL*. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.
- 11) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequency filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.
- 12) The **-tr=** argument specifies pre- and post-filtering processing of the multi-channel time series. If:

- **-tr=+/-1**, the mean of the time series is removed before time filtering
- **-tr=+/-2**, the drift from the time series is removed before time filtering. The drift for the time series is estimated using the formula: $drift = (tseries(ntime) - tseries(1)) / (ntime - 1)$
- **-tr=+/-3**, the least-squares line from the time series is removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the mean, drift or least-squares line are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 13) The **-win=** argument controls the form of the window, which will be convolved with the ideal filter response. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., the “ideal” filter).

The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1..

- 14) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to 1.e+20.

- 15) The **-ngp=** argument can be used if you have memory problems when running `comp_fftfiler_3d` on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.

- 16) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 17) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The `-cdf5` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 20) If the multi-channel time series has a seasonal or diurnal cycle, use `comp_stl_3d` or `comp_clim_4d` to remove the pure harmonic components from the time series before filtering.
- 21) It is assumed that the data has no missing values.
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 23) For more details on windowed filtering, see
- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_fftfilter_3d` creates an output NetCDF file that contains the filtered time series estimated from the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable and `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_filt(ntime, nlat, nlon)` : the filtered time series for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

The filtered multi-channel time series is packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For windowed filtering a real multi-channel monthly time series between 18 and 30 months (e.g., bi-ennial time scale) from a tridimensional NetCDF variable called `mslp` extracted from the file `mslp.monthly.mean_ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `tbo_mslp_ncep2.nc`, use the following command :

```
$ comp_fftfilter_3d \  
-f=mslp.monthly.mean_ncep2.nc \  
-v=mslp \  
-m=mesh_mask_mslp_ncep2.nc \  
-pl=18 \  
-ph=30 \  
-o=tbo_mslp_ncep2.nc
```


2.20 comp_fftfILTER_4d

2.20.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.20.2 Latest revision

30/04/2026

2.20.3 Purpose

Filter a real multi-channel time series in a selected frequency band by windowed filtering [Iacobucci_Noullez] . The multi-channel time series is extracted from a fourimensional variable readed from a NetCDF dataset and can also be detrended before windowed filtering at the user option.

The windowed filter is applied to the time series in the frequency domain [Iacobucci_Noullez] . The filter is obtained by convolving a raised-cosine window with the ideal rectangular filter response function. This filter is stationary and symmetric, therefore, it induces no phase-shift and is a good candidate for extracting frequency-defined series components from short-length time series.

Applying the filter in the frequency domain assumes implicitly that the multi-channel time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series. The selected ideal filter gain vector (in the frequency domain) is multiplied with the Fourier transform of a raised-cosine window to get the final filter frequency response. This filter frequency response is then multiplied with the Fourier transforms of the multi-channel time series and the resulting sequences are inverted back in the time domain to obtain the final filtered multi-channel time series. The specific form of the raised-cosine window can be controlled by the user with the help of the **-win=** argument, described below.

Additionally, the filtering can be done separately for different segments of equal length of the selected multi-channel time series if the multi-channel time series are not continuous in time (see the **-p=** argument).

This procedure returns the filtered multi-channel time series in a NetCDF dataset. If the NetCDF variable is one or tridimensional use *comp_fftfILTER_1d* or *comp_fftfILTER_3d*, respectively, instead of *comp_fftfILTER_4d*. If the time series has a seasonal (or diurnal) cycle, use *comp_stl_4d* in order to estimate and remove the harmonic components of the time series before using *comp_fftfILTER_4d*.

Finally, use *comp_spectvar_4d* to compute variance estimates in the selected frequency band for the real multi-channel time series before or after filtering the real multi-channel time series.

2.20.4 Further Details

Usage

```
$ comp_fftfILTER_4d \
  -f=input_netCDF_file \
  -v=netCDF_variable \
  -m=input_mesh_mask_netCDF_file           (optional) \
  -g=grid_type                             (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                             (optional) \
  -y=lat1,lat2                             (optional) \
  -z=level1,level2                         (optional) \
```

(continues on next page)

(continued from previous page)

```

-t=time1,time2                (optional) \
-o=output_netcdf_file         (optional) \
-p=period_length              (optional) \
-pl=minimum_period            (optional) \
-ph=maximum_period            (optional) \
-tr=trend_removal             (optional : 0, 1, 2, 3, -1, -2, -3) \
-win=window_choice            (optional : 0.5 > 1.) \
-mi=missing_value             (optional) \
-ngp=number_of_grid_points    (optional) \
-double                       (optional) \
-bigfile                      (optional) \
-hdf5                         (optional) \
-cdf5                         (optional) \
-tlimited                      (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to `n` which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named `filt_netcdf_variable.nc`
- p=** the *period_length* is set to `time2 - time1 + 1`, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** *trend_removal* is set to 0, which means that no detrending is done before filtering
- win=0.54** the Hamming window is convolved with the ideal filter response
- mi=** the *missing_value* for the output variable is equal to `1.e+20`
- ngp=** the *number_of_grid_points* is set to the number of points in the selected domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a windowed filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF *mesh_mask* variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_fftfiler_4d*.

- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 6) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *period_length* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *period_length*.

- 7) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 1.

- 8) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 1 and less than the length of the time series or the *period_length* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the time series.
- 10) Setting **-pl=PL**, **-ph=PH** and $PH \ll PL$ is also allowed and performs band rejection of periods between *PH* and *PL*. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.
- 11) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.

- 12) The **-tr=** argument specifies pre- and post-filtering processing of the multi-channel time series. If:
- **-tr=+/-1**, the mean of the time series is removed before time filtering
 - **-tr=+/-2**, the drift from the time series is removed before time filtering. The drift for the time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
 - **-tr=+/-3**, the least-squares line from the time series is removed before time filtering.
- If **-tr=-1**, **-2** or **-3**, the mean, drift or least-squares line are reintroduced post-filtering, respectively.
- For other values of the **-tr=** argument, nothing is done before or after filtering.
- If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.
- The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.
- 13) The **-win=** argument controls the form of the window, which will be convolved with the ideal filter response. By default, a Hamming window is used (e.g., **-win=0.54**).
- Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., the “ideal” filter).
- The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1..
- 14) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.
- If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.
- 15) The **-ngp=** argument can be used if you have memory problems when running `comp_fftfiler_3d` on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.
- 16) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.
- By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 17) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.
- If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 18) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.
- If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 19) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.
- If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 20) If the multi-channel time series has a seasonal or diurnal cycle, use `comp_stl_4d` or `comp_clim_4d` to remove the pure harmonic components from the time series before filtering.
- 21) It is assumed that the data has no missing values.
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 23) For more details on windowed filtering, see
 - “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_fftfiler_4d` creates an output NetCDF file that contains the filtered time series estimated from the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the vertical and spatial dimensions of the input NetCDF variable and `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_filt(ntime, nlev, nlat, nlon)` : the filtered time series for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

The filtered multi-channel time series is packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For windowed filtering a real multi-channel monthly time series between 18 and 30 months (e.g., biennial time scale) from a fourdimensional NetCDF variable called `uwnd` extracted from the file `uwnd.monthly.mean.ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `qbo_uwnd_ncep2.nc`, use the following command :

```
$ comp_fftfiler_4d \
-f=uwnd.monthly.mean.ncep2.nc \
-v=uwnd \
-m=mesh_mask_uwnd_ncep2.nc \
-pl=18 \
-ph=30 \
-o=qbo_uwnd_ncep2.nc
```

2.21 comp_filt_rot_eof_3d

2.21.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.21.2 Latest revision

30/04/2026

2.21.3 Purpose

Performs an orthogonal rotation of selected standardized Principal Components (PC) time series extracted from a previous Empirical Orthogonal Function (EOF) analysis or Principal Component Analysis (PCA) of a tridimensional NetCDF variable towards low-frequency, high-frequency or band-pass components at the user option [[Wills_etal](#)] . The associated factor loading matrix (e.g., the associated scaled EOF patterns corresponding to the selected rotated PC time series) are also rotated and the new rotated EOF or PCA model is stored in a NetCDF file.

Spatial EOF patterns (e.g., scaled EOFs) and PC time series derived from EOF analysis or PCA are constrained to be both spatially and temporally orthogonal to one another so that they maximize the variance they described over the entire analysis domain and selected time period [[Jolliffe](#)] [[vonStorch_Zwiers](#)] . However, the spatial orthogonality of the EOFs is a strong and undesirable constraint imposed on the EOFs and, as a result, EOFs are subject to problems such as domain dependence and inaccurate representation of physical relationships embedded in the data [[Jolliffe](#)] [[Richman](#)] .

Orthogonal rotation of selected standardized PC time series can be used to obtain spatial or temporal patterns (e.g., rotated scaled EOFs and PC time series) that are hopefully more physically meaningful by relaxing the spatial orthogonality of the rotated scaled EOFs, while preserving the temporal orthogonality of the rotated standardized PC time series. Preserving the orthogonality of the (rotated) PC time series implies that these rotated PCs describe the same amount of total variance as the original (unrotated) PC time series, which is a very useful property, but this variance is more equally (or differently) distributed among the rotated PC time series [[Jolliffe](#)] [[Richman](#)] .

In classical orthogonal rotation PCA methods (e.g., like varimax rotation see [comp_ortho_rot_eof_3d](#) for details), the objective is to make the rotated standardized PC time series as simple as possible to interpret by attempting to constrain the spatial coefficients (e.g., the scaled EOFs) associated with each of them to be large or very small in absolute magnitude, but avoiding as much as possible spatial loadings of moderate size [[Jolliffe](#)] [[Jackson](#)] . On the other hand, here in [comp_filt_rot_eof_3d](#), the focus of the orthogonal rotation is to obtain a cleaner representation of the different time scales embedded in the selected PC time series such that these time scales will not be mixed on the same PC time series after the orthogonal rotation [[Wills_etal](#)] .

It can be shown that a rotation matrix, which will maximize such clean representation of the time scales of interest, can be estimated as the eigenvectors of the covariance matrix between the selected standardized PC time series filtered in the frequency band of interest. This type of method is called Low-Frequency Components Analysis (LFCA) [[Wills_etal](#)] .

Here, LFCA is considered as an orthogonal rotation method of the original standardized PC time series of a (partial) EOF or PCA model. The `nrot` by `nrot` orthogonal matrix (where `nrot` is the number of rotated PCs) used to rotate the original PCs is computed as the eigenvectors of the covariance (e.g., symmetric positive-definite) matrix between the filtered selected standardized PC time series [[Wills_etal](#)] .

In [comp_filt_rot_eof_3d](#), the filtering of the original PC time series is performed with the help of a windowed FFT filter specified by the values of the `-pl=`, `-ph=`, `-tr=` and `-win=` arguments. See [[Iacobucci_Noullez](#)] for more details on the windowed FFT filter used here. This windowed filter is applied to the selected PC time series in the frequency domain [[Iacobucci_Noullez](#)] . The filter is obtained by convolving a raised-cosine window with the selected ideal rectangular filter response function (which is specified with the help of the `-pl=` and `-ph=` arguments). The specific form of the raised-cosine window can be controlled by the user with the help of the `-win=` argument described below. This filter is stationary and symmetric, therefore, it induces no phase-shift and is a good candidate for extracting frequency-defined series components from short-length time series. This windowed filter is also implemented in [comp_fftfilter_3d](#), where more details on its properties are provided.

Using as input an EOF NetCDF file produced by [comp_eof_3d](#) or [comp_eof_miss_3d](#), [comp_filt_rot_eof_3d](#) extracts selected PC time series from this input file (see the description of `-se=` argument below for details) and performs the

orthogonal rotation of the selected standardized PC time series towards modes that have more energy in the frequency bands of interest selected with the help of the **-pl=** and **-ph=** arguments. `comp_filt_rot_eof_3d` also reads the associated scaled EOFs from the input EOF NetCDF file and computes the regression spatial patterns associated with the rotated standardized PC time series (e.g., the orthogonal projection onto the orthonormal set formed by the new rotated standardized PC time series). The percentages of variance described by the new rotated standardized PC time series are also computed and stored in the output NetCDF file, as are the ratios of filtered variance to total variance for the rotated standardized PC time series.

The selected standardized PC time series, which will be rotated, are determined on entry by the value of the **-se=** argument, which is a list of PC (or EOF) numbers.

An output NetCDF dataset containing these rotated PC standardized time series, their associated regression spatial patterns and the explained variance and ratio statistics is created.

If the NetCDF variable used in the original EOF analysis is fourdimensional use `comp_filt_rot_eof_4d` instead of `comp_filt_rot_eof_3d`.

2.21.4 Further Details

Usage

```
$ comp_filt_rot_eof_3d\
-f=eof_netcdf_file \
-v=eof_netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-pl=minimum_period \
-ph=maximum_period \
-se=selected_eofs           (optional) \
-g=grid_type                (optional : n, t, u, v, w, f) \
-r=resolution               (optional : r2, r4) \
-b=nlon_orca, nlat_orca     (optional) \
-x=lon1,lon2                (optional) \
-y=lat1,lat2                (optional) \
-t=time1,time2              (optional) \
-tr=trend_removal           (optional : 0, 1, 2, 3, -1, -2, -3) \
-win=window_choice          (optional : 0.5 > 1.) \
-o=output_rot_eof_netcdf_file (optional) \
-mi=missing_value           (optional) \
-double                     (optional) \
-bigfile                    (optional) \
-hdf5                       (optional) \
-cdf5                       (optional) \
-tlimited                    (optional)
```

By default

- se=** all the PC time series stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF EOF variable, *eof_netcdf_variable*, is assumed to be regular or Gaussian
- r=** if the input *eof_netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument

- x=** the whole longitude domain associated with the *eof_netcdf_variable*
- y=** the whole latitude domain associated with the *eof_netcdf_variable*
- t=** the whole time period associated with the input PC time series
- tr=0** *trend_removal* is set to 0, which means that no detrending is done before filtering the PC time series
- win=0.54** the Hamming window is convolved with the ideal filter response
- o=** the *output_rot_eof_netcdf_file* is named *filt_rot_eof_netcdf_variable.nc*
- mi=** the *missing_value* attribute for the rotated EOFs in the output NetCDF file is set to $1.e+20$
- double** the results of the rotated EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=***eof_netcdf_variable* argument specifies the NetCDF variable for which an EOF analysis was originally computed by *comp_eof_3d* (or *comp_eof_miss_3d*) and the **-f=***input_eof_netcdf_file* argument specifies that the resulting EOF patterns and PC time series must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_3d* or *comp_eof_miss_3d*.

These EOF patterns and PC time series are used to compute the orthogonal rotation matrix which will be applied to the selected EOF model.

- 2) The **-m=***input_mesh_mask_netcdf_file* argument specifies the land-sea mask to apply to the *eof_netcdf_variable* (which contains the scaled eigenvectors of the EOF analysis) for transforming this tridimensional NetCDF variable as a rectangular matrix before performing the rotated EOF analysis. It is assumed that this land-sea mask is exactly the same as the one used in the original EOF analysis, whose results are stored in the input NetCDF file *eof_netcdf_file*.
- 3) The **-se=***selected_eofs* argument allows the user to select the standardized PC time series which must be included in the orthogonal rotation. The list of selected PCs may be given in two formats:
 - **-se=***1,3,...,nn* allows to include *pc1,pc3,...* and *pcnn* in the orthogonal rotation
 - **-se=***1 : 4* allows to include from *pc1* to *pc4* in the orthogonal rotation.

The two forms of the **-se=** argument may be combined and repeated any number of times, but duplicate PC numbers are not allowed. If the **-se=** argument is not specified, all the PC time series stored in the *input_eof_netcdf_file* are used in the orthogonal rotation.

- 4) If **-g=** is set to *t, u, v, w* or *f* it is assumed that the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

In this case, the duplicate points from the ORCA grid are removed before the rotated EOF analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask in the *input_mesh_mask_netcdf_file* (e.g., the variable named *grid_typemask*). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_3d* if the 2-D grid-mesh is regular or gaussian.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
 - **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.
- 6) If the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 7) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are not specified, the geographical domain used in the rotated EOF analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_filt_rot_eof_3d*.

- 8) If the **-t=time1,time2** argument is missing, the whole time period associated with the original PC time series is used to compute the orthogonal rotation matrix.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have *ntime = time2 - time1 + 1* observations.
- 9) The geographical shapes of the *eof_netcdf_variable* (in the *eof_netcdf_file*) and the land-sea mask (in the *input_mesh_mask_netcdf_file*) must agree.
- 10) The **-pl=** argument specifies the minimum period of oscillation of the filtered PC time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 11) The **-ph=** argument specifies the maximum period of oscillation of the filtered PC time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument.

- 12) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the PC time series.

- 13) Setting **-pl=PL**, **-ph=PH** and $PH < PL$ is also allowed and performs band rejection of periods between PH and PL. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.
- 14) Setting both **-pl=0** and **-ph=0** is not allowed.
- 15) The **-tr=trend_removal** argument specifies pre- and post-filtering processing of the multi-channel PC time series. If:

- **-tr=+/-1**, the mean of the PC time series is removed before time filtering
- **-tr=+/-2**, the drift from the PC time series is removed before time filtering. The drift for the PC time series is estimated using the formula: $drift = (tseries(ntime) - tseries(1)) / (ntime - 1)$
- **-tr=+/-3**, the least-squares line from the PC time series is removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the mean, drift or least-squares line are reintroduced post-filtering, respectively. The covariance matrix between the (detrended) and filtered PC time series is then computed and the eigenvectors of this symmetric positive-definite matrix, ranked in decreasing order of the associated positive eigenvalues are calculated in order to obtain the orthogonal rotation matrix to apply to the selected EOFs and PC time series.

For other values of the **-tr=** argument, nothing is done before or after filtering the PC time series.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 16) The **-win=** argument controls the form of the window, which will be convolved with the ideal filter response. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., the “ideal” filter).

The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1.

- 17) The **-mi=missing_value** argument specifies the missing value indicator associated with the rotated EOF NetCDF variable in the *output_rot_eof_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.
- 18) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file, *output_rot_eof_netcdf_file*.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 19) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 21) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 22) It is assumed that the original EOF variable has no missing values excepted those associated with a constant land-sea mask.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on EOF analysis, PCA and orthogonal rotation of PC time series to selected-frequency components and windowed filtering of time series, see
 - “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
 - “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 487 pp., 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
 - “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
 - “Rotation of principal components” by Richman, M.B., International Journal of Climatology, Vol. 6, 293-335, 1986. <https://doi.org/10.1002/joc.3370060305>
 - “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>
 - “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_filt_rot_eof_3d` creates an output NetCDF file that contains the rotated standardized PC time series, their associated rotated EOF patterns and the variances described by the computed rotated PC time series. The ratios of filtered variance to total variance for the rotated standardized PC time series are also stored in the output NetCDF file.

The number of rotated PCs and EOFs, `nrot`, stored in the output NetCDF dataset is determined by the `-se=selected_eofs` argument. The number of observations in the output NetCDF dataset is determined from the `-t=time1,time2` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable `eof_netcdf_variable` and `nrot` is the number of selected PC time series, which have been rotated, as determined by the list specified with the `-se=selected_eofs` argument) :

- 1) `pc_number(nrot)` : the list of rotated PC time series as specified with the `-se=selected_eofs` argument.
- 2) `eof_netcdf_variable_filt_rot_eof(nrot,nlat,nlon)` : the rotated EOF patterns. These rotated EOFs are scaled such that they give the scalar products (if `-a=scp` in the original EOF analysis), covariances (`-a=cov` in the original EOF analysis) or correlations (`-a=cor` in the original EOF analysis) between the original observed variables and the rotated PC time series.

The rotated EOF patterns are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `eof_netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, this output NetCDF variable is filled with missing values.

- 3) `eof_netcdf_variable_filt_rot_pc`(`ntime`,`nrot`) : the rotated standardized PC time series.

The rotated PC time series are always standardized to unit variance.

- 4) `eof_netcdf_variable_filt_rot_std`(`nrot`) : the standard-deviations of the rotated PC time series.

The squares of these statistics are equal to the raw variance described by the rotated PC time series.

- 5) `eof_netcdf_variable_filt_rot_var`(`nrot`) : the proportion of variance explained by each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

- 6) `eof_netcdf_variable_filt_rot_ratio`(`nrot`) : the ratio of filtered variance to total variance for each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

Examples

- 1) For computing a rotated 10-EOF analysis towards biennial modes (e.g., time scales between 18 and 30 months) from a previous EOF analysis (stored in the file `eof_HadISST1_1m_197902_200501_sst_oi.nc`) of a tridimensional NetCDF variable named ```sst`, which contains monthly SST data; and stored the results of the rotated EOF analysis in a NetCDF file named `filt_rot_eof_HadISST1_1m_197902_200501_sst_oi.nc`, use the following command :

```
$ comp_filt_rot_eof_3d \  
-f=eof_HadISST1_1m_197902_200501_sst_oi.nc \  
-v=sst \  
-se=1:10 \  
-pl=18 \  
-ph=30 \  
-m=HadISST1_mask.nc \  
-o=filt_rot_eof_HadISST1_1m_197902_200501_sst_oi.nc
```

2.22 comp_filt_rot_eof_4d

2.22.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.22.2 Latest revision

30/04/2026

2.22.3 Purpose

Performs an orthogonal rotation of selected standardized Principal Components (PC) time series extracted from a previous Empirical Orthogonal Function (EOF) analysis or Principal Component Analysis (PCA) of a fourdimensional NetCDF variable towards low-frequency, high-frequency or band-pass components at the user option [[Wills_etal](#)] . The associated factor loading matrix (e.g., the associated scaled EOF patterns corresponding to the selected rotated PC time series) are also rotated and the new rotated EOF or PCA model is stored in a NetCDF file.

Spatial EOF patterns (e.g., scaled EOFs) and PC time series derived from EOF analysis or PCA are constrained to be both spatially and temporally orthogonal to one another so that they maximize the variance they described over the entire analysis domain and selected time period [[Jolliffe](#)] [[vonStorch_Zwiers](#)] . However, the spatial orthogonality of the EOFs is a strong and undesirable constraint imposed on the EOFs and, as a result, EOFs are subject to problems

such as domain dependence and inaccurate representation of physical relationships embedded in the data [Jolliffe] [Richman] .

Orthogonal rotation of selected standardized PC time series can be used to obtain spatial or temporal patterns (e.g., rotated scaled EOFs and PC time series) that are hopefully more physically meaningful by relaxing the spatial orthogonality of the rotated scaled EOFs, while preserving the temporal orthogonality of the rotated standardized PC time series. Preserving the orthogonality of the (rotated) PC time series implies that these rotated PCs describe the same amount of total variance as the original (unrotated) PC time series, which is a very useful property, but this variance is more equally (or differently) distributed among the rotated PC time series [Jolliffe] [Richman] .

In classical orthogonal rotation PCA methods (e.g., like varimax rotation see [comp_ortho_rot_eof_4d](#) for details), the objective is to make the rotated standardized PC time series as simple as possible to interpret by attempting to constrain the spatial coefficients (e.g., the scaled EOFs) associated with each of them to be large or very small in absolute magnitude, but avoiding as much as possible spatial loadings of moderate size [Jolliffe] [Jackson] . On the other hand, here in [comp_filt_rot_eof_4d](#), the focus of the orthogonal rotation is to obtain a cleaner representation of the different time scales embedded in the selected PC time series such that these time scales will not be mixed on the same PC time series after the orthogonal rotation [Wills_etal] .

It can be shown that a rotation matrix, which will maximize such clean representation of the time scales of interest, can be estimated as the eigenvectors of the covariance matrix between the selected standardized PC time series filtered in the frequency band of interest. This type of method is called Low-Frequency Components Analysis (LFCA) [Wills_etal] .

Here, LFCA is considered as an orthogonal rotation method of the original standardized PC time series of a (partial) EOF or PCA model. The `nrot` by `nrot` orthogonal matrix (where `nrot` is the number of rotated PCs) used to rotate the original PCs is computed as the eigenvectors of the covariance (e.g., symmetric positive-definite) matrix between the filtered selected standardized PC time series [Wills_etal] .

In [comp_filt_rot_eof_4d](#), the filtering of the original PC time series is performed with the help of a windowed FFT filter specified by the values of the `-pl=`, `-ph=`, `-tr=` and `-win=` arguments. See [Iacobucci_Noullez] for more details on the windowed FFT filter used here. This windowed filter is applied to the selected PC time series in the frequency domain [Iacobucci_Noullez] . The filter is obtained by convolving a raised-cosine window with the selected ideal rectangular filter response function (which is specified with the help of the `-pl=` and `-ph=` arguments). The specific form of the raised-cosine window can be controlled by the user with the help of the `-win=` argument described below. This filter is stationary and symmetric, therefore, it induces no phase-shift and is a good candidate for extracting frequency-defined series components from short-length time series. This windowed filter is also implemented in [comp_fftfiler_4d](#), where more details on its properties are provided.

Using as input an EOF NetCDF file produced by [comp_eof_4d](#), [comp_filt_rot_eof_4d](#) extracts selected PC time series from this input file (see the description of `-se=` argument below for details) and performs the orthogonal rotation of the selected standardized PC time series towards modes that have more energy in the frequency bands of interest selected with the help of the `-pl=` and `-ph=` arguments. [comp_filt_rot_eof_4d](#) also reads the associated scaled EOFs from the input EOF NetCDF file and computes the regression spatial patterns associated with the rotated standardized PC time series (e.g., the orthogonal projection onto the orthonormal set formed by the new rotated standardized PC time series). The percentages of variance described by the new rotated standardized PC time series are also computed and stored in the output NetCDF file, as are the ratios of filtered variance to total variance for the rotated standardized PC time series.

The selected standardized PC time series, which will be rotated, are determined on entry by the value of the `-se=` argument, which is a list of PC (or EOF) numbers.

An output NetCDF dataset containing these rotated PC standardized time series, their associated regression spatial patterns and the explained variance and ratio statistics is created.

If the NetCDF variable used in the original EOF analysis is tridimensional use [comp_filt_rot_eof_3d](#) instead of [comp_filt_rot_eof_4d](#).

2.22.4 Further Details

Usage

```
$ comp_filt_rot_eof_4d\
-f=eof_netcdf_file \
-v=eof_netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-pl=minimum_period \
-ph=maximum_period \
-se=selected_eofs (optional) \
-g=grid_type (optional : n, t, u, v, w, f) \
-r=resolution (optional : r2, r4) \
-b=nlon_orca, nlat_orca, nlevel_orca (optional) \
-x=lon1,lon2 (optional) \
-y=lat1,lat2 (optional) \
-z=level1,level2 (optional) \
-t=time1,time2 (optional) \
-tr=trend_removal (optional : 0, 1, 2, 3, -1, -2, -3) \
-win=window_choice (optional : 0.5 > 1.) \
-o=output_rot_eof_netcdf_file (optional) \
-mi=missing_value (optional) \
-double (optional) \
-bigfile (optional) \
-hdf5 (optional) \
-cdf5 (optional) \
-tlimited (optional)
```

By default

- se=** all the PC time series stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF EOF variable, *eof_netcdf_variable*, is assumed to be regular or Gaussian
- r=** if the input *eof_netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-n=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *eof_netcdf_variable*
- y=** the whole latitude domain associated with the *eof_netcdf_variable*
- z=** the whole vertical resolution associated with the *eof_netcdf_variable*
- t=** the whole time period associated with the input PC time series
- tr=0** *trend_removal* is set to 0, which means that no detrending is done before filtering the PC time series
- win=0.54** the Hamming window is convolved with the ideal filter response
- o=** the *output_rot_eof_netcdf_file* is named *filt_rot_eof_netcdf_variable.nc*
- mi=** the *missing_value* attribute for the rotated EOFs in the output NetCDF file is set to *1.e+20*

- double** the results of the rotated EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=eof_netcdf_variable** argument specifies the NetCDF variable for which an EOF analysis was originally computed by *comp_eof_4d*) and the **-f=input_eof_netcdf_file** argument specifies that the resulting EOF patterns and PC time series must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_4d*.

These EOF patterns and PC time series are used to compute the orthogonal rotation matrix which will be applied to the selected EOF model.

- 2) The **-m=input_mesh_mask_netcdf_file** argument specifies the land-sea mask to apply to the *eof_netcdf_variable* (which contains the scaled eigenvectors of the EOF analysis) for transforming this fourdimensional NetCDF variable as a rectangular matrix before performing the rotated EOF analysis. It is assumed that this land-sea mask is exactly the same as the one used in the original EOF analysis, whose results are stored in the input NetCDF file *eof_netcdf_file*.
- 3) The **-se=selected_eofs** argument allows the user to select the standardized PC time series which must be included in the orthogonal rotation. The list of selected PCs may be given in two formats:

- **-se=1,3,...,nn** allows to include *pc1,pc3,...* and *pcnn* in the orthogonal rotation
- **-se=1 : 4** allows to include from *pc1* to *pc4* in the orthogonal rotation.

The two forms of the **-se=** argument may be combined and repeated any number of times, but duplicate PC numbers are not allowed. If the **-se=** argument is not specified, all the PC time series stored in the *input_eof_netcdf_file* are used in the orthogonal rotation.

- 4) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

In this case, the duplicate points from the ORCA grid are removed before the rotated EOF analysis, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 3-D mesh-mask in the *input_mesh_mask_netcdf_file* (e.g., the variable named *grid_typedmask*). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_4d* if the 3-D grid-mesh is regular or gaussian.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model

- **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.
- 6) If the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
 - 7) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are not specified, the geographical domain used in the rotated EOF analysis is determined from the attributes of the input mesh-mask NetCDF variable named *grid_typedmask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit*, *lat2_Northern_limit*, *level1_First_level* and *level2_Last_level*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_filt_rot_eof_4d](#).

- 8) If the **-t=time1,time2** argument is missing, the whole time period associated with the original PC time series is used to compute the orthogonal rotation matrix.
- The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have *ntime = time2 - time1 + 1* observations.
- 9) The geographical shapes of the *eof_netcdf_variable* (in the *eof_netcdf_file*) and the land-sea mask (in the *input_mesh_mask_netcdf_file*) must agree.
 - 10) The **-pl=** argument specifies the minimum period of oscillation of the filtered PC time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 11) The **-ph=** argument specifies the maximum period of oscillation of the filtered PC time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument.

- 12) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the PC time series.
- 13) Setting **-pl=PL**, **-ph=PH** and $PH \ll PL$ is also allowed and performs band rejection of periods between *PH* and *PL*. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.
- 14) Setting both **-pl=0** and **-ph=0** is not allowed.
- 15) The **-tr=trend_removal** argument specifies pre- and post-filtering processing of the multichannel PC time series. If:

- **-tr=+/-1**, the mean of the PC time series is removed before time filtering

- **-tr=**+/-2, the drift from the PC time series is removed before time filtering. The drift for the PC time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
- **-tr=**+/-3, the least-squares line from the PC time series is removed before time filtering.

If **-tr=-1**, -2 or -3, the mean, drift or least-squares line are reintroduced post-filtering, respectively. The covariance matrix between the (detrended) and filtered PC time series is then computed and the eigenvectors of this symmetric positive-definite matrix, ranked in decreasing order of the associated positive eigenvalues are calculated in order to obtain the orthogonal rotation matrix to apply to the selected EOFs and PC time series.

For other values of the **-tr=** argument, nothing is done before or after filtering the PC time series.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 16) The **-win=** argument controls the form of the window, which will be convolved with the ideal filter response. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., the “ideal” filter).

The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1..

- 17) The **-mi=missing_value** argument specifies the missing value indicator associated with the rotated EOF NetCDF variable in the *output_rot_eof_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to 1.e+20.
- 18) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file, *output_rot_eof_netcdf_file*.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 19) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 21) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 22) It is assumed that the original EOF variable has no missing values excepted those associated with a constant land-sea mask.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on EOF analysis, PCA and orthogonal rotation of PC time series to selected-frequency components and windowed filtering of time series, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
- “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 487 pp., 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
- “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
- “Rotation of principal components” by Richman, M.B., International Journal of Climatology, Vol. 6, 293-335, 1986. <https://doi.org/10.1002/joc.3370060305>
- “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>
- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_filt_rot_eof_4d` creates an output NetCDF file that contains the rotated standardized PC time series, their associated rotated EOF patterns and the variances described by the computed rotated PC time series. The ratios of filtered variance to total variance for the rotated standardized PC time series are also stored in the output NetCDF file.

The number of rotated PCs and EOFs, `nrot`, stored in the output NetCDF dataset is determined by the `-se=selected_eofs` argument. The number of observations in the output NetCDF dataset is determined from the `-t=time1,time2` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat`, `nlon` and `nlev` are the length of the spatial and vertical dimensions of the input NetCDF variable `eof_netcdf_variable` and `nrot` is the number of selected PC time series, which have been rotated as determined by the list specified with the `-se=selected_eofs` argument) :

- 1) `pc_number(nrot)` : the list of rotated PC time series as specified with the `-se=selected_eofs` argument.
- 2) `eof_netcdf_variable_filt_rot_eof(nrot,nlev,nlat,nlon)` : the rotated EOF patterns. These rotated EOFs are scaled such that they give the scalar products (if `-a=scp` in the original EOF analysis), covariances (`-a=cov` in the original EOF analysis) or correlations (`-a=cor` in the original EOF analysis) between the original observed variables and the rotated PC time series.

The rotated EOF patterns are packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `eof_netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, this output NetCDF variable is filled with missing values.

- 3) `eof_netcdf_variable_filt_rot_pc(ntime,nrot)` : the rotated standardized PC time series.
The rotated PC time series are always standardized to unit variance.
- 4) `eof_netcdf_variable_filt_rot_std(nrot)` : the standard-deviations of the rotated PC time series.
The squares of these statistics are equal to the raw variance described by the rotated PC time series.
- 5) `eof_netcdf_variable_filt_rot_var(nrot)` : the proportion of variance explained by each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

- 6) *eof_netcdf_variable_filt_rot_ratio*(*nrot*) : the ratio of filtered variance to total variance for each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

Examples

- 1) For computing a rotated 10-EOF analysis towards biennial modes (e.g., time scales between 18 and 30 months) from a previous EOF analysis (stored in the file `eof_1m_197902_200501_votemper_oi.nc`) of a four-dimensional NetCDF variable named `votemper` and store the results of the rotated EOF analysis in a NetCDF file named `filt_rot_eof_1m_197902_200501_votemper_oi.nc`, use the following command :

```
$ comp_filt_rot_eof_4d \
-f=eof_1m_197902_200501_votemper_oi.nc \
-v=votemper \
-se=1:10 \
-pl=18 \
-ph=30 \
-m=oi_mask.nc \
-o=filt_rot_eof_1m_197902_200501_votemper_oi.nc
```

2.23 comp_freq_func_1d

2.23.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.23.2 Latest revision

30/04/2026

2.23.3 Purpose

Estimate the frequency response function (e.g., the transfer function) of a symmetric linear filter (e.g., a high, low or band-pass Lanczos or Hamming filter in this version of `comp_freq_func_1d`) as computed by *comp_lanczos_filter_3d* or *comp_symlin_filter_3d* (or the corresponding procedures for uni-dimensional or four-dimensional NetCDF variables).

For more details on the frequency response function of a frequency filter, see [[Bloomfield](#)].

The frequency response function is computed at `NFREQ` frequencies where `NFREQ` is the value given at the **-nf=** argument.

The `NFREQ` frequencies are regularly sampled between 0 and the Nyquist frequency if the optional argument **-fourfreq** is not used or are the `NFREQ` Fourier frequencies $2\pi j / \text{NFREQ}$ for $j=0$ to `NFREQ-1` if this argument is used.

The frequency response function is written in a NetCDF dataset on output for later use.

2.23.4 Further Details

Usage

```
$ comp_freq_func_ld \  
-nf=number_of_frequencies \  
-pl=minimum_period \  
-ph=maximum_period \  
-nfc=number_of_filter_coefficients (optional) \  
-o=output_netcdf_file (optional) \  
-mi=missing_value (optional) \  
-hamming (optional) \  
-win=window_choice (optional : 0.5 > 1.) \  
-fourfreq (optional) \  
-notestf (optional) \  
-double (optional) \  
-hdf5 (optional) \  
-cdf5 (optional) \  
-tlimited (optional)
```

By default

- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter.
- o=** the *output_netcdf_file* is named `filter_freq_func.nc`
- mi=** the *missing_value* for the output variable is equal to `1.e+20`
- hamming** a Lanczos window filter is used. If the **-hamming** argument is specified a Hamming window filter is used instead
- win=** a Hamming window (e.g., **-win=0.54**) is convolved with the filter response by default if the **-hamming** argument is used, meaning that a Hamming window filter is estimated. If the **-hamming** argument is not used, this argument has no effect
- fourfreq** the frequency response function is not computed at the Fourier frequencies. If the **-fourfreq** argument is used the frequency response function is estimated at the Fourier frequencies
- notestf** Normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where PH and PL are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the $[0., 0.5]$ frequency interval. By using the **-notestf** argument you can get ride of this limitation
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time (e.g., frequency) dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-nf=** argument specifies the number of frequencies at which the frequency response function must be evaluated. The **-nf=** argument is a strictly positive integer.

- 2) The **-pl=** argument specifies the minimum period of oscillation of the resulting filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 3) The **-ph=** argument specifies the maximum period of oscillation of the resulting filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the multichannel time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 4) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the time series.
- 5) Setting both **-pl=0** and **-ph=0** is not allowed since in that case, no frequency filtering is done.
- 6) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to *K*, the first and last $(K-1)/2$ time observations in the resulting filtered time series will be affected by end effects. Thus, the user must choose the number of filter terms, *K*, as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operations (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 7) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 8) If the **-hamming** argument is specified, a Hamming window filter is used instead of Lanczos window filter.
- 9) The **-win=** argument controls the form of the window which will be convolved with the filter if a Hamming window filter is requested with the **-hamming** argument. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., “ideal” filter).

This argument has an effect only if the **-hamming** argument is also specified. The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1.

- 10) If the **-fourfreq** argument is used the frequency response function is evaluated at the Fourier frequencies $2\pi j/NFREQ$ for $j=0$ to $NFREQ-1$, where $NFREQ$ is the value of the **-nf=** argument.
- 11) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $[0., 0.5]$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 12) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 13) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 14) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 15) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 16) For more details on Lanczos or symmetric linear filtering and examples of use in the climate literature, see

- “Fourier analysis of time series- An introduction”, by P. Bloomfield, John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Lanczos filtering in one and two dimensions”, by C. Duchon, 1016-1022, Journal of applied meteorology, vol. 18, 1979. [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

`comp_freq_func_1d` creates an output NetCDF file that contains the frequency response function of the selected filter. The output NetCDF dataset contains the following NetCDF variable (in the description below, *NFREQ* is the *number_of_frequencies* specified for the **-nf=** argument) :

- 1) `transfert_function(NFREQ)` : the frequency response function of the selected filter at the selected frequencies.

Examples

- 1) For estimating the frequency response function of a band-passed Lanczos filter (in the biennial time scale if we assume monthly time series since **-pl=18** and **-ph=30** are specified) at 200 frequencies regularly sampled between 0 and the Nyquist frequency, use the following commands :

```
$ comp_freq_func_1d \
-nf=200 \
-pl=18 \
-ph=30 \
-o=freq_func.nc
```

- 2) For estimating the frequency response function of the same band-passed Lanczos filter, but at the 200 Fourier frequencies between 0 and the Nyquist frequency, use the following commands :

```
$ comp_freq_func_1d \
-nf=200 \
-pl=18 \
-ph=30 \
-fourfreq \
-o=freq_func.nc
```

2.24 comp_index_1d

2.24.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.24.2 Latest revision

30/04/2026

2.24.3 Purpose

Compute an index time series from two time series extracted from two unidimensional variables in one or two NetCDF datasets and stored the new index time series in a NetCDF dataset.

The two original time series can be computed by *comp_serie_3d*, *comp_serie_miss_3d* or similar procedures for fourdimensional NetCDF variables.

Different operations are available for estimating the index time series from the original time series (see the description of the **-op=** argument below).

Different operations are also available for transforming or normalizing the index time series after its estimation from the two original time series (see the description of the **-a=** argument below).

2.24.4 Further Details

Usage

```
$ comp_index_1d \
-f=input_netcdf_file \
-v=netcdf_variable \
-v2=netcdf_variable2 \
-f2=input_netcdf_file2 (optional) \
-t=timea1,timea2 (optional) \
-t2=timeb1,timeb2 (optional) \
-a=type_of_transformation (optional : scp, cov, cor) \
-p=periodicity (optional) \
-o=output_netcdf_file (optional) \
-n=output_netcdf_variable (optional) \
-op=type_of_operation (optional : +, -, *, /) \
-sm=smoothing_factor (optional) \
-mi=missing_value (optional) \
-3d (optional) \
```

(continues on next page)

(continued from previous page)

-double	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- f2= the *input_netcdf_file2* is the same as *input_netcdf_file*. This means that the *netcdf_variable2* is extracted from the same NetCDF dataset as *netcdf_variable*
- t= the whole time period associated with the *netcdf_variable*
- t2= the whole time period associated with the *netcdf_variable2*
- a= the *type_of_transformation* is set to `scp`. This means that the index time series is computed from the raw original time series and is not normalized or transformed
- p= the *periodicity* is set to 1. This means that the index time series does not have a seasonal cycle.
- o= the *output_netcdf_file* is named `index_netcdf_variable.netcdf_variable2.nc`
- n= the *output_netcdf_variable* is named `netcdf_variable_netcdf_variable2_index`
- op= the *type_of_operation* is set to `-`. This means that the index time series is computed as the differences between the two original time series
- sm= no smoothing is applied to the index time series
- mi= the *missing_value* in the output NetCDF file is set to `1.e+20`
- 3d the *output_netcdf_variable* is defined as an unidimensional NetCDF variable. However, if **-3d** is activated, the *output_netcdf_variable* is defined as an tridimensional NetCDF variable but with two dummy dimensions (e.g., with a length equal to 1)
- double the data are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the data are stored as double-precision floating point numbers
- hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** specifies the first unidimensional time series for computing the index and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The **-v2=netcdf_variable2** specifies the second unidimensional time series for computing the index and the **-f2=input_netcdf_file2** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file2*. If the **-f2=** argument is absent, it is assumed that this *netcdf_variable2* is also in the NetCDF dataset specifies by the **-f=** argument.

- 3) If the **-t=timea1,timea2** argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the first time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = timea2 - timea1 + 1$ time observations.

- 4) If the **-t=timeb1,timeb2** argument is missing, the whole time period associated with the *netcdf_variable2* is used to compute the second time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = timeb2 - timeb1 + 1$ time observations.

- 5) The selected number of time observations by the **-t=** and **-t2=** arguments must be equal. This means that the following equality must hold

$$timea2 - timea1 + 1 = timeb2 - timeb1 + 1$$

otherwise, an error message will be issued and the program will stop.

- 6) The **-a=type_of_transformation** argument specifies how the index time series is normalized or transformed. If:

- **-a=scp**, the raw index time series is stored
- **-a=cov**, the anomalies of the raw index time series are computed and stored in the output NetCDF file
- **-a=cor**, the standardized anomalies of the raw index time series are computed and stored in the output NetCDF file.

The default is **-a=scp** meaning that the raw index time series is stored.

- 7) If the **-p=** argument is specified and **-a=cov** or **-a=cor**, the anomalies or the standardized anomalies are computed from the raw index time series by taken into account the *periodicity* of the data as specified by the **-p=** argument.

- 8) The **-op=type_of_operation** argument specifies how the index time series is computed from the two original time series. If:

- **-op=-**, the default, the index is computed as the difference between the two original time series
- **-op=+**, the index is computed as the sum of the two original time series
- **-op=***, the index is computed as the product of the two original time series
- **-op=/**, the index is computed as the quotient of the two original time series.

- 9) **-sm=smoothing_factor** means that the (transformed) index time series must be smoothed with a moving average of approximately $2 * smoothing_factor + 1$ terms. *smoothing_factor* must be an integer greater than 0. However, if missing values are present in the time original time series or generated in the computation of the index, smoothing is not allowed.

- 10) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contains the computed time series in the output NetCDF file, *output_netcdf_file*, specified by the **-o=output_netcdf_file** argument.

- 11) Missing values are allowed in both NetCDF variables associated with the two original time series. However, if missing values are present, the **-sm=** argument is not allowed.

- 12) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variable in the *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.

- 13) The **-3d** argument specify that the index time series must be stored as a tridimensional NetCDF variable with two dummy dimensions in the output NetCDF file.

By default, the index time series is stored as an unidimensional NetCDF variable.

- 14) The **-double** argument specifies that, the index time series is stored as double-precision floating point numbers in the output NetCDF file.

By default, the index time series is stored as single-precision floating point numbers in the output NetCDF file.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_index_1d` creates an output NetCDF file that contains the computed index time series. The output NetCDF dataset contains the following NetCDF variable (in the description below, `ntime` is the number of time steps selected with the **-t=** and **-t2=** arguments):

- 1) *output_netcdf_variable*(`ntime`) : the computed index time series defined as an unidimensional variable.

or if the **-3d** argument has been specified :

- 1) *output_netcdf_variable*(`ntime, 1, 1`) : the computed index time series defined as a tridimensional variable with two dummy dimensions.

Examples

- 1) For computing an index time series named `siod` as the difference between two unidimensional NetCDF variables both called `sst` and extracted, respectively, from the files `HadISST1_2m_187001_200702_swiosst_nt67.nc` and `HadISST1_2m_187001_200702_seiosst_nt67.nc`, and store the result in a file named `HadISST1_2m_187001_200702_sdisst_nt67.nc`, use the following command :

```
$ comp_index_1d \  
-f=HadISST1_2m_187001_200702_swiosst_nt67.nc -v=sst \  
-f2=HadISST1_2m_187001_200702_seiosst_nt67.nc -v2=sst \  
-o=HadISST1_2m_187001_200702_sdisst_nt67.nc -n=siod
```

2.25 comp_integral_4d

2.25.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.25.2 Latest revision

30/04/2026

2.25.3 Purpose

Perform vertical integrations (e.g., through the depth of the ocean or the height of the atmosphere) from a fourdimensional variable extracted from a NetCDF dataset.

Different options are available for computing the vertical integrals in the selected domain and for each time steps of the fourdimensional variable (see the description of the **-d=** argument below).

The computed integrals are stored in an output NetCDF dataset.

If the NetCDF variable has missing values, in addition to those associated with a constant atmosphere or ocean mask, use *comp_integral_miss_4d* instead of *comp_integral_4d*.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP flag. Moreover, this procedure computes the vertical integrals with only one pass through the data and an out-of-core strategy which is efficient on huge datasets.

2.25.4 Further Details

Usage

```
$ comp_integral_4d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -z=level1,level2            (optional) \
  -t=time1,time2              (optional) \
  -d=type_of_integral          (optional : dist3, ident) \
  -mi=missing_value           (optional) \
  -double                      (optional) \
  -compact                    (optional) \
  -bigfile                    (optional) \
  -hdf5                       (optional) \
  -cdf5                       (optional) \
  -tlimited                    (optional)
```

By default

- g=** the *grid_type* is set to `n`, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- d=** the *type_of_integral* is set to `dist3`. This means that the vertical integrals are computed as weighted averages and that the weight associated with each layer is proportional to the volume or the mass associated with it
- mi=** by default, the *missing_value* in the output NetCDF variable is set to `1.e+20`
- double** the integrals are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the integrals are stored as double-precision floating point numbers
- compact** the output NetCDF file is not compacted
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the vertical integrals must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used for computing the vertical integrals.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.

The **-z=** argument is used for determining the limits of the vertical integrations, which are performed.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using `comp_integral_4d`.
- 3) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used for computing the vertical integrals.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `n_time = time2 - time1 + 1` time observations.
- 4) It is assumed that the data has no missing values excepted those associated with a constant atmosphere or ocean mask (as specified by the input mesh-mask NetCDF file).

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to `n`, it is assumed that the 3-D grid-mesh is regular or Gaussian.

- 6) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (*input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*) must agree.

- 7) The **-d=** argument specifies the weighting method for computing the integrals. If:

- **-d=dist3**, the vertical integrals are computed as weighted averages. Each layer is weighted accordingly to the volume or weight associated with it. In this case, it is assumed that the input mesh-mask NetCDF file has a variable named “e3”//grid, where grid is the value of the **-g=** argument, which is the third scale factor associated with the grid-mesh of the input_netcdf_variable. In the case of a regular or gaussian grid, this third scale factor is automatically computed by *comp_clim_4d*, *comp_clim_miss_4d* or *comp_stat_4d*. This third scale factor variable provides the weights for computing the vertical integrals in *comp_integral_4d* if **-d=dist3** is specified.
- **-d=ident**, the vertical integrals are computed as simple sums. Each layer has the same weight (equals to 1) when computing the integrals. In this case, it is not required that the input mesh-mask NetCDF file contains a third scale factor as a variable.

- 8) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contain the computed vertical integrals in the output NetCDF file, *output_netcdf_file*, specified by the **-o=** argument.

- 9) The **-mi=missing_value** argument specifies the missing value indicator associated with the *output_netcdf_variable* in the *output_netcdf_file*. If the **-mi=** argument is not specified, *missing_value* is set to `1.e+20` in the output NetCDF dataset.

- 10) The **-double** argument specify that the interals are stored as double-precision floating point numbers in the output NetCDF file.

By default, the vertical integrals are stored as single-precision floating point numbers in the output NetCDF file.

- 11) If the **-compact** argument is specified and a domain is selected (with the **-x=** and **-y=** arguments) then only vertical integrals for the selected domain will be output. By default, the whole 2-D grid is stored (with missing values outside the selected domain) in the output NetCDF file.

- 12) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 13) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 14) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 15) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.

Outputs

`comp_integral_4d` creates an output NetCDF file that contains the vertical integrals computed for each time step and grid point selected from the input NetCDF variable and the horizontal coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file*. The NetCDF variable storing the integrals will have the same horizontal dimensions as the input NetCDF variable in the file *input_netcdf_file* (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable) and *ntime* time observations (corresponding to the number of time steps selected with `-t=` argument) :

- 1) *output_netcdf_variable* (`ntime, nlat, nlon`) : the computed vertical integrals for the select domain and time steps associated with the input NetCDF variable.

By default, the whole grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument `-compact` is used, the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the `-x=` and `-y=` arguments (e.g., in this case `nlat=lat2-lat1+1` and `nlon=lon2-lon1+1`). The number of time steps written in the output NetCDF file (e.g., `ntime`) is determined from the `-t=` argument.

Examples

- 1) For computing vertical integrals from the file `ST7_1m_0101_20012_grid_T_votemper.nc`, which includes a fourdimensional NetCDF variable `votemper`, and store the results in a NetCDF variable named `int_votemper` in the file `integral_temp.orca2.nc`, use the following command (note that the vertical integrals are computed as weighted averages for each time step and grid-point associated with the whole horizontal 2-D grid-mesh of the input NetCDF variable) :

```
$ comp_integral_4d \  
-f=ST7_1m_0101_20012_grid_T_votemper.nc \  
-v=votemper \  
-m=meshmask.orca2.nc \  
-o=integral_temp.orca2.nc \  
-n=int_votemper \  
-g=t \  
-d=dist3
```

2.26 comp_integral_miss_4d

2.26.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.26.2 Latest revision

30/04/2026

2.26.3 Purpose

Perform vertical integrations (e.g., through the depth of the ocean or the height of the atmosphere) from a fourdimensional variable with missing values extracted from a NetCDF dataset.

It is assumed that the input NetCDF variable has a scalar *missing_value* or *_FillValue* attribute, which allows a proper identification of the missing values in the dataset, otherwise the procedure will exit with an error message.

Different options are available for computing the vertical integrals in the selected domain and for each time steps of the fourdimensional variable (see the description of the **-d=** argument below).

The computed integrals are stored in an output NetCDF dataset.

If the NetCDF variable has no missing values, excepted those associated with a constant atmosphere or ocean mask, use *comp_integral_4d* instead of *comp_integral_miss_4d*.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP flag. Moreover, this procedure computes the vertical integrals with only one pass through the data and an out-of-core strategy which is efficient on huge datasets.

2.26.4 Further Details

Usage

```
$ comp_integral_miss_4d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -z=level1,level2            (optional) \
  -t=time1,time2              (optional) \
  -d=type_of_integral         (optional : dist3, ident) \
  -mi=missing_value           (optional) \
  -double                     (optional) \
  -compact                    (optional) \
  -bigfile                    (optional) \
  -hdf5                       (optional) \
  -cdf5                       (optional) \
  -tlimited                    (optional)
```

By default

- g=** the *grid_type* is set to *n*, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*

- d=** the *type_of_integral* is set to `dist3`. This means that the vertical integrals are computed as weighted averages and that the weight associated with each layer is proportional to the volume or the mass associated with it
- mi=** by default, the *missing_value* in the output NetCDF variable is set to `1.e+20`
- double** the integrals are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the integrals are stored as double-precision floating point numbers
- compact** the output NetCDF file is not compacted
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the vertical integrals must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used for computing the vertical integrals.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.

The **-z=** argument is used for determining the limits of the vertical integrations, which are performed.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using `comp_integral_miss_4d`.

- 3) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used for computing the vertical integrals.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.

- 4) It is assumed that the data has no missing values excepted those associated with a constant atmosphere or ocean mask (as specified by the input mesh-mask NetCDF file).
- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to `n`, it is assumed that the 3-D grid-mesh is regular or Gaussian.

- 6) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (*input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*) must agree.
- 7) It is assumed that the data have missing values in addition to those associated with a constant atmosphere or ocean mask. If it is not the case, use [comp_integral_4d](#) instead of `comp_integral_miss_4d`.

It is further assumed that the specified *netcdf_variable* has a scalar `missing_value` or `_FillValue` attribute and that missing values in the data are identified by the value of this `missing_value` or `_FillValue` attribute.

- 8) The **-d=** argument specifies the weighting method for computing the integrals. If:
 - **-d=dist3**, the vertical integrals are computed as weighted averages. Each layer is weighted accordingly to the volume or weight associated with it. In this case, it is assumed that the input mesh-mask NetCDF file has a variable named "e3"//grid, where grid is the value of the **-g=** argument, which is the third scale factor associated with the grid-mesh of the input_netcdf_variable. In the case of a regular or gaussian grid, this third scale factor is automatically computed by *comp_clim_4d*, *comp_clim_miss_4d* or *comp_stat_4d*. This third scale factor variable provides the weights for computing the vertical integrals in *comp_integral_miss_4d* if **-d=dist3** is specified.
 - **-d=ident**, the vertical integrals are computed as simple sums. Each layer has the same weight (equals to 1) when computing the integrals. In this case, it is not required that the input mesh-mask NetCDF file contains a third scale factor as a variable.
- 9) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contain the computed vertical integrals in the output NetCDF file, *output_netcdf_file*, specified by the **-o=** argument.
- 10) The **-mi=missing_value** argument specifies the missing value indicator associated with the *output_netcdf_variable* in the *output_netcdf_file*. If the **-mi=** argument is not specified, *missing_value* is set to 1.e+20 in the output NetCDF dataset.
- 11) The **-double** argument specify that the intervals are stored as double-precision floating point numbers in the output NetCDF file.
By default, the vertical integrals are stored as single-precision floating point numbers in the output NetCDF file.
- 12) If the **-compact** argument is specified and a domain is selected (with the **-x=** and **-y=** arguments) then only vertical integrals for the selected domain will be output. By default, the whole 2-D grid is stored (with missing values outside the selected domain) in the output NetCDF file.
- 13) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.
If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 14) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.
If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 15) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.
If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 16) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.

Outputs

comp_integral_miss_4d creates an output NetCDF file that contains the vertical integrals computed for each time step and grid point selected from the input NetCDF variable and the horizontal coordinate

NetCDF variables of the input NetCDF dataset *input_netcdf_file* . This NetCDF variable storing the integrals will have the same horizontal dimensions as the input NetCDF variable in the file *input_netcdf_file* (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the input NetCDF variable) and *ntime* time observations (corresponding to the number of time steps selected with **-t=** argument) :

- 1) *output_netcdf_variable* (*ntime*, *nlat*, *nlon*) : the computed vertical integrals for the select domain and time steps associated with the input NetCDF variable.

By default, the whole grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used, the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=** and **-y=** arguments (e.g., in this case *nlat*=*lat2-lat1*+1 and *nlon*=*lon2-lon1*+1). The number of time steps written in the output NetCDF file (e.g., *ntime*) is determined from the **-t=** argument.

Examples

- 1) For computing vertical integrals from the file *ST7_1m_0101_20012_grid_T_votemper.nc*, which includes a fourdimensional NetCDF variable *votemper*, and store the results in a NetCDF variable named *int_votemper* in the file *integral_temp.orca2.nc*, use the following command (note that the vertical integrals are computed as weighted averages for each time step and grid-point associated with the whole horizontal 2-D grid-mesh of the input NetCDF variable) :

```
$ comp_integral_miss_4d \  
-f=ST7_1m_0101_20012_grid_T_votemper.nc \  
-v=votemper \  
-m=meshmask.orca2.nc \  
-o=integral_temp.orca2.nc \  
-n=int_votemper \  
-g=t \  
-d=dist3
```

2.27 comp_invert_eof_3d

2.27.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.27.2 Latest revision

03/03/2026

2.27.3 Purpose

Approximate a tridimensional NetCDF variable (or parts of it) from its Empirical Orthogonal Function (EOF) decomposition or from results of a Singular Value Decomposition (SVD) analysis.

Using as input a NetCDF file produced by *comp_eof_3d*, *comp_eof_miss_3d* or *comp_svd_3d*, this procedure computes an approximation of a tridimensional NetCDF variable, packed as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 2-D grid-mesh associated with the tridimensional NetCDF variable), of the form:

$$\mathbf{X} = \mathbf{AB} + \mathbf{E}$$

where

- **A** is an *ntime* by *k* matrix of *k* selected principal component time series
- **B** is the *k* by *nv* matrix of the *k* associated eigenvectors (stored rowwise)
- **E** is an *ntime* by *nv* matrix of residuals.

If the selected principal components are the first *k*, when the principal components are sorted by descending order of the eigenvalues, the matrix product **AB** is a least-squares solution to the problem of minimizing of the sum of all the squared elements of **E**. In other words, the first *k* principal components of **X** are the best linear predictors of the observed variables among all possible sets of *k* variables.

This type of approximation can also be computed from the results of a previous SVD analysis if the argument **-svd** is activated, however in this case the computed approximation is not necessarily optimal in the least square sense.

If the NetCDF variable is fourdimensional use `comp_invert_eof_4d` instead of `comp_invert_eof_3d`.

An output NetCDF dataset containing the matrix product **AB** repacked as a tridimensional variable is created.

2.27.4 Further Details

Usage

```
$ comp_invert_eof_3d \
  -f=input_eof_netcdf_file \
  -v=netcdf_variable \
  -se=selected_eofs                (optional) \
  -x=lon1,lon2                     (optional) \
  -y=lat1,lat2                     (optional) \
  -t=time1,time2                   (optional) \
  -l=selected_time_period          (optional) \
  -a=type_of_analysis              (optional : scp, cov, cor) \
  -c=input_climatology_netcdf_file (optional) \
  -o=output_netcdf_file            (optional) \
  -mi=missing_value                (optional) \
  -svd                             (optional) \
  -double                          (optional) \
  -bigfile                         (optional) \
  -hdf5                            (optional) \
  -cdf5                            (optional) \
  -tlimited                         (optional)
```

By default

- se=** the selected principal components are those stored in the *input_eof_netcdf_file*; in other words, the NetCDF variable *netcdf_variable* is approximated with the number of EOFs (or SVDs) stored in the NetCDF file *eof_input_netcdf_file*
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- l=** the whole time period as determined by the **-t=** argument

- a=** the *type_of_analysis* is set to `scp`. This means that the eigenvectors (or singular vectors) and eigenvalues (or singular values) have been computed from the sums of squares and cross-products matrix between the observed variables if an EOF (or SVD) model is used
- c=** this argument is not used if the *type_of_analysis* is set to `scp`
- o=** the *output_netcdf_file* is named `approx_netcdf_variable.nc`
- mi=** the *missing_value* attribute in the output NetCDF file is set to `1.e+20`
- svd** the *input_eof_netcdf_file* is assumed to be produced by `comp_eof_3d` or `comp_eof_miss_3d`. However, if **-svd** is activated, a file produced by `comp_svd_3d` is assumed, this means that the approximation is done from a set of singular vectors and singular variables of a previous SVD analysis
- double** the results of the EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=***netcdf_variable* argument specifies the NetCDF variable which must be approximated from its EOF decomposition or a previous SVD analysis. The EOF (or SVD) model is extracted from the NetCDF file *input_eof_netcdf_file* specified by the **-f=** argument. This NetCDF file must have exactly the same format as the files produced by `comp_eof_3d` or `comp_svd_3d`.
- 2) The **-se=** argument allows the user to select the EOFs (or SVDs) which must be included in the approximation model. The EOFs (or SVDs) list may be given in two formats:
 - **-se=1, 3, . . . , nn** allows to include eof1, eof3, . . . and eofnn in the EOF (or SVD) model
 - **-se=1 : 4** allows to include from eof1 to eof4 in the EOF (or SVD) model.

The two forms of the **-se=** argument may be combined and repeated any number of times. Duplicate EOF (or SVD) numbers are not allowed. If the **-se=** argument is not specified, the NetCDF variable is approximated with the number of EOFs (or SVDs) stored in the *input_eof_netcdf_file*.

- 3) If the **-x=***lon1, lon2* and **-y=***lat1, lat2* arguments are missing the whole geographical domain associated with the *netcdf_variable* is approximated by the selected EOF (or SVD) model (as specified by the **-se=** argument).

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to `comp_mask_3d` for transforming geographical coordinates as indices before using `comp_invert_eof_3d`.

- 4) If the **-t=***time1, time2* argument is missing, the whole time period associated with the *netcdf_variable* is approximated by the selected EOF (or SVD) model.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ observations if the **-l=** argument is missing.

- 5) The **-l=** argument lists the indices of the time steps which must be included in the output file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is not used). The list may be specified in two formats:

- **-l=n1,n2,...nn** allows to select for *n1*, *n2*, ... and *nn* time steps
- **-l=n1:n2** allows to select time steps from *n1* to *n2*.

Be careful with time period limits, when specifying the **-l=** argument list.

The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed.

- 6) The **-a=** argument specifies if the observed variables have been centered or standardized with an input climatology (specified with the **-c=** argument) before the EOF (or SVD) analysis:

- **-a=scp** means that the EOF (or SVD) analysis was done on the raw data
- **-a=cov** means that the EOF (or SVD) analysis was done on the anomalies
- **-a=cor** means that the EOF (or SVD) analysis was done on the standardized anomalies.

In all cases, the raw data are approximated if the **-a=** argument is used.

- 7) The *input_climatology_netcdf_file* is needed only if **-a=cov** or **-a=cor**.
- 8) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 9) The **-svd** argument specifies that the *input_eof_netcdf_file* is produced by *comp_svd_3d* instead of *comp_eof_3d*. This means that the approximation will be done from the singular vectors and singular variables of a previous SVD analysis stored in *comp_svd_3d*.
- 10) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 11) The **-mi=missing_value** argument specifies the missing value indicator associated with the netcdf variable *netcdf_variable* in the NetCDF file *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to 1.e+20.
- 12) The **-double** argument specify that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 13) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 14) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 16) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 17) For more details on EOF or SVD analysis in the climate literature, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300

Outputs

`comp_invert_eof_3d` creates an output NetCDF file that contains a least squares approximation of the input NetCDF variable computed from selected eigenvectors and principal components of an EOF analysis or from singular vectors and variables of a SVD analysis.

This least squares approximation is repacked as a tridimensional NetCDF variable with the same dimensions as the input NetCDF variable specified in the **-v=** argument.

This output NetCDF dataset contains the following NetCDF variable (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable in the initial EOF or SVD analysis; `ntime` is the number of time steps selected with the **-l=** and **-t=** arguments) :

- 1) *netcdf_variable*(`ntime`, `nlat`, `nlon`) : a least squares approximation of the input NetCDF variable computed with the help of the selected EOF (or SVD) model.

Examples

- 1) For computing a 10-EOF approximation of a NetCDF variable named `sst` from the NetCDF file `eof_HadISST1_2m_197902_200501_sst_oi.nc` produced by `comp_eof_3d` and store the results in a NetCDF file named `HadISST1_2m_197902_200501_sst_oi_10pc.nc`, use the following command :

```
$ comp_invert_eof_3d \  
-f=eof_HadISST1_2m_197902_200501_sst_oi.nc \  
-v=sst \  
-a=scp \  
-n=10 \  
-o=HadISST1_2m_197902_200501_sst_oi_10pc.nc
```

2.28 comp_invert_eof_4d

2.28.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.28.2 Latest revision

30/04/2026

2.28.3 Purpose

Approximate a fourdimensional NetCDF variable (or parts of it) from its Empirical Orthogonal Function (EOF) decomposition or from results of a Singular Value Decomposition (SVD) analysis.

Using as input a NetCDF file produced by `comp_eof_4d` or `comp_svd_3d`, this procedure computes an approximation of a fourdimensional NetCDF variable, packed as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 3-D grid-mesh associated with the tridimensional NetCDF variable), of the form:

$$\mathbf{X} = \mathbf{AB} + \mathbf{E}$$

where

- **A** is an *ntime* by *k* matrix of *k* selected principal component time series
- **B** is the *k* by *nv* matrix of the *k* associated eigenvectors (stored rowwise)
- **E** is an *ntime* by *nv* matrix of residuals.

If the selected principal components are the first *k* when the principal components are sorted by descending order of the eigenvalues, the matrix product **AB** is a least-squares solution to the problem of minimizing of the sum of all the squared elements of **E**. In other words, the first *k* principal components of **X** are the best linear predictors of the observed variables among all possible sets of *k* variables.

This type of approximation can also be computed from the results of a previous SVD analysis if the argument **-svd** is activated, however in this case the computed approximation is not necessarily optimal in the least square sense.

If the NetCDF variable is tridimensional use `comp_invert_eof_3d` instead of `comp_invert_eof_4d`.

An output NetCDF dataset containing the matrix product **AB** repacked as a fourdimensional variable is created.

2.28.4 Further Details

Usage

```
$ comp_invert_eof_4d \
  -f=input_eof_netcdf_file \
  -v=netcdf_variable \
  -se=selected_eofs           (optional) \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -z=level1,level2            (optional) \
  -t=time1,time2              (optional) \
  -l=selected_time_period     (optional) \
  -a=type_of_analysis         (optional : scp, cov, cor) \
```

(continues on next page)

(continued from previous page)

```

-c=input_climatology_netcdf_file  (optional) \
-o=output_netcdf_file             (optional) \
-mi=missing_value                 (optional) \
-svd                              (optional) \
-double                           (optional) \
-bigfile                          (optional) \
-hdf5                             (optional) \
-cdf5                             (optional) \
-tlimited                          (optional)

```

By default

- se=** the selected principal components are those stored in the *input_eof_netcdf_file*; in other words, the NetCDF variable *netcdf_variable* is approximated with the number of EOFs (or SVDs) stored in the NetCDF file *eof_input_netcdf_file*
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- l=** the whole time period as determined by the **-t=** argument
- a=** the *type_of_analysis* is set to `scp`. This means that the eigenvectors (or singular vectors) and eigenvalues (or singular values) have been computed from the sums of squares and cross-products matrix between the observed variables if an EOF (or SVD) model is used
- c=** this argument is not used if the *type_of_analysis* is set to `scp`
- o=** the *output_netcdf_file* is named `approx_netcdf_variable.nc`
- mi=** the *missing_value* attribute in the output NetCDF file is set to `1.e+20`
- svd** the *input_eof_netcdf_file* is assumed to be produced by *comp_eof_4d*. However, if **-svd** is activated, a file produced by *comp_svd_3d* is assumed, this means that the approximation is done from a set of singular vectors and singular variables of a previous SVD analysis
- double** the results of the EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be approximated from its EOF analysis. The EOF model is extracted from the NetCDF file *input_eof_netcdf_file* specified by the **-f=** argument.

This NetCDF file must have exactly the same format as the files produced by *comp_eof_4d* or *comp_svd_3d*.

- 2) The **-se=** argument allows the user to select the EOFs (or SVDs) which must be included in the approximation model. The EOFs (or SVDs) list may be given in two formats:

- **-se=1, 3, . . . , nn** allows to include eof1, eof3, . . . and eofnn in the EOF (or SVD) model
- **-se=1 : 4** allows to include from eof1 to eof4 in the EOF (or SVD) model.

The two forms of the **-se=** argument may be combined and repeated any number of times. Duplicate EOF (or SVD) numbers are not allowed. If the **-se=** argument is not specified, the NetCDF variable is approximated with the number of EOFs (or SVDs) stored in the *input_eof_netcdf_file*.

- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is approximated by the selected EOF (or SVD) model (as specified by the **-se=** argument).

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_4d* for transforming geographical coordinates as indices before using *comp_invert_eof_4d*.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is approximated by the selected EOF (or SVD) model.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ observations if the **-l=** argument is missing.

- 5) The **-l=** argument lists the indices of the time steps which must be included in the output file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is not used). The list may be specified in two formats:

- **-l=n1,n2, . . . nn** allows to select for *n1*, *n2*, . . . and *nn* time steps
- **-l=n1:n2** allows to select time steps from *n1* to *n2*.

Be careful with time period limits, when specifying the **-l=** argument list.

The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed.

- 6) The **-a=** argument specifies if the observed variables have been centered or standardized with an input climatology (specified with the **-c=** argument) before the EOF (or SVD) analysis:

- **-a=scp** means that the EOF (or SVD) analysis was done on the raw data
- **-a=cov** means that the EOF (or SVD) analysis was done on the anomalies
- **-a=cor** means that the EOF (or SVD) analysis was done on the standardized anomalies.

In all cases, the raw data are approximated if the **-a=** argument is used.

- 7) The *input_climatology_netcdf_file* is needed only if **-a=cov** or **-a=cor**.
- 8) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 9) The **-svd** argument specifies that the *input_eof_netcdf_file* is produced by *comp_svd_3d* instead of *comp_eof_4d*. This means that the approximation will be done from the singular vectors and singular variables of a previous SVD analysis stored in *comp_svd_3d*.

- 10) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 11) The **-mi=missing_value** argument specifies the missing value indicator associated with the netcdf variable *netcdf_variable* in the NetCDF file *output_netcdf_file*.

If the **-mi=** argument is not specified *missing_value* is set to 1.e+20.

- 12) The **-double** argument specify that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 13) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 14) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 16) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 17) For more details on EOF or SVD analysis in the climate literature, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300

Outputs

`comp_invert_eof_4d` creates an output NetCDF file that contains a least squares approximation of the input NetCDF variable computed from selected eigenvectors and principal components of an EOF analysis or from singular vectors and variables of a SVD analysis.

This least squares approximation is repacked as a fourdimensional NetCDF variable with the same dimensions as the EOFs (or SVDs) of the input NetCDF variable specified in the **-v=** argument.

This output NetCDF dataset contains the following NetCDF variable (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the vertical and spatial dimensions of the input NetCDF variable; `ntime` is the number of time steps selected with the **-l=** and **-t=** arguments) :

- 1) *netcdf_variable*(*ntime*,*nlev*,*nlat*,*nlon*) : a least squares approximation of the input NetCDF variable computed with the help of the selected EOF (or SVD) model.

Examples

- 1) For computing a 10-EOF approximation of a NetCDF variable named *votemper* from the NetCDF file *eof_votemper.t.nc* produced by *comp_eof_4d* and store the results in a NetCDF file named *ST7_1m_0101_20012_grid_T_votemper_10pc.nc*, use the following command :

```
$ comp_invert_eof_4d \
-f=eof_votemper.t.nc \
-v=votemper \
-a=scp \
-n=10 \
-o=ST7_1m_0101_20012_grid_T_votemper_10pc.nc
```

2.29 comp_lanczos_filter_1d

2.29.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.29.2 Latest revision

30/04/2026

2.29.3 Purpose

Filter a real time series in a selected frequency band by Lanczos filtering [Bloomfield] [Duchon]. The time series is extracted from a uni- or bidimensional variable readed from a NetCDF dataset and can also be detrended before Lanczos filtering at the user option.

The number of coefficients used to build the Lanczos filter can be selected by the user and the Lanczos filter can be applied to the time series in the time or frequency domain, also at the user option. This gives to the user some control on the desired end-effects of the filter (e.g., applying the filter in the frequency domain assumes implicitly that the time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series; on the other hand, applying the filter in the time domain implies some loss of data or some distortions of the desired response function of the filter at both ends of the filtered time series).

Additionally, the filtering can be done separately for different segments of equal length of the selected time series if this time series is not continuous in time.

The frequency response function (e.g., the transfer function) of the selected Lanczos filter can be computed by *comp_freq_func_1d*. See the references cited below for more details on Lanczos filtering [Bloomfield] [Duchon].

This procedure returns the filtered real time series in a NetCDF dataset. If the NetCDF variable is tri- or four-dimensional use *comp_lanczos_filter_3d* or *comp_lanczos_filter_4d*, respectively, instead of *comp_lanczos_filter_1d*. If the time series has a seasonal (or diurnal) cycle, use *comp_stl_1d* in order to estimate and remove the harmonic components of the time series before using *comp_lanczos_filter_1d*.

If you need more control on the filtering parameters, including other windows, use *comp_symlin_filter_1d* instead of *comp_lanczos_filter_1d*.

2.29.4 Further Details

Usage

```
$ comp_lanczos_filter_ld \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-t=time1,time2 (optional) \  
-o=output_netcdf_file (optional) \  
-ni=index_for_2d_netcdf_variable (optional) \  
-p=periodicity (optional) \  
-pl=minimum_period (optional) \  
-ph=maximum_period (optional) \  
-tr=trend_removal (optional : 0, 1, 2, 3, -1, -2, -3) \  
-nfc=number_of_filter_coefficients (optional) \  
-mi=missing_value (optional) \  
-notestf (optional) \  
-usefft (optional) \  
-double (optional) \  
-hdf5 (optional) \  
-cdf5 (optional) \  
-tlimited (optional)
```

By default

- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- ni=** if the *netcdf_variable* is bidimensional, the first time series is used
- p=** the *periodicity* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** the *trend_removal* is set to 0, which means that no detrending is done before filtering
- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter
- mi=** the *missing_value* for the output variable is equal to $1.e+20$
- notestf** normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where PH and PL are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the $[0., 0.5]$ frequency interval. By using the **-notestf** argument you can get ride of this limitation
- usefft** the Lanczos filter is applied in the time domain. When you specify the **-usefft** argument the filter will be applied in the frequency domain, using an FFT algorithm and multiplication, instead of a convolution in the time domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

-cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a Lanczos filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.

2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

3) The **-ni=index_for_2d_netcdf_variable** argument specifies the index for selecting the time series if the *netcdf_variable* is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *index_for_2d_netcdf_variable* to 1.

4) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *periodicity* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *periodicity*.

5) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

6) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

7) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the time series.

8) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.

9) The **-tr=** argument specifies pre- and post-filtering processing of the time series. If:

- **-tr=+/-1**, the mean of the time series is removed before time filtering
- **-tr=+/-2**, the drift from the time series is removed before time filtering. The drift for the time series is estimated using the formula : $drift = (tseries(n_{time}) - tseries(1)) / (n_{time} - 1)$
- **-tr=+/-3**, the least-squares line from the time series is removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the mean, drift or least-squares line are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 10) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients for the filtering of the time series. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to K , the first and last $(K-1)/2$ time observations in the output NetCDF file will be affected by end effects. Thus, the user must choose the number of filter terms, K , as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operation (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 11) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 12) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $[0., 0.5]$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 13) The **-usefft** argument specifies that the filter must be applied in the frequency domain by using a Fast Fourier Transform and the convolution theorem. Moreover, if the **-usefft** argument is specified, the values at the ends of the output filtered series are computed implicitly by assuming that the input series is part of a periodic sequence.
- 14) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) If the time series has a seasonal or diurnal cycle, use *comp_stl_1d* to remove the pure harmonic components from the time series before filtering.

- 18) It is assumed that the data has no missing values.
- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 20) For more details on Lanczos filtering and examples of use in the climate literature, see
 - “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
 - “Lanczos filtering in one and two dimensions”, by Duchon, C., Journal of applied meteorology, vol. 18, 1016-1022, 1979. doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

`comp_lanczos_filter_1d` creates an output NetCDF file that contains the filtered time series estimated from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_filt(ntime)` : the filtered time series for the time series associated with the input NetCDF variable.

Examples

- 1) For Lanczos filtering a real monthly time series between 18 and 30 months (e.g., biennial time scale) from a NetCDF variable called `sst` extracted from the file `sst.monthly.nino34.nc`, which includes a monthly time series, and store the results in the NetCDF file `qbo_sst_nino34.nc`, use the following command :

```
$ comp_lanczos_filter_1d \
  -f=sst.monthly.nino34.nc \
  -v=sst \
  -pl=18 \
  -ph=30 \
  -o=qbo_sst_nino34.nc
```

2.30 comp_lanczos_filter_3d

2.30.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.30.2 Latest revision

30/04/2026

2.30.3 Purpose

Filter a real multi-channel time series in a selected frequency band by Lanczos filtering [Bloomfield] [Duchon]. The multi-channel time series is extracted from a tridimensional variable readed from a NetCDF dataset and can also be detrended before Lanczos filtering at the user option.

The number of coefficients used to build the Lanczos filter can be selected by the user and the Lanczos filter can be applied to the multi-channel time series in the time or frequency domain, also at the user option. This gives to the user some control on the desired end-effects of the filter (e.g., applying the filter in the frequency domain assumes implicitly that the multi-channel time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series; on the other hand, applying the filter in the time domain implies some loss of data or some distortions of the desired response function of the filter at both ends of the filtered time series).

Additionally, the filtering can be done separately for different segments of equal length of the selected multi-channel time series if this time series is not continuous in time.

The frequency response function (e.g., the transfer function) of the selected Lanczos filter can be computed by `comp_freq_func_1d`. See the references cited below for more details on Lanczos filtering [Bloomfield] [Duchon].

This procedure returns the filtered real multi-channel time series in a NetCDF dataset. If the NetCDF variable is uni- or fourdimensional use `comp_lanczos_filter_1d` or `comp_lanczos_filter_4d`, respectively, instead of `comp_lanczos_filter_3d`. If the multi-channel time series has a seasonal (or diurnal) cycle, use `comp_clim_3d` and `comp_norm_3d` or `comp_stl_3d` in order to estimate and remove the harmonic components of the time series before using `comp_lanczos_filter_3d`.

If you need more control on the filtering parameters, including other windows, use `comp_symlin_filter_3d` instead of `comp_lanczos_filter_3d`.

This procedure is parallelized if OpenMP is used.

2.30.4 Further Details

Usage

```
$ comp_lanczos_filter_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file           (optional) \
-g=grid_type                             (optional : n, t, u, v, w, f) \
-x=lon1,lon2                             (optional) \
-y=lat1,lat2                             (optional) \
-t=time1,time2                           (optional) \
-o=output_netcdf_file                    (optional) \
-p=periodicity                           (optional) \
-pl=minimum_period                       (optional) \
-ph=maximum_period                       (optional) \
-tr=trend_removal                        (optional : 0, 1, 2, 3, -1, -2, -3) \
-nfc=number_of_filter_coefficients       (optional) \
-mi=missing_value                        (optional) \
-ngp=number_of_grid_points                (optional) \
-notestf                                 (optional) \
-usefft                                  (optional) \
-double                                 (optional) \
-bigfile                                 (optional) \
-hdf5                                    (optional) \
-cdf5                                    (optional) \
-tlimited                                 (optional)
```

By default

-m= an `input_mesh_mask_netcdf_file` is not used

- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- p=** the *periodicity* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** the *trend_removal* is set to 0, which means that no detrending is done before filtering
- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter
- mi=** the *missing_value* for the output variable is equal to $1.e+20$
- ngp=** the *number_of_grid_points* is set to the number of grid points in the selected domain
- notestf** normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where *PH* and *PL* are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the $[0., 0.5]$ frequency interval. By using the **-notestf** argument you can get ride of this limitation
- usefft** the Lanczos filter is applied in the time domain. When you specify the **-usefft** argument the filter will be applied in the frequency domain, using an FFT algorithm, multiplication and an inverse FFT instead of a convolution in the time domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a Lanczos filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF *mesh_mask* variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).
If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_lanczos_filter_3d*.

- 5) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 6) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *periodicity* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *periodicity*.

- 7) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 8) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the multi-channel time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the multi-channel time series.
- 10) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.
- 11) The **-tr=** argument specifies pre- and post-filtering processing of the multi-channel time series. If:
- **-tr=+/-1**, the means of the time series are removed before time filtering
 - **-tr=+/-2**, the drifts from the time series are removed before time filtering. The drift for each time series is estimated using the formula: $drift = (tseries(n_{time}) - tseries(1)) / (n_{time} - 1)$
 - **-tr=+/-3**, the least-squares lines from the multi-channel time series are removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the means, drifts or least-squares lines are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 12) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients for the filtering of the multi-channel time series. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to K , the first and last $(K-1)/2$ time observations in the output NetCDF file will be affected by end effects. Thus, the user must choose the number of filter terms, K , as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operation (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 13) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 14) The **-ngp=** argument can be used if you have memory problems when running `comp_lanczos_filter_3d` on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.

- 15) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $[0., 0.5]$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 16) The **-usefft** argument specifies that the Lanczos filter must be applied in the frequency domain by using a Fast Fourier Transform and the convolution theorem. Moreover, if the **-usefft** argument is specified, the values at the ends of the output filtered series are computed implicitly by assuming that the input series is part of a periodic sequence.

- 17) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The `-cdf5` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) If the multi-channel time series has a seasonal or diurnal cycle, use *comp_stl_3d* or *comp_clim_3d* to remove the pure harmonic components from the time series before filtering.
- 22) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on Lanczos filtering and examples of use in the climate literature, see
- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
 - “Lanczos filtering in one and two dimensions”, by Duchon, C., Journal of applied meteorology, vol. 18, 1016-1022, 1979. doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

comp_lanczos_filter_3d creates an output NetCDF file that contains the filtered time series estimated from the multi-channel time series associated with the input NetCDF variable. The output NetCDF data set contains the following NetCDF variable (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the input NetCDF variable and *ntime* is the selected number of time observations) :

- 1) *netcdf_variable_filt* (*ntime*, *nlat*, *nlon*) : the filtered time series for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

The filtered multi-channel time series is packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For filtering a real multi-channel monthly time series between 18 and 30 months (e.g., biennial time scale) from a tridimensional NetCDF variable called *mslp* extracted from the file *mslp.monthly.mean_ncep2.nc*, which includes monthly time series, and store the results in the NetCDF file *tbo_mslp_ncep2.nc*, use the following command :

```
$ comp_lanczos_filter_3d \  
-f=mslp.monthly.mean_ncep2.nc \  
-v=mslp \  
-m=mesh_mask_mslp_ncep2.nc \  
-pl=18 \  
-ph=30 \  
-o=tbo_mslp_ncep2.nc
```

2.31 comp_lanczos_filter_4d

2.31.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.31.2 Latest revision

30/04/2026

2.31.3 Purpose

Filter a real multi-channel time series in a selected frequency band by Lanczos filtering [Bloomfield] [Duchon]. The multi-channel time series is extracted from a fourdimensional variable readed from a NetCDF dataset and can also be detrended before Lanczos filtering at the user option.

The number of coefficients used to build the Lanczos filter can be selected and the Lanczos filter can be applied to the multi-channel time series in the time or frequency domain, also at the user option. This gives to the user some control on the desired end-effects of the filter (e.g., applying the filter in the frequency domain assumes implicitly that the multi-channel time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series; on the other hand, applying the filter in the time domain implies some loss of data or some distortions of the desired response function of the filter at both ends of the filtered time series).

Additionally, the filtering can be done separately for different segments of equal length of the selected multi-channel time series if this time series is not continuous in time.

The frequency response function (e.g., the transfer function) of the selected Lanczos filter can be computed by *comp_freq_func_1d*. See the references cited below for more details on Lanczos filtering [Bloomfield] [Duchon].

This procedure returns the filtered real multi-channel time series in a NetCDF dataset. If the NetCDF variable is uni- or tridimensional use *comp_lanczos_filter_1d* or *comp_lanczos_filter_3d*, respectively, instead of *comp_lanczos_filter_4d*. If the multi-channel time series has a seasonal (or diurnal) cycle, use *comp_clim_4d* and *comp_norm_4d* or *comp_stl_4d* in order to estimate and remove the harmonic components of the time series before using *comp_lanczos_filter_4d*.

If you need more control on the filtering parameters, including other windows, use *comp_symlin_filter_4d* instead of *comp_lanczos_filter_4d*.

This procedure is parallelized if OpenMP is used.

2.31.4 Further Details

Usage

```
$ comp_lanczos_filter_4d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file           (optional) \
  -g=grid_type                             (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                             (optional) \
  -y=lat1,lat2                             (optional) \
  -z=level1,level2                         (optional) \
  -t=time1,time2                           (optional) \
```

(continues on next page)

(continued from previous page)

```

-o=output_netcdf_file           (optional) \
-p=periodicity                 (optional) \
-pl=minimum_period             (optional) \
-ph=maximum_period             (optional) \
-tr=trend_removal              (optional : 0, 1, 2, 3, -1, -2, -3) \
-nfc=number_of_filter_coefficients (optional) \
-mi=missing_value              (optional) \
-ngp=number_of_grid_points      (optional) \
-notestf                       (optional) \
-usefft                       (optional) \
-double                       (optional) \
-bigfile                       (optional) \
-hdf5                         (optional) \
-cdf5                         (optional) \
-tlimited                       (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to `n` which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named `filt_netcdf_variable.nc`
- p=** the *periodicity* is set to `time2 - time1 + 1`, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** the *trend_removal* is set to 0, which means that no detrending is done before filtering
- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter
- mi=** the *missing_value* for the output variable is equal to `1.e+20`
- ngp=** the *number_of_grid_points* is set to the number of grid points in the selected domain
- notestf** normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where `PH` and `PL` are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the `[0., 0.5]` frequency interval. By using the **-notestf** argument you can get ride of this limitation
- usefft** the Lanczos filter is applied in the time domain. When you specify the **-usefft** argument the filter will be applied in the frequency domain, using an FFT algorithm and multiplication, instead of a convolution in the time domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a Lanczos filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical and vertical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF mesh_mask variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $n_{lon} + lon1 + 1$ to *lon2* where n_{lon} is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_lanczos_filter_4d*.

- 5) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 6) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *periodicity* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *periodicity*.

- 7) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 8) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the multi-channel time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is computed and applied to the multi-channel time series.
- 10) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.
- 11) The **-tr=** argument specifies pre- and post-filtering processing of the multi-channel time series. If
 - **-tr=+/-1**, the means of the time series are removed before time filtering
 - **-tr=+/-2**, the drifts from the time series are removed before time filtering. The drift for each time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
 - **-tr=+/-3**, the least-squares lines from the multi-channel time series are removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the means, drifts or least-squares lines are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 12) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients for the filtering of the multi-channel time series. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to *K*, the first and last $(K-1)/2$ time observations in the output NetCDF file will be affected by end effects. Thus, the user must choose the number of filter terms, *K*, as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operation (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 13) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 14) The **-ngp=** argument can be used if you have memory problems when running `comp_lanczos_filter_4d` on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.
- 15) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $(0.0, 0.5)$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 16) The **-usefft** argument specifies that the Lanczos filter must be applied in the frequency domain by using a Fast Fourier Transform and the convolution theorem. Moreover, if the **-usefft** argument is specified, the values at the ends of the output filtered series are computed implicitly by assuming that the input series is part of a periodic sequence.
- 17) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) If the multi-channel time series has a seasonal or diurnal cycle, use *comp_stl_4d* or *comp_clim_4d* to remove the pure harmonic components from the time series before filtering.
- 22) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on Lanczos filtering and examples of use in the climate literature, see

- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Lanczos filtering in one and two dimensions”, by Duchon, C., Journal of applied meteorology, vol. 18, 1016-1022, 1979. doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

comp_lanczos_filter_4d creates an output NetCDF file that contains the filtered time series estimated from the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, *nlev*, *nlat* and *nlon* are the lengths of the geographical dimensions of the input NetCDF variable and *ntime* is the selected number of time observations) :

- 1) `netcdf_variable_filt(ntime, nlev, nlat, nlon)` : the filtered time series for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

The filtered multi-channel time series is packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For filtering a real multi-channel monthly time series between 18 and 30 months (e.g., biennial time scale) from a fourdimensional NetCDF variable called `uwnd` extracted from the file `uwnd.monthly.mean.ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `qbo_uwnd_ncep2.nc`, use the following command :

```
$ comp_lanczos_filter_4d \
-f=uwnd.monthly.mean.ncep2.nc \
-v=uwnd \
-m=mesh_mask_uwnd_ncep2.nc \
-pl=18 \
-ph=30 \
-o=qbo_uwnd_ncep2.nc
```

2.32 comp_loess_rot_eof_3d

2.32.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.32.2 Latest revision

30/04/2026

2.32.3 Purpose

Performs an orthogonal rotation of selected standardized Principal Components (PC) time series extracted from a previous Empirical Orthogonal Function (EOF) analysis or Principal Component Analysis (PCA) of a tridimensional NetCDF variable towards low-frequency or high-frequency components using a LOESS smoother [Wills_etal] [Cleveland] [Cleveland_Devlin] . The associated factor loading matrix (e.g., the associated scaled EOF patterns corresponding to the selected rotated PC time series) are also rotated and the new rotated EOF or PCA model is stored in a NetCDF file.

Spatial EOF patterns (e.g., scaled EOFs) and PC time series derived from EOF analysis or PCA are constrained to be both spatially and temporally orthogonal to one another so that they maximize the variance they described over the entire analysis domain and selected time period [Jolliffe] [vonStorch_Zwiers] . However, the spatial orthogonality of the EOFs is a strong and undesirable constraint imposed on the EOFs and, as a result, EOFs are subject to problems such as domain dependence and inaccurate representation of physical relationships embedded in the data [Jolliffe] [Richman] .

Orthogonal rotation of selected standardized PC time series can be used to obtain spatial or temporal patterns (e.g., rotated scaled EOFs and PC time series) that are hopefully more physically meaningful by relaxing the spatial orthogonality of the rotated scaled EOFs, while preserving the temporal orthogonality of the rotated standardized PC time

series. Preserving the orthogonality of the (rotated) PC time series implies that these rotated PCs describe the same amount of total variance as the original (unrotated) PC time series, which is a very useful property, but this variance is more equally (or differently) distributed among the rotated PC time series [Jolliffe] [Richman] .

In classical orthogonal rotation PCA methods (e.g., like varimax rotation, see [comp_ortho_rot_eof_3d](#) for details), the objective is to make the rotated standardized PC time series as simple as possible to interpret by attempting to constrain the spatial coefficients (e.g., the scaled EOFs) associated with each of them to be large or very small in absolute magnitude, but avoiding as much as possible spatial loadings of moderate size [Jolliffe] [Jackson] . On the other hand, here in [comp_loess_rot_eof_3d](#), the focus of the orthogonal rotation is to obtain a cleaner representation of the different time scales embedded in the selected PC time series such that these time scales will not be mixed on the same PC time series after the orthogonal rotation [Wills_etal] .

It can be shown that a rotation matrix, which will maximize such clean representation of the time scales of interest, can be estimated as the eigenvectors of the covariance matrix between the selected standardized PC time series filtered in the frequency bands of interest. This type of generic method is called Low-Frequency Components Analysis (LFCA) [Wills_etal] .

Here, LFCA is considered as an orthogonal rotation method of the original standardized PC time series of a (partial) EOF or PCA model. The `nrot` by `nrot` orthogonal matrix (where `nrot` is the number of rotated PCs) used to rotate the original PCs is computed as the eigenvectors of the covariance (e.g., symmetric positive-definite) matrix between the filtered selected standardized PC time series [Wills_etal] .

In [comp_loess_rot_eof_3d](#), the filtering of the original PC time series is performed with the help of a LOESS smoother specified by the values of the `-nt=`, `-itdeg=`, `-ntjump=` and `-a=` arguments. See [Cleveland] [Cleveland_Devlin] for more details on the LOESS smoother used here. In the LOESS procedure, the analyzed PC time series are decomposed into two terms:

$$\mathbf{PC}(t) = \mathbf{T}(t) + \mathbf{R}(t)$$

where t refers to a time index, the \mathbf{T} term is used to quantify the trend and low-frequency variations in the PC time series, and, finally, the \mathbf{R} term contains the residual (high-frequency) component of the PC time series. The trend is estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., LOESS) to data windows whose length is chosen by the user (see description of the `-nt=` and `-itdeg=` arguments below for details) and the residual component is estimated by difference with the original PC time series. This LOESS smoother is a good candidate for extracting frequency-defined series components from time series. This filter is also implemented in [comp_trend_3d](#), where more details are provided.

This LOESS smoother is applied to the selected PC time series before their covariance matrix and its associated eigenvectors are computed. At the user option, the covariance matrix can be computed from the trend or residual components of the PC time series (see description of the `-a=` argument below for details). In the first case, the rotated standardized PC time series will be ordered from low- to high-frequency modes and, in the second case, they will be ordered from high- to low-frequency modes.

The purpose here is exactly the same as in [comp_filt_rot_eof_3d](#) excepted that [comp_loess_rot_eof_3d](#) uses a LOESS smoother [Cleveland] [Cleveland_Devlin] instead of a windowed filtering [Iacobucci_Noullez] to filter the original PC time series.

In summary, using as input an EOF NetCDF file produced by [comp_eof_3d](#) or [comp_eof_miss_3d](#), [comp_loess_rot_eof_3d](#) extracts selected PC time series from this input file (see the description of `-se=` argument below for details) and performs the orthogonal rotation of the selected standardized PC time series towards modes that have more energy in the frequency bands of interest selected with the help of the `-nt=`, `-itdeg=` and `-a=` arguments. [comp_loess_rot_eof_3d](#) also reads the associated scaled EOFs from the input EOF NetCDF file and computes the regression spatial patterns associated with the rotated standardized PC time series (e.g., the orthogonal projection onto the orthonormal set formed by the new rotated standardized PC time series). The percentages of variance described by the new rotated standardized PC time series are also computed and stored in the output NetCDF file, as are the ratios of filtered variance to total variance for the rotated standardized PC time series.

The selected standardized PC time series, which will be rotated, are determined on entry by the value of the `-se=` argument, which is a list of PC (or EOF) numbers.

An output NetCDF dataset containing these rotated PC standardized time series, their associated regression spatial patterns and the explained variance and ratio statistics is created.

If the NetCDF variable used in the original EOF analysis is fourdimensional use `comp_loess_rot_eof_4d` instead of `comp_loess_rot_eof_3d`.

2.32.4 Further Details

Usage

```
$ comp_loess_rot_eof_3d\
-f=eof_netcdf_file \
-v=eof_netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-nt=trend_smoother_length      (nt) \
-se=selected_eofs              (optional) \
-g=grid_type                   (optional : n, t, u, v, w, f) \
-r=resolution                  (optional : r2, r4) \
-b=nlon_orca, nlat_orca        (optional) \
-x=lon1,lon2                   (optional) \
-y=lat1,lat2                   (optional) \
-t=time1,time2                 (optional) \
-itdeg=trend_smoother_degree  (itdeg) (optional : 0, 1, 2) \
-ntjump=trend_skipping_value  (ntjump) (optional) \
-a=type_of_analysis           (optional : trend, residual) \
-o=output_rot_eof_netcdf_file (optional) \
-mi=missing_value             (optional) \
-double                       (optional) \
-bigfile                      (optional) \
-hdf5                        (optional) \
-cdf5                        (optional) \
-tlimited                      (optional)
```

By default

- se=** all the PC time series stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF EOF variable, *eof_netcdf_variable*, is assumed to be regular or Gaussian
- r=** if the input *eof_netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *eof_netcdf_variable*
- y=** the whole latitude domain associated with the *eof_netcdf_variable*
- t=** the whole time period associated with the input PC time series
- itdeg=** the *trend_smoother_degree* is set to 1
- ntjump=** the *trend_skipping_value* is set to *nt*/10 where *nt* is the value of the *trend_smoother_length* argument

- a=** the *type_of_analysis* is set to `trend`. This means that the trend components of the PC times series are used to compute the orthogonal rotation matrix
- o=** the *output_rot_eof_netcdf_file* is named `loess_rot_eof_netcdf_variable.nc`
- mi=** the *missing_value* attribute for the rotated EOFs in the output NetCDF file is set to `1.e+20`
- double** the results of the rotated EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=*eof_netcdf_variable*** argument specifies the NetCDF variable for which an EOF analysis was originally computed by *comp_eof_3d* (or *comp_eof_miss_3d*) and the **-f=*input_eof_netcdf_file*** argument specifies that the resulting EOF patterns and PC time series must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_3d* or *comp_eof_miss_3d*.

These EOF patterns and PC time series are used to compute the orthogonal rotation matrix which will be applied to the selected EOF model.

- 2) The **-m=*input_mesh_mask_netcdf_file*** argument specifies the land-sea mask to apply to the *eof_netcdf_variable* (which contains the scaled eigenvectors of the EOF analysis) for transforming this tridimensional NetCDF variable as a rectangular matrix before performing the rotated EOF analysis. It is assumed that this land-sea mask is exactly the same as the one used in the original EOF analysis, whose results are stored in the input NetCDF file *eof_netcdf_file*.
- 3) The **-se=*selected_eofs*** argument allows the user to select the standardized PC time series which must be included in the orthogonal rotation. The list of selected PCs may be given in two formats:
 - **-se=1,3,...,nn** allows to include *pc1*, *pc3*,... and *pcnn* in the orthogonal rotation
 - **-se=1 : 4** allows to include from *pc1* to *pc4* in the orthogonal rotation.

The two forms of the **-se=** argument may be combined and repeated any number of times, but duplicate PC numbers are not allowed. If the **-se=** argument is not specified, all the PC time series stored in the *input_eof_netcdf_file* are used in the orthogonal rotation.

- 4) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model. In this case, the duplicate points from the ORCA grid are removed before the rotated EOF analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to `n`, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask in the *input_mesh_mask_netcdf_file* (e.g., the variable named *grid_typemask*). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_3d* if the 2-D grid-mesh is regular or gaussian.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (e.g., if the input NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:

- **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
- **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.

- 6) If the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

- 7) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are not specified, the geographical domain used in the rotated EOF analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typemask* (e.g., `lon1_Eastern_limit`, `lon2_Western_limit`, `lat1_Southern_limit` and `lat2_Northern_limit`), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where `nlon` is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_loess_rot_eof_3d](#).

- 8) If the **-t=time1,time2** argument is missing, the whole time period associated with the original PC time series is used to compute the orthogonal rotation matrix.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` observations.

- 9) The geographical shapes of the *eof_netcdf_variable* (in the *eof_netcdf_file*) and the land-sea mask (in the *input_mesh_mask_netcdf_file*) must agree.
- 10) The **-nt=** argument specifies the length of the trend smoother, `nt`. The value of `nt` should be an odd integer greater than or equal to 3. As `nt` increases the values of the trend components of the PC time series, which are used to compute the orthogonal rotation matrix (if **-a=trend**) become smoother.
- 11) The **-itdeg=** argument specifies the degree of the locally-fitted polynomial in trend smoothing of the PC time series. The value is 0, 1 or 2.
- 12) The **-ntjump=** argument specifies the skipping value for trend smoothing of the PC time series. The trend smoother skips ahead `ntjump` points and then linearly interpolates in between. The value of `ntjump` should be a positive integer; if `ntjump` is set to 1, a trend smooth is calculated at all points in the PC time series. To make the procedure run faster, a reasonable choice for `ntjump` is 10% or 20% of `nt`.
- 13) The **-a=** argument specifies if the residuals from the trend components or the trend components are used to compute the orthogonal rotation matrix. If:
- **-a=trend**, the trend components are used.
 - **-a=residual**, the residuals from the trends are used.

The default is **-a=trend**, e.g., the trend components are used to estimate the orthogonal rotation matrix. More precisely, the covariance matrix between the trend components of the PC time series is then computed and the eigenvectors of this symmetric positive-definite matrix, ranked in decreasing order of the associated (positive) eigenvalues are calculated in order to obtain the orthogonal rotation matrix to apply to the selected EOFs and PC time series.

- 14) The **-mi=missing_value** argument specifies the missing value indicator associated with the rotated EOF NetCDF variable in the *output_rot_eof_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1. \times 10^{20}$.
- 15) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file, *output_rot_eof_netcdf_file*.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 16) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) It is assumed that the original EOF variable has no missing values excepted those associated with a constant land-sea mask.
- 20) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 21) For more details on EOF analysis, PCA and orthogonal rotation of PC time series to selected-frequency components and LOESS smoothing of time series, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
- “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 487 pp., 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
- “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
- “Rotation of principal components” by Richman, M.B., International Journal of Climatology, Vol. 6, 293-335, 1986. <https://doi.org/10.1002/joc.3370060305>
- “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>
- “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: 10.1080/01621459.1979.10481038

- “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: [10.1080/01621459.1988.10478639](https://doi.org/10.1080/01621459.1988.10478639)
- “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>
- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_loess_rot_eof_3d` creates an output NetCDF file that contains the rotated standardized PC time series, their associated rotated EOF patterns and the variances described by the computed rotated PC time series. The ratios of filtered variance to total variance for the rotated standardized PC time series are also stored in the output NetCDF file.

The number of rotated PCs and EOFs, `nrot`, stored in the output NetCDF dataset is determined by the `-se=selected_eofs` argument. The number of observations in the output NetCDF dataset is determined from the `-t=time1,time2` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable `eof_netcdf_variable` and `nrot` is the number of selected PC time series, which have been rotated, as determined by the list specified with the `-se=selected_eofs` argument) :

- 1) `pc_number(nrot)` : the list of rotated PC time series as specified with the `-se=selected_eofs` argument.
- 2) `eof_netcdf_variable_loess_rot_eof(nrot, nlat, nlon)` : the rotated EOF patterns. These rotated EOFs are scaled such that they give the scalar products (if `-a=scp` in the original EOF analysis), covariances (`-a=cov` in the original EOF analysis) or correlations (`-a=cor` in the original EOF analysis) between the original observed variables and the rotated PC time series.

The rotated EOF patterns are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `eof_netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, this output NetCDF variable is filled with missing values.

- 3) `eof_netcdf_variable_loess_rot_pc(ntime, nrot)` : the rotated standardized PC time series.
The rotated PC time series are always standardized to unit variance.
- 4) `eof_netcdf_variable_loess_rot_std(nrot)` : the standard-deviations of the rotated PC time series.
The squares of these statistics are equal to the raw variance described by the rotated PC time series.
- 5) `eof_netcdf_variable_loess_rot_var(nrot)` : the proportion of variance explained by each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).
- 6) `eof_netcdf_variable_loess_rot_ratio(nrot)` : the ratio of filtered variance to total variance for each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

Examples

- 1) For computing a rotated 10-EOF analysis towards low-frequency modes from a previous EOF analysis (stored in the file `eof_HadISST1_1m_197902_200501_sst_oi.nc`) of a tridimensional NetCDF variable named ```sst`, which contains monthly SST data; and stored the results of the rotated EOF analysis in a NetCDF file named `loess_rot_eof_HadISST1_1m_197902_200501_sst_oi.nc`, use the following command :


```
$ comp_loess_rot_eof_3d \
-f=eof_HadISST1_1m_197902_200501_sst_oi.nc \
-v=sst \
-se=1:10 \
-nt=48 \
-m=HadISST1_mask.nc \
-o=loess_rot_eof_HadISST1_1m_197902_200501_sst_oi.nc
```

2.33 comp_loess_rot_eof_4d

2.33.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.33.2 Latest revision

30/04/2026

2.33.3 Purpose

Performs an orthogonal rotation of selected standardized Principal Components (PC) time series extracted from a previous Empirical Orthogonal Function (EOF) analysis or Principal Component Analysis (PCA) of a fourdimensional NetCDF variable towards low-frequency or high-frequency components using a LOESS smoother [Wills_etal] [Cleveland] [Cleveland_Devlin]. The associated factor loading matrix (e.g., the associated scaled EOF patterns corresponding to the selected rotated PC time series) are also rotated and the new rotated EOF or PCA model is stored in a NetCDF file.

Spatial EOF patterns (e.g., scaled EOFs) and PC time series derived from EOF analysis or PCA are constrained to be both spatially and temporally orthogonal to one another so that they maximize the variance they described over the entire analysis domain and selected time period [Jolliffe] [vonStorch_Zwiers]. However, the spatial orthogonality of the EOFs is a strong and undesirable constraint imposed on the EOFs and, as a result, EOFs are subject to problems such as domain dependence and inaccurate representation of physical relationships embedded in the data [Jolliffe] [Richman].

Orthogonal rotation of selected standardized PC time series can be used to obtain spatial or temporal patterns (e.g., rotated scaled EOFs and PC time series) that are hopefully more physically meaningful by relaxing the spatial orthogonality of the rotated scaled EOFs, while preserving the temporal orthogonality of the rotated standardized PC time series. Preserving the orthogonality of the (rotated) PC time series implies that these rotated PCs describe the same amount of total variance as the original (unrotated) PC time series, which is a very useful property, but this variance is more equally (or differently) distributed among the rotated PC time series [Jolliffe] [Richman].

In classical orthogonal rotation PCA methods (e.g., like varimax rotation, see [comp_ortho_rot_eof_4d](#) for details), the objective is to make the rotated standardized PC time series as simple as possible to interpret by attempting to constrain the spatial coefficients (e.g., the scaled EOFs) associated with each of them to be large or very small in absolute magnitude, but avoiding as much as possible spatial loadings of moderate size [Jolliffe] [Jackson]. On the other hand, here in `comp_loess_rot_eof_4d`, the focus of the orthogonal rotation is to obtain a cleaner representation of the different time scales embedded in the selected PC time series such that these time scales will not be mixed on the same PC time series after the orthogonal rotation [Wills_etal].

It can be shown that a rotation matrix, which will maximize such clean representation of the time scales of interest, can be estimated as the eigenvectors of the covariance matrix between the selected standardized PC time series filtered

in the frequency bands of interest. This type of method is called Low-Frequency Components Analysis (LFCA) [Wills_etal] .

Here, LFCA is considered as an orthogonal rotation method of the original standardized PC time series of a (partial) EOF or PCA model. The `nrot` by `nrot` orthogonal matrix (where `nrot` is the number of rotated PCs) used to rotate the original PCs is computed as the eigenvectors of the covariance (e.g., symmetric positive-definite) matrix between the filtered selected standardized PC time series [Wills_etal] .

In `comp_loess_rot_eof_4d`, the filtering of the original PC time series is performed with the help of a LOESS smoother specified by the values of the `-nt=`, `-itdeg=`, `-ntjump=` and `-a=` arguments. See [Cleveland] [Cleveland_Devlin] for more details on the LOESS smoother used here. In the LOESS procedure, the analyzed PC time series are decomposed into two terms:

$$\mathbf{PC}(t) = \mathbf{T}(t) + \mathbf{R}(t)$$

where t refers to a time index, the \mathbf{T} term is used to quantify the trend and low-frequency variations in the PC time series, and, finally, the \mathbf{R} term contains the residual (high-frequency) component of the PC time series. The trend is estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., LOESS) to data windows whose length is chosen by the user (see description of the `-nt=` and `-itdeg=` arguments below for details) and the residual component is estimated by difference with the original PC time series. This LOESS smoother is a good candidate for extracting frequency-defined series components from time series. This filter is also implemented in `comp_trend_4d`, where more details are provided.

This LOESS smoother is applied to the selected PC time series before their covariance matrix and its associated eigenvectors are computed. At the user option, the covariance matrix can be computed from the trend or residual components of the PC time series (see description of the `-a=` argument below for details). In the first case, the rotated standardized PC time series will be ordered from low- to high-frequency modes and, in the second case, they will be ordered from high- to low-frequency modes.

The purpose here is exactly the same as in `comp_filt_rot_eof_4d` excepted that `comp_loess_rot_eof_4d` uses a LOESS smoother [Cleveland] [Cleveland_Devlin] instead of a windowed filtering [Iacobucci_Noullez] to filter the original PC time series.

In summary, using as input an EOF NetCDF file produced by `comp_eof_4d`, `comp_loess_rot_eof_4d` extracts selected PC time series from this input file (see the description of `-se=` argument below for details) and performs the orthogonal rotation of the selected standardized PC time series towards modes that have more energy in the frequency bands of interest selected with the help of the `-nt=`, `-itdeg=` and `-a=` arguments. `comp_loess_rot_eof_4d` also reads the associated scaled EOFs from the input EOF NetCDF file and computes the regression spatial patterns associated with the rotated standardized PC time series (e.g., the orthogonal projection onto the orthonormal set formed by the new rotated standardized PC time series). The percentages of variance described by the new rotated standardized PC time series are also computed and stored in the output NetCDF file, as are the ratios of filtered variance to total variance for the rotated standardized PC time series.

The selected standardized PC time series, which will be rotated, are determined on entry by the value of the `-se=` argument, which is a list of PC (or EOF) numbers.

An output NetCDF dataset containing these rotated PC standardized time series, their associated regression spatial patterns and the explained variance and ratio statistics is created.

If the NetCDF variable used in the original EOF analysis is tridimensional use `comp_loess_rot_eof_3d` instead of `comp_loess_rot_eof_4d`.

2.33.4 Further Details

Usage

```
$ comp_loess_rot_eof_4d\
  -f=eof_netcdf_file \
  -v=eof_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -nt=trend_smoother_length      (nt) \
  -se=selected_eofs              (optional) \
  -g=grid_type                   (optional : n, t, u, v, w, f) \
  -r=resolution                  (optional : r2, r4) \
  -b=nlon_orca, nlat_orca, nlevel_orca (optional) \
  -x=lon1,lon2                  (optional) \
  -y=lat1,lat2                  (optional) \
  -z=level1,level2              (optional) \
  -t=time1,time2                (optional) \
  -itdeg=trend_smoother_degree   (itdeg) (optional : 0, 1, 2 ) \
  -ntjump=trend_skipping_value   (ntjump) (optional) \
  -a=type_of_analysis            (optional : trend, residual) \
  -o=output_rot_eof_netcdf_file (optional) \
  -mi=missing_value              (optional) \
  -double                        (optional) \
  -bigfile                       (optional) \
  -hdf5                          (optional) \
  -cdf5                          (optional) \
  -tlimited                       (optional)
```

By default

- se=** all the PC time series stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF EOF variable, *eof_netcdf_variable*, is assumed to be regular or Gaussian
- r=** if the input *eof_netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-n=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *eof_netcdf_variable*
- y=** the whole latitude domain associated with the *eof_netcdf_variable*
- z=** the whole vertical resolution associated with the *eof_netcdf_variable*
- t=** the whole time period associated with the input PC time series
- itdeg=** the *trend_smoother_degree* is set to 1
- ntjump=** the *trend_skipping_value* is set to *nt/10* where *nt* is the value of the *trend_smoother_length* argument
- a=** the *type_of_analysis* is set to *trend*. This means that the trend components of the PC times series are used to compute the orthogonal rotation matrix
- o=** the *output_rot_eof_netcdf_file* is named *loess_rot_eof_netcdf_variable.nc*
- mi=** the *missing_value* attribute for the rotated EOFs in the output NetCDF file is set to *1.e+20*

- double** the results of the rotated EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=eof_netcdf_variable** argument specifies the NetCDF variable for which an EOF analysis was originally computed by *comp_eof_4d*) and the **-f=input_eof_netcdf_file** argument specifies that the resulting EOF patterns and PC time series must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_4d*.

These EOF patterns and PC time series are used to compute the orthogonal rotation matrix which will be applied to the selected EOF model.

- 2) The **-m=input_mesh_mask_netcdf_file** argument specifies the land-sea mask to apply to the *eof_netcdf_variable* (which contains the scaled eigenvectors of the EOF analysis) for transforming this fourdimensional NetCDF variable as a rectangular matrix before performing the rotated EOF analysis. It is assumed that this land-sea mask is exactly the same as the one used in the original EOF analysis, whose results are stored in the input NetCDF file *eof_netcdf_file*.
- 3) The **-se=selected_eofs** argument allows the user to select the standardized PC time series which must be included in the orthogonal rotation. The list of selected PCs may be given in two formats:
 - **-se=1,3,...,nn** allows to include *pc1,pc3,...* and *pcnn* in the orthogonal rotation
 - **-se=1 : 4** allows to include from *pc1* to *pc4* in the orthogonal rotation.

The two forms of the **-se=** argument may be combined and repeated any number of times, but duplicate PC numbers are not allowed. If the **-se=** argument is not specified, all the PC time series stored in the *input_eof_netcdf_file* are used in the orthogonal rotation.

- 4) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model. In this case, the duplicate points from the ORCA grid are removed before the rotated EOF analysis, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 3-D mesh-mask in the *input_mesh_mask_netcdf_file* (e.g., the variable named *grid_ttypemask*). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_4d* if the 3-D grid-mesh is regular or gaussian.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model

- **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.
- 6) If the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
 - 7) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are not specified, the geographical domain used in the rotated EOF analysis is determined from the attributes of the input mesh-mask NetCDF variable named *grid_typedmask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit*, *lat2_Northern_limit*, *level1_First_level* and *level2_Last_level*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_loess_rot_eof_4d*.

- 8) If the **-t=time1,time2** argument is missing, the whole time period associated with the original PC time series is used to compute the orthogonal rotation matrix.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ observations.
- 9) The geographical shapes of the *eof_netcdf_variable* (in the *eof_netcdf_file*) and the land-sea mask (in the *input_mesh_mask_netcdf_file*) must agree.
- 10) The **-nt=** argument specifies the length of the trend smoother, *nt*. The value of *nt* should be an odd integer greater than or equal to 3. As *nt* increases the values of the trend components of the PC time series, which are used to compute the orthogonal rotation matrix (if **-a=trend**) become smoother.
- 11) The **-itdeg=** argument specifies the degree of the locally-fitted polynomial in trend smoothing of the PC time series. The value is 0, 1 or 2.
- 12) The **-ntjump=** argument specifies the skipping value for trend smoothing of the PC time series. The trend smoother skips ahead *ntjump* points and then linearly interpolates in between. The value of *ntjump* should be a positive integer; if *ntjump* is set to 1, a trend smooth is calculated at all points in the PC time series. To make the procedure run faster, a reasonable choice for *ntjump* is 10% or 20% of *nt*.
- 13) The **-a=** argument specifies if the residuals from the trend components or the trend components are used to compute the orthogonal rotation matrix. If:
 - **-a=trend**, the trend components are used.
 - **-a=residual**, the residuals from the trends are used.

The default is **-a=trend**, e.g., the trend components are used to estimate the orthogonal rotation matrix. More precisely, the covariance matrix between the trend components of the PC time series is then computed and the eigenvectors of this symmetric positive-definite matrix, ranked in decreasing order of the associated (positive) eigenvalues are calculated in order to obtain the orthogonal rotation matrix to apply to the selected EOFs and PC time series.

- 14) The **-mi=missing_value** argument specifies the missing value indicator associated with the rotated EOF NetCDF variable in the *output_rot_eof_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.
- 15) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file, *output_rot_eof_netcdf_file*.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 16) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) It is assumed that the original EOF variable has no missing values excepted those associated with a constant land-sea mask.
- 20) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 21) For more details on EOF analysis, PCA and orthogonal rotation of PC time series to selected-frequency components and LOESS smoothing of time series, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
- “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 487 pp., 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
- “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
- “Rotation of principal components” by Richman, M.B., International Journal of Climatology, Vol. 6, 293-335, 1986. <https://doi.org/10.1002/joc.3370060305>
- “Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures”, by Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L., Geophysical Research Letters, Vol. 45, 2487-2496, 2018. <https://doi.org/10.1002/2017GL076327>
- “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: 10.1080/01621459.1979.10481038
- “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: 10.1080/01621459.1988.10478639

- “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>
- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_loess_rot_eof_4d` creates an output NetCDF file that contains the rotated standardized PC time series, their associated rotated EOF patterns and the variances described by the computed rotated PC time series. The ratios of filtered variance to total variance for the rotated standardized PC time series are also stored in the output NetCDF file.

The number of rotated PCs and EOFs, `nrot`, stored in the output NetCDF dataset is determined by the `-se=selected_eofs` argument. The number of observations in the output NetCDF dataset is determined from the `-t=time1,time2` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the vertical and spatial dimensions of the input NetCDF variable `eof_netcdf_variable` and `nrot` is the number of selected PC time series, which have been rotated as determined by the list specified with the `-se=selected_eofs` argument) :

- 1) `pc_number(nrot)` : the list of rotated PC time series as specified with the `-se=selected_eofs` argument.
- 2) `eof_netcdf_variable_loess_rot_eof(nrot,nlev,nlat,nlon)` : the rotated EOF patterns. These rotated EOFs are scaled such that they give the scalar products (if `-a=scp` in the original EOF analysis), covariances (`-a=cov` in the original EOF analysis) or correlations (`-a=cor` in the original EOF analysis) between the original observed variables and the rotated PC time series.

The rotated EOF patterns are packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `eof_netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, this output NetCDF variable is filled with missing values.

- 3) `eof_netcdf_variable_loess_rot_pc(ntime,nrot)` : the rotated standardized PC time series.

The rotated PC time series are always standardized to unit variance.

- 4) `eof_netcdf_variable_loess_rot_std(nrot)` : the standard-deviations of the rotated PC time series.

The squares of these statistics are equal to the raw variance described by the rotated PC time series.

- 5) `eof_netcdf_variable_loess_rot_var(nrot)` : the proportion of variance explained by each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).
- 6) `eof_netcdf_variable_loess_rot_ratio(nrot)` : the ratio of filtered variance to total variance for each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

Examples

- 1) For computing a rotated 10-EOF analysis towards low-frequency modes from a previous EOF analysis (stored in the file `eof_1m_197902_200501_votemper_oi.nc`) of a fourdimensional NetCDF variable named `votemper` and store the results of the rotated EOF analysis in a NetCDF file named `loess_rot_eof_1m_197902_200501_votemper_oi.nc`, use the following command :

```
$ comp_loess_rot_eof_4d \  
-f=eof_1m_197902_200501_votemper_oi.nc \  
-v=votemper \  
-se=1:10 \  
-nt=48 \  
-m=oi_mask.nc \  
-o=loess_rot_eof_1m_197902_200501_votemper_oi.nc
```

2.34 comp_mask_3d

2.34.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.34.2 Latest revision

30/04/2026

2.34.3 Purpose

Create a mesh-mask variable by specifying geographical bounds in indices or degrees and/or by modifying an input mesh-mask readed from a NetCDF dataset and store the resulting mesh-mask variable in a NetCDF dataset if the **-o=** argument (described below) is specified.

If an output NetCDF dataset is not specified with the **-o=** argument, the geographical bounds of the selected domain are printed as indices on output of the procedure. Such indices can then be used for specifying the **-x=** and **-y=** arguments for the selected domain in other NCSTAT procedures such as use [comp_serie_3d](#).

If you want to create a mesh-mask variable for a fourdimensional NetCDF variable, use [comp_mask_4d](#) instead of `comp_mask_3d`.

2.34.4 Further Details

Usage

```
$ comp_mask_3d \  
-f=input_netcdf_file \  
-vlon=longitude_variable \  
-vlat=latitude_variable \  
-g=grid_type (optional : n, t, u, v, w, f) \  
-x=lon1,lon2 (optional) \  
-y=lat1,lat2 (optional) \  
-a=type_of_coordinates (optional : indices, degrees) \  
-imsk=in_mask_value (optional : ident, 0, 1) \  
-omsk=out_mask_value (optional : ident, 0, 1) \  
-fmsk=input_mask_netcdf_file (optional) \  
-vmask=input_mask_netcdf_variable (optional) \  
-val=mask_value (optional) \  
-rel=mask_relation (optional : eq, gt, ge, lt, le) \  
-vout=output_mask_netcdf_variable (optional) \  

```

(continues on next page)

(continued from previous page)

-o=output_mask_netcdf_file	(optional) \
-noscalfac	(optional) \
-hdf5	(optional) \
-cdf5	(optional)

By default

- g= the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input *latitude_variable* and *longitude_variable* is assumed to be regular or Gaussian
- x= the whole longitude range associated with the *longitude_variable*
- y= the whole latitude range associated with the *latitude_variable*
- a= the *type_of_coordinates* is set to *indices*. This means that the geographical bounds specified in the -x= and -y= arguments are given as indices not in degrees of longitude and latitude
- imsk= the *in_mask_value* is set to 1. This means that points inside the specified domain are selected
- omsk= the *out_mask_value* is set to 0. This means that points outside the specified domain are not selected
- fmsk= an *input_mask_netcdf_file* is not used
- vmsk= an *input_mask_netcdf_variable* is not used
- val= the *mask_value* is set to 1.
- rel= the *mask_relation* is set to *eq*
- vout= the *output_mask_netcdf_variable* is named *grid_type/mask* if an output NetCDF file is created
- o= an *output_mask_netcdf_file* is not created
- noscalfac if the -fmsk= argument is present, the scale factors NetCDF variables, if they exist in the *input_mask_netcdf_file*, are automatically copied to the output NetCDF file. If the -noscalfac argument is specified, the scale factors NetCDF variables are not copied in the output NetCDF file
- hdf5 a NetCDF classical format file is created. If -hdf5 is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5 a NetCDF classical format file is created. If -cdf5 is activated, the output NetCDF file is a CDF5 format file

Remarks

- 1) The -vlon= and -vlat= arguments specify the coordinate variables from which the mesh-mask variable will be constructed and the -f=input_netcdf_file argument specifies that these coordinate NetCDF variables must be extracted from the NetCDF file *input_netcdf_file*. The longitude and latitude coordinate NetCDF variables may be one or two dimensional arrays.
- 2) If the -x=lon1,lon2 and -y=lat1,lat2 arguments are missing the whole geographical domain associated with the *longitude_variable* and *latitude_variable* is used to construct the mesh-mask.

If -a=indices (the default value), the longitude (specified with the -x= argument) or latitude (specified with the -y= argument) range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

If -a=degrees, the longitude or latitude range must be a vector of two integers specifying the longitude and latitude limits of the domain in degrees of longitude or latitude, respectively. In that case, negative values are

allowed for *lon1*, *lon2*, *lat1* and *lat2*. Moreover, the longitudes *lon1* and *lon2* are shifted to be in the interval between -180 and 180 degrees, assuming that the grid is periodic in longitude.

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the input 2-D grid-mesh associated with the input coordinate variables is from an experiment with the ORCA model (R2, R4 or R05 resolutions).

This argument is also used to determine the name of the *output_mask_netcdf_variable* in the output NetCDF file if the **-vout=** argument is not specified. In that case, the *output_mask_netcdf_variable* will be named *grid_type//mask*.

- 4) The **-imsk=** argument specifies the value (0, 1 or *ident*) to be given to grid points inside of the specified domain (as specified by the **-x=** and **-y=** arguments). By default, values inside the specified domain are set to 1 (e.g., **-imsk=1**).

If **-imsk=ident**, the **-vmsk=** and **-fmsk=** arguments must also be given and the values inside the selected domain are read from the *input_mask_netcdf_variable* and are modified according to the logical relation determined by the **-rel=** and **-val=** arguments.

- 5) The **-omsk=** argument is identical to **-imsk=** argument, but concerns points outside the specified domain (as specified by the **-x=** and **-y=** arguments). By default, values outside the specified domain are set to 0 (**-omsk=0**).

If **-omsk=ident**, the **-vmsk=** and **-fmsk=** arguments must also be given and the values outside the selected domain are read from the *input_mask_netcdf_variable* and are modified according to the logical relation determined by the **-rel=** and **-val=** arguments.

- 6) If **-imsk=ident** or **-omsk=ident**, the **-fmsk=** and **-vmsk=** arguments must be specified. In that case, the output mesh-mask variable *output_mask_netcdf_variable* is first initialized from the *input_mask_netcdf_variable* (as specified by the **-vmsk=** and **-fmsk=** arguments) inside the domain if **-imsk=ident** or outside the domain if **-omsk=ident**, and then modified in these regions according to the logical relationship:

- **output_mask(i,j)** = 1 if **input_mask(i,j)** .*mask_relation*. *mask_value* is *true*
- **output_mask(i,j)** = 0 otherwise

where *mask_relation* is determined from the **-rel=** argument and *mask_value* from the **-val=** argument.

By default, *mask_relation* is set to *eq* and *mask_value* is set to 1. . . *mask_value* must be specified as a real number.

The **-fmsk=** argument must be present, if the **-vmsk=** argument is specified, otherwise the program will stop.

If the **-fmsk=** and **-vmsk=** arguments are not specified, the output mesh-mask variable is computed by applying the *in_mask_value* value (as specified by the **-imsk=** argument) inside the specified domain and the *out_mask_value* (as specified by the **-omsk=** argument) outside the specified domain.

- 7) The geographical shapes of the *input_mask_netcdf_variable* (in the *input_mask_netcdf_file*) must agree with the input coordinate variables *longitude_variable* and *latitude_variable* (in the *input_netcdf_file*) if an *input_mask_netcdf_file* is used.

- 8) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 9) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 10) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_mask_3d` creates an output NetCDF file if the `-o=` argument is specified. This NetCDF dataset will contain the constructed mesh-mask (e.g., with values equal to 0 or 1) in a NetCDF variable called *output_mask_netcdf_variable* whose spatial dimensions are the same as those associated with the 2-D grid associated with the *longitude_variable* and *latitude_variable* specified with the `-vlon=` and `-vlat=` arguments (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the input 2-D grid) :

- 1) *output_mask_netcdf_variable*(*nlat*,*nlon*) : the constructed mesh-mask (e.g., with values equal to 0 or 1) on the 2-D grid-mesh associated with the input coordinate variables.

If the *longitude_variable* and *latitude_variable* in the *input_netcdf_file* are unidimensional, they are also copied in the output NetCDF file.

Optionally, if an *input_mask_netcdf_file* is specified with the `-fmsk=` argument and this NetCDF file contains the scale factors associated with the 2-D grid associated with the input *longitude_variable* and *latitude_variable* NetCDF variables, the output NetCDF file will also contain the following variables (if the `-noscalefac` argument is not specified) :

- 1) *netcdf_variable_e1grid_type*(*nlat*,*nlon*) : the first scale factor associated with the 2-D grid-mesh of the input coordinate variables.
- 2) *netcdf_variable_e2grid_type*(*nlat*,*nlon*) : the second scale factor associated with the 2-D grid-mesh of the input coordinate variables.

Multiplying the two scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the *output_mask_netcdf_variable*.

2.35 comp_mask_4d

2.35.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.35.2 Latest revision

30/04/2026

2.35.3 Purpose

Create a mesh-mask variable by specifying geographical bounds in indices or geographical units (e.g., degrees, meters or pressure) and/or by modifying an input mesh-mask readed from a NetCDF dataset and store the resulting mesh-mask variable in a NetCDF dataset if the `-o=` argument (described below) is specified.

If an output NetCDF dataset is not specified with the `-o=` argument, the geographical bounds of the selected domain are printed as indices on output of the procedure. Such indices can then be used for specifying the `-x=`, `-y=` and `-z=` arguments for the selected domain in other NCSTAT procedures such as use *comp_serie_4d*.

If you want to create a mesh-mask variable for a tridimensional NetCDF variable, use *comp_mask_3d* instead of *comp_mask_4d*.

2.35.4 Further Details

Usage

```
$ comp_mask_4d \
  -f=input_netcdf_file \
  -vlon=longitude_variable \
  -vlat=latitude_variable \
  -vlev=level_variable \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -z=level1,level2            (optional) \
  -a=type_of_coordinates      (optional : indices, units) \
  -imsk=in_mask_value         (optional : ident, 0, 1) \
  -omsk=out_mask_value        (optional : ident, 0, 1) \
  -fmsk=input_mask_netcdf_file (optional) \
  -vmsk=input_mask_netcdf_variable (optional) \
  -val=mask_value             (optional) \
  -rel=mask_relation          (optional : eq, gt, ge, lt, le) \
  -vout=output_mask_netcdf_variable (optional) \
  -o=output_mask_netcdf_file  (optional) \
  -noscalfac                  (optional) \
  -hdf5                       (optional) \
  -cdf5                        (optional)
```

By default

- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input *latitude_variable* and *longitude_variable* is assumed to be regular or Gaussian
- x=** the whole longitude range associated with the *longitude_variable*
- y=** the whole latitude range associated with the *latitude_variable*
- z=** the whole vertical range associated with the *level_variable*
- a=** the *type_of_coordinates* is set to *indices*. This means that the geographical bounds specified in the **-x=** and **-y=** arguments are given as indices not in geographical units (e.g., degrees of longitude and latitude or meters for example)
- imsk=** the *in_mask_value* is set to 1. This means that points inside the specified domain are selected
- omsk=** the *out_mask_value* is set to 0. This means that points outside the specified domain are not selected
- fmsk=** an *input_mask_netcdf_file* is not used
- vmsk=** an *input_mask_netcdf_variable* is not used
- val=** the *mask_value* is set to 1 .
- rel=** the *mask_relation* is set to *eq*
- vout=** the *output_mask_netcdf_variable* is named *grid_type/mask* if an output NetCDF file is created

-o= an *output_mask_netcdf_file* is not created

-noscalfac if the **-fmsk=** argument is present, the scale factors NetCDF variables, if they exist in the *input_mask_netcdf_file*, are automatically copied to the output NetCDF file. If the **-noscalfac** argument is specified, the scale factors NetCDF variables are not copied in the output NetCDF file

-hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

-cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

Remarks

- 1) The **-vlon=**, **-vlat=** and **-vlev=** arguments specify the coordinate variables from which the mesh-mask variable will be constructed and the **-f=input_netcdf_file** argument specifies that these coordinate NetCDF variables must be extracted from the NetCDF file *input_netcdf_file*. The longitude and latitude coordinate NetCDF variables may be one or two dimensional arrays.

- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain associated with the *longitude_variable* and *latitude_variable*, and the whole vertical resolution associated with the *level_variable* are used to construct the mesh-mask.

If **-a=indices** (the default value), the longitude (specified with the **-x=** argument), latitude (specified with the **-y=** argument) or level (specified with the **-z=** argument) range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

If **-a=units**, the longitude or latitude range must be a vector of two integers specifying the longitude and latitude limits of the domain in degrees of longitude or latitude, respectively. In that case, negative values are allowed for *lon1*, *lon2*, *lat1* and *lat2*. Moreover, the longitudes *lon1* and *lon2* are shifted to be in the interval between -180 and 180 degrees, assuming that the grid is periodic in longitude. The vertical range must be a vector of two integers specifying the first and last levels of the domain in the unit of the NetCDF variable given in the **-vlev=** argument.

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the input 3-D grid-mesh associated with the input coordinate variables is from an experiment with the ORCA model (R2, R4 or R05 resolutions).

This argument is also used to determine the name of the *output_mask_netcdf_variable* in the output NetCDF file if the **-vout=** argument is not specified. In that case, the *output_mask_netcdf_variable* will be named *grid_type//mask*.

- 4) The **-imsk=** argument specifies the value (0, 1 or *ident*) to be given to grid points inside of the specified domain (as specified by the **-x=**, **-y=** and **-z=** arguments). By default, values inside the specified domain are set to 1 (e.g., **-imsk=1**).

If **-imsk=ident**, the **-vmsk=** and **-fmsk=** arguments must also be given and the values inside the selected domain are read from the *input_mask_netcdf_variable* and are modified according to the logical relation determined by the **-rel=** and **-val=** arguments.

- 5) The **-omsk=** argument is identical to **-imsk=** argument, but concerns points outside the specified domain (as specified by the **-x=**, **-y=** and **-z=** arguments). By default, values outside the specified domain are set to 0 (**-omsk=0**).

If **-omsk=ident**, the **-vmsk=** and **-fmsk=** arguments must also be given and the values outside the selected domain are read from the *input_mask_netcdf_variable* and are modified according to the logical relation determined by the **-rel=** and **-val=** arguments.

- 6) If **-imsk=ident** or **-omsk=ident**, the **-fmsk=** and **-vmsk=** arguments must be specified. In that case, the output mesh-mask variable *output_mask_netcdf_variable* is first initialized from the *input_mask_netcdf_variable*

(as specified by the **-vmsk=** and **-fmsk=** arguments) inside the domain if **-imsk=ident** or outside the domain if **-omsk=ident**, and then modified in these regions according to the logical relationship:

- **output_mask**(i,j,k) = 1 if **input_mask**(i,j,k) *mask_relation* *mask_value* is true
- **output_mask**(i,j,k) = 0 otherwise

where *mask_relation* is determined from the **-rel=** argument and *mask_value* from the **-val=** argument.

By default, *mask_relation* is set to `eq` and *mask_value* is set to 1. . *mask_value* must be specified as a real number.

The **-fmsk=** argument must be present, if the **-vmsk=** argument is specified, otherwise the program will stop.

If the **-fmsk=** and **-vmsk=** arguments are not specified, the output mesh-mask variable is computed by applying the *in_mask_value* value (as specified by the **-imsk=** argument) inside the specified domain and the *out_mask_value* (as specified by the **-omsk=** argument) outside the specified domain.

- 7) The geographical shapes of the *input_mask_netcdf_variable* (in the *input_mask_netcdf_file*) must agree with the input coordinate variables *longitude_variable*, *latitude_variable* and *level_variable* (in the *input_netcdf_file*) if an *input_mask_netcdf_file* is used.
- 8) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 9) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 10) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_mask_4d` creates an output NetCDF file if the **-o=** argument is specified. This NetCDF dataset will contain the constructed mesh-mask (e.g., with values equal to 0 or 1) in a NetCDF variable called *output_mask_netcdf_variable* whose geographical dimensions are the same as those associated with the 3-D grid associated with the *longitude_variable*, *latitude_variable* and *level_variable* specified with the **-vlon=**, **-vlat=** and **-vlev=** arguments (in the description below, *nlat*, *nlon* and *nlev* are the lengths of the geographical dimensions of the input 3-D grid) :

- 1) *output_mask_netcdf_variable* (*nlev*, *nlat*, *nlon*) : the constructed mesh-mask (e.g., with values equal to 0 or 1) on the 3-D grid-mesh associated with the input coordinate variables.

If the *longitude_variable*, *latitude_variable* and *level_variable* in the *input_netcdf_file* are unidimensional, they are also copied in the output NetCDF file.

Optionally, if an *input_mask_netcdf_file* is specified with the **-fmsk=** argument and this NetCDF file contains the scale factors associated with the 3-D grid associated with the input *longitude_variable*, *latitude_variable* and *level_variable* NetCDF variables, the output NetCDF file will also contain the following variables (if the **-noscalfac** argument is not specified) :

- 1) *netcdf_variable_e1grid_type*(*nlat*, *nlon*) : the first scale factor associated with the 2-D grid-mesh of the input coordinate variables.
- 2) *netcdf_variable_e2grid_type*(*nlat*, *nlon*) : the second scale factor associated with the 2-D grid-mesh of the input coordinate variables.
- 3) *netcdf_variable_e3grid_type*(*nlev*, 1, 1) : the third scale factor associated with the 3-D grid-mesh of the input coordinate variables.

Multiplying the first two scale factors together gives the surface of each cell in the 3-D grid-mesh associated with the *output_mask_netcdf_variable*.

2.36 comp_norm_3d

2.36.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.36.2 Latest revision

30/04/2026

2.36.3 Purpose

Select, transform and normalize a multi-channel time series from a tridimensional NetCDF variable extracted from a NetCDF dataset.

The procedure allows a large variety of transformations on the input tridimensional NetCDF variable such as:

- 1) removing and applying *scale_factor* and *add_offset* attributes if they are present
- 2) changing the *missing_value* attribute (with the **-mi=** argument)
- 3) applying a given mesh-mask given in input of the procedure (with the **-m=** argument)
- 4) selecting only specific time steps in the output file (with the **-t=**, **-l=** and **-p=** arguments)
- 5) centering or standardizing the time series associated with selected cells of the 2-D grid-mesh associated with the input tridimensional NetCDF variable (with the **-a=** argument) with the help of an input climatology (specified with the **-c=** argument)
- 6) reducing the spatial dimensions of the output NetCDF dataset (with the **-compact** argument).

An output NetCDF file containing the transformed tridimensional variable is created.

If your data contains missing values use *comp_norm_miss_3d* instead of *comp_norm_3d* to transform your dataset.

Finally, if the NetCDF variable is fourdimensional use *comp_norm_4d* instead of *comp_norm_3d*.

2.36.4 Further Details

Usage

```

$ comp_norm_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file      (optional) \
-g=grid_type                        (optional : n, t, u, v, w, f) \
-r=resolution                        (optional : r2, r4) \
-b=nlon_orca, nlat_orca              (optional) \
-x=lon1,lon2                        (optional) \
-y=lat1,lat2                        (optional) \
-t=time1,time2                      (optional) \
-c=input_climatology_netcdf_file    (optional) \
-a=type_of_transformation            (optional : scp, cov, cor, spa, tim) \
-d=type_of_distance                  (optional : dist2, ident) \
-o=output_netcdf_file                (optional) \
-p=periodicity                      (optional) \
-l=selected_time_period              (optional) \
-cv=climatology_netcdf_variable     (optional) \
-mi=missing_value                   (optional) \
-double                             (optional) \
-bigfile                            (optional) \
-hdf5                               (optional) \
-cdf5                               (optional) \
-compact                            (optional) \
-tlimited                            (optional)

```

By default

- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_transformation* is set to *scp*. This means that the time series associated with the input *netcdf_variable* are output as raw data without any centering or standardization
- d=** the *type_of_distance* is set to *dist2*
- o=** the *output_netcdf_file* is named *norm_netcdf_variable.nc*
- p=** the *periodicity* is equal to the periodicity of the climatology if **-a=cov** or **-a=cor** or to *time2 - time1 + 1* if **-a=scp**
- l=** the whole time period as specified by the **-t=time1, time2** argument
- cv=** this argument, which specifies the name of the NetCDF variable used to compute the *input_climatology_netcdf_file*, has by default the same value as the **-v=** argument
- mi=** the *missing_value* in the output NetCDF file is set to *1.e+20*
- double** the data are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the data are stored as double-precision floating point numbers

- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- compact** the output NetCDF file is not compacted
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be transformed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* before transforming this tridimensional NetCDF variable. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series which must be written in the output NetCDF file. In other words, an *input_mesh_mask_netcdf_file* must be specified if the input *netcdf_variable* has masked values over all the land or ocean grid-points.

Note also that an *input_mesh_mask_netcdf_file* is required if **-a=spa** and **-d=dist2** are specified in order to get the scale factors of the 2-D grid-mesh associated with the input *netcdf_variable* as these scale factors are used to compute the weighted standard-deviation over the selected domain with this combination of input arguments.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_3d](#) or [comp_mask_3d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using [comp_norm_3d](#).

- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using [comp_norm_3d](#).

- 4) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model. In this case, the duplicate points from the NEMO grid are removed before the transformation, as far as possible, and, in particular, if the whole globe is used as the geographical domain. On output, the duplicate points are restored when writing the output file, if and only if, the whole globe is used as the geographical domain.

If **-g=** is set to *n*, it is assumed that the grid is regular or gaussian and has no duplicate points.

- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (i.e., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the used resolution. If

- **-r=r2** the input NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO model
 - **-r=r4** the input NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO model.
- 7) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the NEMO grid must be specified explicitly with the **-b=** argument.
- 8) The **-a=** argument specifies if the data must be centered or standardized with an input climatology (specified with the **-c=** argument):
- **-a=scp** means that the raw data are output
 - **-a=cov** means that the anomalies are output
 - **-a=cor** means that the standardized anomalies are output
 - **-a=spa** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the specified domain for each selected time step
 - **-a=tim** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the selected time steps for each grid-point.
- 9) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**.
- 10) If **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 11) The geographical shapes of the *netcdf_variable* (*input_netcdf_file*), the mask (*input_mesh_mask_netcdf_file*), the scale factors (*input_mesh_mask_netcdf_file*), and the climatology (*input_climatology_netcdf_file*) must agree.
- 12) The **-d=type_of_distance** argument is used only if **-a=spa** is specified. If:
- **-d=dist2**, the anomalies are standardized by a weighted standard-deviation. The sum of squares associated with a grid-point is weighted accordingly to the surface associated with that grid-point when computing the standard-deviation over the domain
 - **-d=ident** means that the anomalies are standardized by a simple arithmetic mean of the standard-deviations in the domain.
- 13) The **-l=** argument selects the indices of the time steps which must be included in the output NetCDF file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing). The argument list can be specified in two forms:
- **-l=n1,n2,...nn** allows to standardize and select for *n1*, *n2*, ... and *nn* time steps.
If *periodicity* is defined (with **-p=** option or if **-a=** is set to *cov*, *cor* or *spa*), *n1*, *n2*, ... *nn* time steps are selected for each period separately (see second example below)
 - **-l=n1:n2** allows to standardize and select time steps from *n1* to *n2* (or from *n1* to *n2* for each period separately, if *periodicity* is defined with **-p=** option or if **-a=** is set to *cov*, *cor* or *spa*).
- The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed.
- Be careful with time period limits when specifying the **-l=** argument.
- 14) If the **-p=** argument is specified and **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the periodicity deduced from the climatology (given by the **-c=** argument) overrides the **-p=** argument.
- The **-p=periodicity_of_data** argument is a strictly positive integer.

- 15) If the variable used to compute the input climatology has not the same name as the variable specified by the **-v=** argument, use the **-cv=** argument to specify the variable name for the climatology.
- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variable* (specified by the **-v=** argument) in the *output_netcdf_file*. *missing_value* must be a real number outside of the range of the *netcdf_variable*.

If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.

- 17) The **-double** argument specifies that the output NetCDF variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_netcdf_file*.
- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) If the **-compact** argument is specified and if a domain is selected (with the **-x=** and **-y=** arguments) then only data for the selected domain will be output. By default, the whole grid is stored (with missing values outside the selected domain).
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.
- 23) It is assumed that the data has no missing values, excepted those associated with a constant land-sea mask (if a mesh-mask NetCDF file is used). If it is the case, use *comp_norm_miss_3d* instead of *comp_norm_3d*.

Outputs

comp_norm_3d creates an output NetCDF file that contains the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file* and the transformed NetCDF variable. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) *netcdf_variable*(*ntime*, *nlat*, *nlon*) : the transformed NetCDF variable as specified by the **-m=**, **-a=** and **-mi=** arguments.

By default, the whole 2-D grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=** and **-y=** arguments (e.g., in this case *nlat*=*lat2-lat1*+1 and *nlon*=*lon2-lon1*+1). The number of time steps written in the output NetCDF file (e.g., *ntime*) is determined from the **-t=**, **-l=** and **-p=** arguments.

Examples

- 1) For computing time series of (monthly) anomalies from a NetCDF variable `sosstsst` stored in a file `ST7_1m_00101_20012_grid_T_sosstsst.nc`, apply a specific mask to the resulting time series and, finally, store the results in the NetCDF file `anoma_ST7_1m_00101_20012_grid_T_sosstsst.nc`, use the following command (note that the output file is compacted):

```
$ comp_norm_3d \  
-f=ST7_1m_00101_20012_grid_T_sosstsst.nc \  
-v=sosstsst \  
-g=t \  
-m=meshmask.indopacific.nc \  
-a=cov \  
-c=clim_ST7_1m_00101_20012_grid_T_sosstsst.nc \  
-o=anoma_ST7_1m_00101_20012_grid_T_sosstsst.nc \  
-compact
```

- 2) For selecting the first 120 days of each year (with a 365 days calendar) from the daily NetCDF file `ST7_1d_00101_20012_grid_T_sosstsst.nc`, which includes a NetCDF variable `sosstsst`, and store the results in the NetCDF file `select_ST7_1d_00101_20012_grid_T_sosstsst.nc`, use the following command :

```
$ comp_norm_3d \  
-f=ST7_1d_00101_20012_grid_T_sosstsst.nc \  
-v=sosstsst \  
-m=meshmask.orca2.nc \  
-g=t \  
-p=365 \  
-l=1:120 \  
-o=select_ST7_1d_00101_20012_grid_T_sosstsst.nc
```

2.37 comp_norm_4d

2.37.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.37.2 Latest revision

30/04/2026

2.37.3 Purpose

Select, transform and normalize a multi-channel time series from a fourdimensional NetCDF variable extracted from a NetCDF dataset.

The procedure allows a large variety of transformations on the input fourdimensional NetCDF variable such as:

- 1) removing and applying `scale_factor` and `add_offset` attributes if they are present
- 2) changing the `missing_value` attribute (with the **-mi=** argument)
- 3) applying a given mesh-mask given in input of the procedure (with the **-m=** argument)

- 4) selecting only specific time steps in the output file (with the **-t=**, **-l=** and **-p=** arguments)
- 5) centering or standardizing the time series associated with selected cells of the 3-D grid-mesh associated with the input tridimensional NetCDF variable (with the **-a=** argument) with the help of an input climatology (specified with the **-c=** argument)
- 6) reducing the spatial dimensions of the output NetCDF dataset (with the **-compact** argument).

An output NetCDF file containing the transformed fourdimensional variable is created.

Finally, if the NetCDF variable is tridimensional use *comp_norm_3d* instead of *comp_norm_4d*.

2.37.4 Further Details

Usage

```
$ comp_norm_4d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file      (optional) \
  -g=grid_type                        (optional : n, t, u, v, w, f) \
  -r=resolution                       (optional : r2, r4) \
  -b=nlon_orca, nlat_orca, nlevel_orca (optional) \
  -x=lon1,lon2                       (optional) \
  -y=lat1,lat2                       (optional) \
  -z=level1,level2                   (optional) \
  -t=time1,time2                     (optional) \
  -c=input_climatology_netcdf_file    (optional) \
  -a=type_of_transformation           (optional : scp, cov, cor, spa, tim) \
  -d=type_of_distance                 (optional : dist2, dist3, ident) \
  -o=output_netcdf_file               (optional) \
  -p=periodicity                     (optional) \
  -l=selected_time_period              (optional) \
  -cv=climatology_netcdf_variable     (optional) \
  -mi=missing_value                   (optional) \
  -double                             (optional) \
  -bigfile                             (optional) \
  -hdf5                                (optional) \
  -cdf5                                (optional) \
  -compact                             (optional) \
  -tlimited                             (optional)
```

By default

- g=** the *grid_type* is set to *n*, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*

- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_transformation* is set to `scp`. This means that the time series associated with the input *netcdf_variable* are output as raw data without any centering or standardization
- d=** the *type_of_distance* is set to `dist3`
- o=** the *output_netcdf_file* is named `norm_netcdf_variable.nc`
- p=** the *periodicity* is equal to the periodicity of the climatology if **-a=cov** or **-a=cor** or to `time2 - time1 + 1` if **-a=scp**
- l=** the whole time period as specified by the **-t=time1, time2** argument
- cv=** this argument, which specifies the name of the NetCDF variable used to compute the *input_climatology_netcdf_file*, has by default the same value as the **-v=** argument
- mi=** the *missing_value* in the output NetCDF file is set to `1.e+20`
- double** the data are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the data are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- compact** the output NetCDF file is not compacted
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be transformed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* before transforming this fourdimensional NetCDF variable. By default, it is assumed that each cell in the 3-D grid-mesh associated with the input fourdimensional NetCDF variable is a valid time series which must be written in the output NetCDF file. In other words, an *input_mesh_mask_netcdf_file* must be specified if the input *netcdf_variable* has masked values over the land or ocean grid-points.

Note also that an *input_mesh_mask_netcdf_file* is required if **-a=spa** and **-d=dist2** or **-d=dist3** are specified in order to get the scale factors of the 3-D grid-mesh associated with the input *netcdf_variable* as these scale factors are used to compute the weighted standard-deviation over the selected domain with this combination of input arguments.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_4d](#) or [comp_mask_4d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using `comp_norm_4d`.

- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_norm_4d*.

- 4) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model. In this case, the duplicate points from the NEMO grid are removed before the transformation, as far as possible, and, in particular, if the whole globe is used as the geographical domain. On output, the duplicate points are restored when writing the output file, if and only if, the whole globe is used as the geographical domain.

If **-g=** is set to *n*, it is assumed that the grid is regular or gaussian and has no duplicate points.

- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (i.e., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the used resolution. If
- **-r=r2** the input NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO model
 - **-r=r4** the input NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO model.
- 7) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the NEMO grid must be specified explicitly with the **-b=** argument.
- 8) The **-a=** argument specifies if the data must be centered or standardized with an input climatology (specified with the **-c=** argument):
- **-a=scp** means that the raw data are output
 - **-a=cov** means that the anomalies are output
 - **-a=cor** means that the standardized anomalies are output
 - **-a=spa** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the specified domain and vertical resolution for each selected time step
 - **-a=tim** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the selected time steps for each grid-point.
- 9) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**.
- 10) If **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 12) The **-d=type_of_distance** argument is used only if **-a=spa** is specified. If:

- **-d=dist3**, the anomalies are standardized by a weighted standard-deviation. The sum of squares associated with a grid-point is weighted accordingly to the mass (or volume) associated with that grid-point when computing the standard-deviation over the domain
 - **-d=dist2**, the anomalies are standardized by a weighted standard-deviation. The sum of squares associated with a grid-point is weighted accordingly to the surface associated with that grid-point when computing the standard-deviation over the domain
 - **-d=ident**, the anomalies are standardized by a simple arithmetic mean of the standard-deviations in the domain.
- 13) The **-l=** argument selects the indices of the time steps which must be included in the output NetCDF file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing). The argument list can be specified in two forms:
- **-l=n1,n2,...nn** allows to standardize and select for *n1*, *n2*, ... and *nn* time steps.
If *periodicity* is defined (with **-p=** option or if **-a=** is set to *cov*, *cor* or *spa*), *n1*, *n2*, ... *nn* time steps are selected for each period separately (see second example)
 - **-l=n1:n2** allows to standardize and select time steps from *n1* to *n2* (or from *n1* to *n2* for each period separately, if *periodicity* is defined with **-p=** option or if **-a=** is set to *cov*, *cor* or *spa*).
- The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed.
- Be careful with time period limits when specifying the **-l=** argument.
- 14) If the **-p=** argument is specified and **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the periodicity deduced from the climatology (given by the **-c=** argument) overrides the **-p=** argument.
The **-p=periodicity_of_data** argument is a strictly positive integer.
- 15) If the variable used to compute the input climatology has not the same name as the variable specified by the **-v=** argument, use the **-cv=** argument to specify the variable name for the climatology.
- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variable* (specified by the **-v=** argument) in the *output_netcdf_file*. *missing_value* must be a real number outside of the range of the *netcdf_variable*. If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.
- 17) The **-double** argument specifies that the output NetCDF variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_netcdf_file*.
- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.
If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.
If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) If the **-compact** argument is specified and a domain is selected (with the **-x=**, **-y=** and **-z=** arguments) then only data for the selected domain will be output. By default, the whole 3-D grid is stored (with missing values outside the selected domain).
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.
- 23) It is assumed that the data has no missing values, excepted those associated with a constant land-sea mask (if a mesh-mask NetCDF file is used).

Outputs

`comp_norm_4d` creates an output NetCDF file that contains the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file* and the transformed NetCDF variable. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, *nlev*, *nlat* and *nlon* are the lengths of the geographical dimensions of the input NetCDF variable) :

- 1) *netcdf_variable*(*ntime*, *nlev*, *nlat*, *nlon*) : the transformed NetCDF variable as specified by the **-m=**, **-a=** and **-mi=** arguments.

By default, the whole 3-D grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used, the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=**, **-y=** and **-z=** arguments (e.g., in this case *nlev*=*level2-level1*+1, *nlat*=*lat2-lat1*+1 and *nlon*=*lon2-lon1*+1). The number of time steps written in the output NetCDF file (e.g., *ntime*) is determined from the **-t=**, **-l=** and **-p=** arguments.

Examples

- 1) For computing time series of (monthly) anomalies from a NetCDF variable *votemper* stored in a file *ST7_1m_00101_20012_grid_T_votemper.nc*, apply a specific mask to the resulting time series and, finally, store the results in the NetCDF file *anoma_ST7_1m_00101_20012_grid_T_votemper.nc*, use the following command (note that the output file is compacted):

```
$ comp_norm_4d \
-f=ST7_1m_00101_20012_grid_T_votemper.nc \
-v=votemper \
-g=t \
-m=meshmask.indopacific.nc \
-a=cov \
-c=clim_ST7_1m_00101_20012_grid_T_votemper.nc \
-o=anoma_ST7_1m_00101_20012_grid_T_votemper.nc \
-compact
```

- 2) For selecting the first 120 days of each year (with a 365 days calendar) from the daily NetCDF file *ST7_1d_00101_20012_grid_T_votemper.nc*, which includes a NetCDF variable *votemper*, and store the results in the NetCDF file *select_ST7_1d_00101_20012_grid_T_votemper.nc*, use the following command :

```
$ comp_norm_4d \
-f=ST7_1d_00101_20012_grid_T_votemper.nc \
-v=votemper \
```

(continues on next page)

(continued from previous page)

```
-m=meshmask.orca2.nc \
-g=t \
-p=365 \
-l=1:120 \
-o=select_ST7_1d_00101_20012_grid_T_votemper.nc
```

2.38 comp_norm_miss_3d

2.38.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.38.2 Latest revision

30/04/2026

2.38.3 Purpose

Select, transform and normalize a multi-channel time series from a tridimensional NetCDF variable with missing values extracted from a NetCDF dataset.

The procedure allows a large variety of transformation on the input tridimensional NetCDF variable such as:

- 1) removing and applying scale_factor and add_offset attributes if they are present
- 2) changing the missing_value attribute (with the **-mi=** argument)
- 3) applying a given mesh-mask given in input of the procedure (with the **-m=** argument)
- 4) selecting only specific time steps in the output file (with the **-t=**, **-l=** and **-p=** arguments)
- 5) centering or standardizing the time series associated with selected cells of the 2-D grid-mesh associated with the input tridimensional NetCDF variable (with the **-a=** argument) with the help of an input climatology (specified with the **-c=** argument)
- 6) reducing the spatial dimensions of the output NetCDF dataset (with the **-compact** argument)

An output NetCDF file containing the transformed tridimensional variable is created.

If your data does not contain missing values use [comp_norm_3d](#) instead of `comp_norm_miss_3d` to transform your dataset.

2.38.4 Further Details

Usage

```
$ comp_norm_miss_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file      (optional) \
-g=grid_type                        (optional : n, t, u, v, w, f) \
-r=resolution                        (optional : r2, r4) \
```

(continues on next page)

(continued from previous page)

-b=nlon_orca, nlat_orca	(optional) \
-x=lon1, lon2	(optional) \
-y=lat1, lat2	(optional) \
-t=time1, time2	(optional) \
-c=input_climatology_netcdf_file	(optional) \
-a=type_of_transformation	(optional : scp, cov, cor, spa, tim) \
-d=type_of_distance	(optional : dist2, ident) \
-o=output_netcdf_file	(optional) \
-p=periodicity	(optional) \
-l=selected_time_period	(optional) \
-cv=climatology_netcdf_variable	(optional) \
-mi=missing_value	(optional) \
-double	(optional) \
-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-compact	(optional) \
-tlimited	(optional)

By default

- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_transformation* is set to *scp*. This means that the time series associated with the input *netcdf_variable* are output as raw data without any centering or standardization
- d=** the *type_of_distance* is set to *dist2*
- o=** the *output_netcdf_file* is named *norm_netcdf_variable.nc*
- p=** the *periodicity* is equal to the periodicity of the climatology if **-a=cov** or **-a=cor** or to *time2 - time1 + 1* if **-a=scp**
- l=** the whole time period as specified by the **-t=time1, time2** argument
- cv=** this argument, which specifies the name of the NetCDF variable used to compute the *input_climatology_netcdf_file*, has by default the same value as the **-v=** argument
- mi=** the *missing_value* in the output NetCDF file is set to *1.e+20*
- double** the data are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the data are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file

- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- compact** the output NetCDF file is not compacted
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be transformed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* before transforming this tridimensional NetCDF variable. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series which must be written in the output NetCDF file. In other words, an *input_mesh_mask_netcdf_file* must be specified if the input *netcdf_variable* has masked values over all the land or ocean grid-points.

Note also that an *input_mesh_mask_netcdf_file* is required if **-a=spa** and **-d=dist2** are specified in order to get the scale factors of the 2-D grid-mesh associated with the input *netcdf_variable* as these scale factors are used to compute the weighted standard-deviation over the selected domain with this combination of input arguments.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to *comp_clim_miss_3d* or *comp_mask_3d* for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_norm_miss_3d*.

- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices before using *comp_norm_miss_3d*.

- 4) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model. In this case, the duplicate points from the NEMO grid are removed before the transformation, as far as possible, and, in particular, if the whole globe is used as the geographical domain. On output, the duplicate points are restored when writing the output file, if and only if, the whole globe is used as the geographical domain.
If **-g=** is set to *n*, it is assumed that the grid is regular or gaussian and has no duplicate points.
- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (i.e., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the used resolution. If

- **-r=r2** the input NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO model

- **-r=r4** the input NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO model.
- 7) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the NEMO grid must be specified explicitly with the **-b=** argument.
 - 8) The **-a=** argument specifies if the data must be centered or standardized with an input climatology (specified with the **-c=** argument):
 - **-a=scp** means that the raw data are output
 - **-a=cov** means that the anomalies are output
 - **-a=cor** means that the standardized anomalies are output
 - **-a=spa** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the specified domain for each selected time step
 - **-a=tim** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the selected time steps for each grid-point.
 - 9) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**.
 - 10) If **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
 - 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
 - 12) The **-d=type_of_distance** argument is used only if **-a=spa** is specified. If:
 - **-d=dist2**, the anomalies are standardized by a weighted standard-deviation. The sum of squares associated with a grid-point is weighted accordingly to the surface associated with that grid-point when computing the standard-deviation over the domain
 - **-d=ident**, the anomalies are standardized by a simple arithmetic mean of the standard-deviations in the domain.
 - 13) The **-l=** argument selects the indices of the time steps which must be included in the output NetCDF file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing). The argument list can be specified in two forms:
 - **-l=n1,n2,...nn** allows to standardize and select for *n1*, *n2*, ... and *nn* time steps.
If *periodicity* is defined (with **-p=** option or if **-a=** is set to *cov*, *cor* or *spa*), *n1*, *n2*, ... *nn* time steps are selected for each period separately (see second example).
 - **-l=n1:n2** allows to standardize and select time steps from *n1* to *n2* (or from *n1* to *n2* for each period separately, if *periodicity* is defined with **-p=** option or if **-a=** is set to *cov*, *cor* or *spa*).

The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed.

Be careful with time period limits when specifying the **-l=** argument.
 - 14) If the **-p=** argument is specified and **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the periodicity deduced from the climatology (given by the **-c=** argument) overrides the **-p=** argument.
The **-p=periodicity_of_data** argument is a strictly positive integer.
 - 15) If the variable used to compute the input climatology has not the same name as the variable specified by the **-v=** argument, use the **-cv=** argument to specify the variable name for the climatology.

- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variable* (specified by the **-v=** argument) in the *output_netcdf_file*. *missing_value* must be a real number outside of the range of the *netcdf_variable*.

If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.

- 17) The **-double** argument specifies that the output NetCDF variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_netcdf_file*.
- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) If the **-compact** argument is specified and if a domain is selected (with the **-x=** and **-y=** arguments) then only data for the selected domain will be output. By default, the whole grid is stored (with missing values outside the selected domain).
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.
- 23) It is assumed that the specified *netcdf_variable* has a scalar *missing_value* or `_FillValue` attribute and that missing values in the data are identified by the value of this *missing_value* or `_FillValue` attribute.

Outputs

`comp_norm_miss_3d` creates an output NetCDF file that contains the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file* and the transformed NetCDF variable. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable)

- 1) *netcdf_variable*(`ntime`, `nlat`, `nlon`) : the transformed NetCDF variable as specified by the **-m=**, **-a=** and **-mi=** arguments.

By default, the whole 2-D grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=** and **-y=** arguments (e.g., in this case `nlat=lat2-lat1+1` and `nlon=lon2-lon1+1`). The number of time steps written in the output NetCDF file (e.g., `ntime` is determined from the **-t=**, **-l=** and **-p=** arguments.

Examples

- 1) For computing time series of (monthly) anomalies from a NetCDF variable `sst` with missing values extracted from a file named `Hadsst2_1m_190001_200512_sst.nc` and store the results in the NetCDF file `anoma_Hadsst2_1m_190001_200512_sst.nc`, use the following command :

```
$ comp_norm_miss_3d \
-f=Hadsst2_1m_190001_200512_sst.nc \
-v=sst \
-a=cov \
-c=clim_ST7_1m_00101_20012_grid_T_sosstsst.nc \
-o=anoma_Hadsst2_1m_190001_200512_sst.nc
```

2.39 comp_norm_miss_4d

2.39.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.39.2 Latest revision

30/04/2026

2.39.3 Purpose

Select, transform and normalize a multi-channel time series from a fourdimensional NetCDF variable with missing extracted from a NetCDF dataset.

The procedure allows a large variety of transformations on the input fourdimensional NetCDF variable such as:

- 1) removing and applying `scale_factor` and `add_offset` attributes if they are present
- 2) changing the `missing_value` attribute (with the **-mi=** argument)
- 3) applying a given mesh-mask given in input of the procedure (with the **-m=** argument)
- 4) selecting only specific time steps in the output file (with the **-t=**, **-l=** and **-p=** arguments)
- 5) centering or standardizing the time series associated with selected cells of the 3-D grid-mesh associated with the input tridimensional NetCDF variable (with the **-a=** argument) with the help of an input climatology (specified with the **-c=** argument)
- 6) reducing the spatial dimensions of the output NetCDF dataset (with the **-compact** argument).

An output NetCDF file containing the transformed fourdimensional variable is created.

If your data does not contain missing values use [comp_norm_4d](#) instead of `comp_norm_miss_4d` to transform your dataset.

Finally, if the NetCDF variable is tridimensional use [comp_norm_miss_3d](#) instead of `comp_norm_miss_4d`.

2.39.4 Further Details

Usage

```
$ comp_norm_miss_4d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file      (optional) \
-g=grid_type                        (optional : n, t, u, v, w, f) \
-r=resolution                        (optional : r2, r4) \
-b=nlon_orca, nlat_orca, nlevel_orca (optional) \
-x=lon1,lon2                        (optional) \
-y=lat1,lat2                        (optional) \
-z=level1,level2                    (optional) \
-t=time1,time2                      (optional) \
-c=input_climatology_netcdf_file    (optional) \
-a=type_of_transformation            (optional : scp, cov, cor, spa, tim) \
-d=type_of_distance                  (optional : dist2, dist3, ident) \
-o=output_netcdf_file                (optional) \
-p=periodicity                       (optional) \
-l=selected_time_period              (optional) \
-cv=climatology_netcdf_variable      (optional) \
-mi=missing_value                    (optional) \
-double                             (optional) \
-bigfile                             (optional) \
-hdf5                                (optional) \
-cdf5                                (optional) \
-compact                             (optional) \
-tlimited                             (optional)
```

By default

- g=** the *grid_type* is set to *n*, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_transformation* is set to *scp*. This means that the time series associated with the input *netcdf_variable* are output as raw data without any centering or standardization
- d=** the *type_of_distance* is set to *dist3*
- o=** the *output_netcdf_file* is named *norm_netcdf_variable.nc*
- p=** the *periodicity* is equal to the periodicity of the climatology if **-a=cov** or **-a=cor** or to *time2 - time1 + 1* if **-a=scp**

- l=** the whole time period as specified by the **-t=time1, time2** argument
- cv=** this argument, which specifies the name of the NetCDF variable used to compute the *input_climatology_netcdf_file*, has by default the same value as the **-v=** argument
- mi=** the *missing_value* in the output NetCDF file is set to $1.e+20$
- double** the data are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the data are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- compact** the output NetCDF file is not compacted
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be transformed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* before transforming this fourdimensional NetCDF variable. By default, it is assumed that each cell in the 3-D grid-mesh associated with the input fourdimensional NetCDF variable is a valid time series which must be written in the output NetCDF file. In other words, an *input_mesh_mask_netcdf_file* must be specified if the input *netcdf_variable* has masked values over the land or ocean grid-points.

Note also that an *input_mesh_mask_netcdf_file* is required if **-a=spa** and **-d=dist2** or **-d=dist3** are specified in order to get the scale factors of the 3-D grid-mesh associated with the input *netcdf_variable* as these scale factors are used to compute the weighted standard-deviation over the selected domain with this combination of input arguments.

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to *comp_clim_miss_4d* or *comp_mask_4d* for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_norm_miss_4d*.

- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_4d* for transforming geographical coordinates as indices before using *comp_norm_miss_4d*.

- 4) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the NetCDF variable is from an experiment with the NEMO model. In this case, the duplicate points from the NEMO grid are removed before the transformation, as far as possible, and, in particular, if the whole globe is used as the geographical domain. On output, the duplicate points are restored when writing the output file, if and only if, the whole globe is used as the geographical domain.
- If **-g=** is set to `n`, it is assumed that the grid is regular or gaussian and has no duplicate points.
- 6) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (i.e., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the used resolution. If
- **-r=r2** the input NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO model
 - **-r=r4** the input NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO model.
- 7) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the NEMO grid must be specified explicitly with the **-b=** argument.
- 8) The **-a=** argument specifies if the data must be centered or standardized with an input climatology (specified with the **-c=** argument):
- **-a=scp** means that the raw data are output
 - **-a=cov** means that the anomalies are output
 - **-a=cor** means that the standardized anomalies are output
 - **-a=spa** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the specified domain and vertical resolution for each selected time step
 - **-a=tim** means that the standardized anomalies are output, but the anomalies are standardized by the standard-deviation averaged over the selected time steps for each grid-point.
- 9) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**.
- 10) If **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 12) The **-d=type_of_distance** argument is used only if **-a=spa** is specified. If:
- **-d=dist3**, the anomalies are standardized by a weighted standard-deviation. The sum of squares associated with a grid-point is weighted accordingly to the mass (or volume) associated with that grid-point when computing the standard-deviation over the domain
 - **-d=dist2**, the anomalies are standardized by a weighted standard-deviation. The sum of squares associated with a grid-point is weighted accordingly to the surface associated with that grid-point when computing the standard-deviation over the domain
 - **-d=ident**, the anomalies are standardized by a simple arithmetic mean of the standard-deviations in the domain.
- 13) The **-l=** argument selects the indices of the time steps which must be included in the output NetCDF file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing). The argument list can be specified in two forms:

- **-l=*n1,n2,...nn*** allows to standardize and select for *n1*, *n2*, ... and *nn* time steps.

If *periodicity* is defined (with **-p=** option or if **-a=** is set to `cov`, `cor` or `spa`), *n1*, *n2*, ... *nn* time steps are selected for each period separately (see second example)

- **-l=*n1:n2*** allows to standardize and select time steps from *n1* to *n2* (or from *n1* to *n2* for each period separately, if *periodicity* is defined with **-p=** option or if **-a=** is set to `cov`, `cor` or `spa`).

The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed.

Be careful with time period limits when specifying the **-l=** argument.

- 14) If the **-p=** argument is specified and **-a=cov**, **-a=cor**, **-a=spa** or **-a=tim**, the periodicity deduced from the climatology (given by the **-c=** argument) overrides the **-p=** argument.

The **-p=periodicity_of_data** argument is a strictly positive integer.

- 15) If the variable used to compute the input climatology has not the same name as the variable specified by the **-v=** argument, use the **-cv=** argument to specify the variable name for the climatology.
- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variable* (specified by the **-v=** argument) in the *output_netcdf_file*. *missing_value* must be a real number outside of the range of the *netcdf_variable*.

If the **-mi=** argument is not specified *missing_value* is set to `1.e+20`.

- 17) The **-double** argument specifies that the output NetCDF variable must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_netcdf_file*.
- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) If the **-compact** argument is specified and a domain is selected (with the **-x=**, **-y=** and **-z=** arguments) then only data for the selected domain will be output. By default, the whole 3-D grid is stored (with missing values outside the selected domain).
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.
- 23) It is assumed that the specified *netcdf_variable* has a scalar *missing_value* or `_FillValue` attribute and that missing values in the data are identified by the value of this *missing_value* or `_FillValue` attribute.

Outputs

`comp_norm_miss_4d` creates an output NetCDF file that contains the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file* and the transformed NetCDF variable. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable) :

- 1) *netcdf_variable*(`ntime`, `nlev`, `nlat`, `nlon`) : the transformed NetCDF variable as specified by the **-m=**, **-a=** and **-mi=** arguments.

By default, the whole 3-D grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used, the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=**, **-y=** and **-z=** arguments (e.g., in this case `nlev=level2-level1+1`, `nlat=lat2-lat1+1` and `nlon=lon2-lon1+1`). The number of time steps written in the output NetCDF file (e.g., `ntime`) is determined from the **-t=**, **-l=** and **-p=** arguments.

Examples

- 1) For computing time series of (monthly) anomalies from a NetCDF variable `votemper` with missing values stored in a file `ST7_1m_00101_20012_grid_T_votemper.nc`, apply a specific mask to the resulting time series and, finally, store the results in the NetCDF file `anoma_ST7_1m_00101_20012_grid_T_votemper.nc`, use the following command (note that the output file is compacted):

```
$ comp_norm_miss_4d \  
-f=ST7_1m_00101_20012_grid_T_votemper.nc \  
-v=votemper \  
-g=t \  
-m=meshmask.indopacific.nc \  
-a=cov \  
-c=clim_ST7_1m_00101_20012_grid_T_votemper.nc \  
-o=anoma_ST7_1m_00101_20012_grid_T_votemper.nc \  
-compact
```

- 2) For selecting the first 120 days of each year (with a 365 days calendar) from the daily NetCDF file `ST7_1d_00101_20012_grid_T_votemper.nc`, which includes a NetCDF variable `votemper`, and store the results in the NetCDF file `select_ST7_1d_00101_20012_grid_T_votemper.nc`, use the following command :

```
$ comp_norm_miss_4d \  
-f=ST7_1d_00101_20012_grid_T_votemper.nc \  
-v=votemper \  
-m=meshmask.orca2.nc \  
-g=t \  
-p=365 \  
-l=1:120 \  
-o=select_ST7_1d_00101_20012_grid_T_votemper.nc
```

2.40 comp_ortho_rot_eof_3d

2.40.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.40.2 Latest revision

30/04/2026

2.40.3 Purpose

Performs an orthogonal rotation of a (partial) Empirical Orthogonal Function (EOF) or Principal Component Analysis (PCA) model (e.g., a factor loading matrix and the associated standardized amplitude time series) computed from a previous EOF or PCA analysis of a tridimensional NetCDF variable. The factor loading matrix (e.g., the associated scaled EOF patterns corresponding to the selected rotated PC time series) are also rotated and the new rotated EOF or PCA model is stored in a NetCDF file.

Spatial EOF patterns (e.g., scaled EOFs) and Principal Component (PC) time series derived from EOF analysis or PCA are constrained to be both spatially and temporally orthogonal to one another so that they maximize the variance they described over the entire analysis domain and selected time period [Jolliffe] [vonStorch_Zwiers] . However, the spatial orthogonality of the EOFs is a strong and undesirable constraint imposed on the EOFs and, as a result, EOFs are subject to problems such as domain dependence and inaccurate representation of physical relationships embedded in the data [Jolliffe] [Richman] .

Orthogonal rotation of selected standardized PC time series can be used to obtain spatial and time patterns (e.g., rotated scaled EOFs and PC time series) that are hopefully more physically meaningful by relaxing the spatial orthogonality of the rotated scaled EOFs, while preserving the temporal orthogonality of the rotated standardized PC time series. Preserving the orthogonality of the (rotated) PC time series implies that these rotated PCs describe the same amount of total variance as the original (unrotated) PC time series, which is a very useful statistical property, but this variance is more equally (or differently) distributed among the rotated PC time series [Jolliffe] [Richman] .

More precisely, the orthogonal rotation in comp_ortho_rot_eof_3d has the objective to make the rotated standardized PC time series as simple as possible to interpret by attempting to constrain the spatial coefficients (or loadings) associated with each of them to be large or very small in absolute magnitude, but avoiding as much as possible spatial loadings of moderate size [Jolliffe] [Jackson] .

The orthogonal rotation matrix, which will perform this linear transformation is obtained by maximizing a generalized orthomax criterion, including quartimax, varimax and equamax rotation methods [Jackson] at the user option (see description of the **-w=** argument below for more details about the selection of the rotation method which will be used). This generalized orthomax criterion is described in details in [Clarkson_Jennrich] [Jackson] and amounts to obtain rotated standardized PC time series which are more spatially localized by relaxing the spatial orthogonality of the original EOF (or PCA) modes, while preserving the orthogonality of the (rotated) standardized PC time series as described above.

The **-w=rotation_factor** argument determines the kind of orthogonal rotation to be computed [Clarkson_Jennrich] [Jackson] and may be set as follows :

- **-w=0** . is the quartimax method [Jackson] , which attempts to get each variable (e.g., grid-point) to load highly on only one (or a few) EOFs or factors.
- **-w=1** . is the varimax method [Jackson] , which attempts to load highly a relatively low number of variables on each EOF or factor. Varimax is the most widely used method of orthogonal rotation.

- **-w=nrot/2.**, where `nrot` is the number of rotated PCs, is the equamax method [Jackson], which is a compromise of the above two.

More generally, the value of the **-w=** argument can be any positive real number, but best values lie in the closed interval $[0., 5. * \text{nrot}]$. Generally, the larger the value is, the more equal is the dispersion of the variance accounted for across the rotated EOFs or factors.

As an illustration, by default, `comp_ortho_rot_eof_3d` uses the varimax rotation method, which simplifies the spatial structure of the rotated (scaled) EOFs by maximizing the sum of the variances of the squared columns loadings of the selected rotated (scaled) EOFs [Jolliffe] [Jackson] [vonStorch_Zwiers]. As a result, the rotated EOFs tend to have coefficients biased towards either large or zero values. Hence these rotated EOFs should be easier to interpret than their unrotated counterparts.

Note that, other methods of rotation exist to obtain more physically based EOFs and PC time series. For example, the orthogonal rotation matrix can be designed to maximize the energy in a selected frequency band for the rotated PC time series, see `comp_filt_rot_eof_3d` for more details.

Using as input an EOF NetCDF file produced by `comp_eof_3d` or `comp_eof_miss_3d`, `comp_ortho_rot_eof_3d` performs an orthogonal rotation of selected standardized PC time series readed from this input EOF NetCDF file and computes the regression spatial patterns associated with these rotated standardized PC time series (e.g., the orthogonal projection onto the orthonormal set formed by the new rotated standardized PC time series). The raw variances and percentages of variance described by the new rotated standardized PC time series are also computed.

The selected standardized PC time series, which will be rotated is determined on entry by the value of the **-se=** argument, which is a list of PC (or EOF) numbers.

Optionally, the rotated PC time series can be estimated with the help of a metric such that the results are weighted by the surface (or volume) associated with each cell in the grid associated with the input tridimensional NetCDF variable (which contains the original scaled EOFs) so that equal areas (or volumes) carry equal weights in the results of the rotated EOF analysis (see the **-d=** argument description).

An output NetCDF dataset containing these rotated PC standardized time series, their associated regression spatial patterns and the explained variance statistics is created.

If the NetCDF variable used in the original EOF analysis is fourdimensional use `comp_ortho_rot_eof_4d` instead of `comp_ortho_rot_eof_3d`.

2.40.4 Further Details

Usage

```
$ comp_ortho_rot_eof_3d\
-f=eof_netcdf_file \
-v=eof_netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-se=selected_eofs           (optional) \
-g=grid_type                (optional : n, t, u, v, w, f) \
-r=resolution               (optional : r2, r4) \
-b=nlon_orca, nlat_orca     (optional) \
-x=lon1,lon2                (optional) \
-y=lat1,lat2                (optional) \
-t=time1,time2              (optional) \
-d=distance                 (optional : dist2, ident) \
-w=rotation_factor          (optional : 0. > +inf) \
-o=output_rot_eof_netcdf_file (optional) \
-mi=missing_value           (optional) \
-knorm                      (optional) \
```

(continues on next page)

(continued from previous page)

-double	(optional) \
-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- se=** all the PC time series stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF EOF variable, *eof_netcdf_variable*, is assumed to be regular or Gaussian
- r=** if the input *eof_netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *eof_netcdf_variable*
- y=** the whole latitude domain associated with the *eof_netcdf_variable*
- t=** the whole time period associated with the input PC time series
- d=** the *distance* is set to the distance used in the original EOF analysis.
- w=** the *rotation_factor* is set to 0., which means that a varimax orthogonal rotation is performed.
- o=** the *output_rot_eof_netcdf_file* is named *ortho_rot_eof_netcdf_variable.nc*
- mi=** the *missing_value* attribute for the rotated EOFs in the output NetCDF file is set to 1.e+20
- knorm** the orthogonal rotation of the standardized PC time series is performed without Kaiser's row normalization. If **-knorm** is activated, Kaiser's row normalization is performed before the orthogonal rotation of the PC time series and the spatial loadings (e.g., the rotated EOFs) associated with the rotated PC time series are rescaled after the orthogonal rotation
- double** the results of the rotated EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=***eof_netcdf_variable* argument specifies the NetCDF variable for which an EOF analysis was originally computed by *comp_eof_3d* (or *comp_eof_miss_3d*) and the **-f=***input_eof_netcdf_file* argument specifies that the

resulting EOF patterns and PC time series must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_3d* or *comp_eof_miss_3d*.

These EOF patterns and PC time series are used to compute the orthogonal rotation matrix which will be applied to the selected EOF model.

- 2) The **-m=input_mesh_mask_netcdf_file** argument specifies the land-sea mask to apply to the *eof_netcdf_variable* (which contains the scaled eigenvectors of the EOF analysis) for transforming this tridimensional NetCDF variable as a rectangular matrix before performing the rotated EOF analysis. It is assumed that this land-sea mask is exactly the same as the one used in the original EOF analysis, whose results are stored in the input NetCDF file *eof_netcdf_file*.

The scale factors associated with the 2-D grid-mesh of the input NetCDF variable (needed if **-d=dist2** is specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.

- 3) The **-se=selected_eofs** argument allows the user to select the standardized PC time series which must be included in the orthogonal rotation. The list of selected PCs may be given in two formats:

- **-se=1,3,...,nn** allows to include *pc1*, *pc3*,... and *pcnn* in the orthogonal rotation
- **-se=1 : 4** allows to include from *pc1* to *pc4* in the orthogonal rotation.

The two forms of the **-se=** argument may be combined and repeated any number of times. Duplicate PC numbers are not allowed. If the **-se=** argument is not specified, all the PC time series stored in the *input_eof_netcdf_file* are used in the orthogonal rotation.

- 4) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model. In this case, the duplicate points from the ORCA grid are removed before the rotated EOF analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typemask*, *e1grid_type* and *e2grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_3d* if the 2-D grid-mesh is regular or gaussian.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:

- **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
- **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.

- 6) If the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

- 7) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are not specified, the geographical domain used in the rotated EOF analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_ortho_rot_eof_3d*.

- 8) If the **-t=***time1,time2* argument is missing, the whole time period associated with the original PC time series is used in the orthogonal rotation.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` observations.

- 9) The geographical shapes of the *eof_netcdf_variable* (in the *eof_netcdf_file*), the land-sea mask and scale factors (in the *input_mesh_mask_netcdf_file*) must agree.
- 10) The **-d=** argument specifies the metric and scalar product used in the rotated EOF analysis:
 - **-d=dist2** means that the orthogonal rotation is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid-point is weighted accordingly to the surface associated with it)
 - **-d=ident** means that the orthogonal rotation is done with the identity metric, e.g., the usual Euclidean distance and scalar product are used in the rotated EOF analysis.

By default, the **-d=** argument is set to the distance used in the original EOF analysis.

- 11) The **-w=***rotation_factor* argument specifies the exact form of the function criterion, which must be maximized over the set of `nrot` by `nrot` orthogonal rotation matrices (where `nrot` is the number of selected PCs as specified with the **-se=** argument). *rotation_factor* is a positive real scalar and may be set as follows:
 - **-w=0.** is the quartimax method, which attempts to get each variable (e.g., grid-point) to load highly on only one (or a few) EOFs or factors.
 - **-w=1.** is the varimax method, which attempts to load highly a relatively low number of variables on each EOF or factor. Varimax is the most widely used method of orthogonal rotation.
 - **-w=nrot/2.**, where `nrot` is the number of rotated PCs, is the equamax method, which is a compromise of the above two.

More generally, *rotation_factor* can be any positive real number, but best values lie in the closed interval $[0., 5. * nrot]$. Generally, the larger the value of *rotation_factor* is, the more equal is the dispersion of the variance accounted for across the rotated EOFs or factors.

- 12) The **-mi=***missing_value* argument specifies the missing value indicator associated with the NetCDF variables in the *output_rot_eof_netcdf_file*.

If the **-mi=** argument is not specified *missing_value* is set to `1.e+20`.

- 13) The **-knorm** argument specifies that orthogonal rotation of the original PC time series must be performed using Kaiser's row normalization of the associated spatial loadings (e.g., the original scaled EOFs) and the computed spatial loadings (e.g., the rotated EOFs) associated with the rotated PC time series are rescaled after the orthogonal rotation.

- 14) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file, *output_rot_eof_netcdf_file*.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 15) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 18) It is assumed that the original EOF variable has no missing values excepted those associated with a constant land-sea mask.

- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 20) For more details on EOF analysis, PCA and orthogonal rotation methods, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
- “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 487 pp., 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
- “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
- “Quartic rotation criteria and algorithms” by Clarkson, D.B., and Jennrich, R.I., Psychometrika, Vol. 53, 251-259, 1988. <https://link.springer.com/article/10.1007/BF02294136>
- “Rotation of principal components” by Richman, M.B., International Journal of Climatology, Vol. 6, 293-335, 1986. <https://doi.org/10.1002/joc.3370060305>

Outputs

`comp_ortho_rot_eof_3d` creates an output NetCDF file that contains the rotated standardized PC time series, their associated rotated EOF patterns and the variances described by the computed rotated PC time series.

The number of rotated PCs and EOFs, `nrot`, stored in the output NetCDF dataset is determined by the **-se=selected_eofs** argument. The number of observations in the output NetCDF dataset is determined from the **-t=time1,time2** argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable *eof_netcdf_variable* and `nrot` is the number of selected PC time series, which have been rotated as determined by the list specified with the **-se=selected_eofs** argument) :

- 1) `pc_number(nrot)` : the list of rotated PC time series as specified with the **-se=selected_eofs** argument.

- 2) `eof_netcdf_variable_ortho_rot_eof(nrot, nlat, nlon)` : the rotated EOF patterns. These rotated EOFs are scaled such that they give the scalar products (if `-a=scp` in the original EOF analysis), covariances (`-a=cov` in the original EOF analysis) or correlations (`-a=cor` in the original EOF analysis) between the original observed variables and the rotated PC time series.

The rotated EOF patterns are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `eof_netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, this output NetCDF variable is filled with missing values.

- 3) `eof_netcdf_variable_ortho_rot_pc(ntime, nrot)` : the rotated standardized PC time series.

The rotated PC time series are always standardized to unit variance.

- 4) `eof_netcdf_variable_ortho_rot_std(nrot)` : the standard-deviations of the rotated PC time series.

The squares of these statistics are equal to the raw variance described by the rotated PC time series.

- 5) `eof_netcdf_variable_ortho_rot_var(nrot)` : the proportion of variance explained by each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

Examples

- 1) For computing a varimax rotated 10-EOF analysis from a previous EOF analysis (stored in the file `eof_HadISST1_1m_197902_200501_sst_oi.nc`) of a tridimensional NetCDF variable named `sst` and store the results of the rotated EOF analysis in a NetCDF file named `rot_eof_HadISST1_1m_197902_200501_sst_oi.nc`, use the following command :

```
$ comp_ortho_rot_eof_3d \
-f=eof_HadISST1_1m_197902_200501_sst_oi.nc \
-v=sst \
-se=1:10 \
-m=HadISST1_mask.nc \
-o=rot_eof_HadISST1_1m_197902_200501_sst_oi.nc
```

2.41 comp_ortho_rot_eof_4d

2.41.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.41.2 Latest revision

30/04/2026

2.41.3 Purpose

Performs an orthogonal rotation of a (partial) Empirical Orthogonal Function (EOF) or Principal Component Analysis (PCA) model (e.g., a factor loading matrix and the associated standardized amplitude time series) computed from a previous EOF or PCA analysis of a fourdimensional NetCDF variable. The factor loading matrix (e.g., the associated scaled EOF patterns corresponding to the selected rotated PC time series) are also rotated and the new rotated EOF or PCA model is stored in a NetCDF file.

Spatial EOF patterns (e.g., scaled EOFs) and Principal Component (PC) time series derived from EOF analysis or PCA are constrained to be both spatially and temporally orthogonal to one another so that they maximize the variance they described over the entire analysis domain and selected time period [Jolliffe] [vonStorch_Zwiers] . However, the spatial orthogonality of the EOFs is a strong and undesirable constraint imposed on the EOFs and, as a result, EOFs are subject to problems such as domain dependence and inaccurate representation of physical relationships embedded in the data [Jolliffe] [Richman] .

Orthogonal rotation of selected standardized PC time series can be used to obtain spatial and time patterns (e.g., rotated scaled EOFs and PC time series) that are hopefully more physically meaningful by relaxing the spatial orthogonality of the rotated scaled EOFs, while preserving the temporal orthogonality of the rotated standardized PC time series. Preserving the orthogonality of the (rotated) PC time series implies that these rotated PCs describe the same amount of total variance as the original (unrotated) PC time series, which is a very useful statistical property, but this variance is more equally (or differently) distributed among the rotated PC time series [Jolliffe] [Richman] .

More precisely, the orthogonal rotation in `comp_ortho_rot_eof_4d` has the objective to make the rotated standardized PC time series as simple as possible to interpret by attempting to constrain the spatial coefficients (or loadings) associated with each of them to be large or very small in absolute magnitude, but avoiding as much as possible spatial loadings of moderate size [Jolliffe] [Jackson] .

The orthogonal rotation matrix, which will perform this linear transformation is obtained by maximizing a generalized orthomax criterion, including quartimax, varimax and equamax rotation methods [Jackson] at the user option (see description of the `-w=` argument below for more details about the selection of the rotation method which will be used). This generalized orthomax criterion is described in details in [Clarkson_Jennrich] [Jackson] and amounts to obtain rotated standardized PC time series which are more spatially localized by relaxing the spatial orthogonality of the original EOF (or PCA) modes, while preserving the orthogonality of the (rotated) standardized PC time series as described above.

The `-w=rotation_factor` argument determines the kind of orthogonal rotation to be computed [Clarkson_Jennrich] [Jackson] and may be set as follows :

- `-w=0` . is the quartimax method [Jackson] , which attempts to get each variable (e.g., grid-point) to load highly on only one (or a few) EOFs or factors.
- `-w=1` . is the varimax method [Jackson] , which attempts to load highly a relatively low number of variables on each EOF or factor. Varimax is the most widely used method of orthogonal rotation.
- `-w=nrot/2` ., where `nrot` is the number of rotated PCs, is the equamax method [Jackson] , which is a compromise of the above two.

More generally, the value of the `-w=` argument can be any positive real number, but best values lie in the closed interval $[0., 5. * nrot]$. Generally, the larger the value is, the more equal is the dispersion of the variance accounted for across the rotated EOFs or factors.

As an illustration, by default, `comp_ortho_rot_eof_4d` uses the varimax rotation method, which simplifies the spatial structure of the rotated (scaled) EOFs by maximizing the sum of the variances of the squared columns loadings of the selected rotated (scaled) EOFs [Jolliffe] [Jackson] [vonStorch_Zwiers] . As a result, the rotated EOFs tend to have coefficients biased towards either large or zero values. Hence these rotated EOFs should be easier to interpret than their unrotated counterparts.

Note that, other methods of rotation exist to obtain more physically based EOFs and PC time series. For example, the orthogonal rotation matrix can be designed to maximize the energy in a selected frequency band for the rotated PC time series, see `comp_filt_rot_eof_4d` for more details.

Using as input an EOF NetCDF file produced by `comp_eof_4d`, `comp_ortho_rot_eof_4d` performs an orthogonal rotation of selected standardized PC time series readed from this input EOF NetCDF file and computes the regression spatial patterns associated with these rotated standardized PC time series (e.g., the orthogonal projection onto the orthonormal set formed by the new rotated standardized PC time series). The raw variances and percentages of variance described by the new rotated standardized PC time series are also computed.

The selected standardized PC time series, which will be rotated is determined on entry by the value of the **-se=** argument, which is a list of PC (or EOF) numbers.

Optionally, the rotated PC time series can be estimated with the help of a metric such that the results are weighted by the surface (or volume) associated with each cell in the grid associated with the input tridimensional NetCDF variable (which contains the original scaled EOFs) so that equal areas (or volumes) carry equal weights in the results of the rotated EOF analysis (see the **-d=** argument description).

An output NetCDF dataset containing these rotated PC standardized time series, their associated regression spatial patterns and the explained variance statistics is created.

If the NetCDF variable used in the original EOF analysis is tridimensional use *comp_ortho_rot_eof_3d* instead of *comp_ortho_rot_eof_4d*.

2.41.4 Further Details

Usage

```
$ comp_ortho_rot_eof_4d\
-f=eof_netcdf_file \
-v=eof_netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-se=selected_eofs           (optional) \
-g=grid_type                (optional : n, t, u, v, w, f) \
-r=resolution               (optional : r2, r4) \
-b=nlon_orca, nlat_orca, nlevel_orca (optional) \
-x=lon1,lon2                (optional) \
-y=lat1,lat2                (optional) \
-z=level1,level2            (optional) \
-t=time1,time2              (optional) \
-d=distance                 (optional : dist2, dist3, ident) \
-w=rotation_factor          (optional : 0. > +inf) \
-o=output_rot_eof_netcdf_file (optional) \
-mi=missing_value           (optional) \
-knorm                      (optional) \
-double                     (optional) \
-bigfile                    (optional) \
-hdf5                       (optional) \
-cdf5                       (optional) \
-tlimited                    (optional)
```

By default

- se=** all the PC time series stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 3-D grid-mesh associated with the input NetCDF EOF variable, *eof_netcdf_variable*, is assumed to be regular or Gaussian
- r=** if the input *eof_netcdf_variable* is from the NEMO ocean model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-n=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *eof_netcdf_variable*

- y=** the whole latitude domain associated with the *eof_netcdf_variable*
- z=** the whole vertical resolution associated with the *eof_netcdf_variable*
- t=** the whole time period associated with the input PC time series
- d=** the *distance* is set to the distance used in the original EOF analysis.
- w=** the *rotation_factor* is set to 0., which means that a varimax orthogonal rotation is performed.
- o=** the *output_rot_eof_netcdf_file* is named `ortho_rot_eof_netcdf_variable.nc`
- mi=** the *missing_value* attribute for the rotated EOFs in the output NetCDF file is set to 1.e+20
- knorm** the orthogonal rotation of the standardized PC time series is performed without Kaiser's row normalization. If **-knorm** is activated, Kaiser's row normalization is performed before the orthogonal rotation of the PC time series and the spatial loadings (e.g., the rotated EOFs) associated with the rotated PC time series are rescaled after the orthogonal rotation
- double** the results of the rotated EOF analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=***eof_netcdf_variable* argument specifies the NetCDF variable for which an EOF analysis was originally computed by *comp_eof_4d* and the **-f=***input_eof_netcdf_file* argument specifies that the resulting EOF patterns and PC time series must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_4d*.

These EOF patterns and PC time series are used to compute the orthogonal rotation matrix which will be applied to the selected EOF model.

- 2) The **-m=***input_mesh_mask_netcdf_file* argument specifies the land-sea mask to apply to the *eof_netcdf_variable* (which contains the scaled eigenvectors of the EOF analysis) for transforming this fourdimensional NetCDF variable as a rectangular matrix before performing the rotated EOF analysis. It is assumed that this land-sea mask is exactly the same as the one used in the original EOF analysis, whose results are stored in the input NetCDF file *eof_netcdf_file*.

The scale factors associated with the 3-D grid-mesh of the input NetCDF variable (needed if **-d=dist2** or **-d=dist3** is specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.

- 3) The **-se=***selected_eofs* argument allows the user to select the standardized PC time series which must be included in the orthogonal rotation. The list of selected PCs may be given in two formats:
 - **-se=**1,3,...,nn allows to include *pc1*, *pc3*,... and *pcnn* in the orthogonal rotation
 - **-se=**1 : 4 allows to include from *pc1* to *pc4* in the orthogonal rotation.

The two forms of the **-se=** argument may be combined and repeated any number of times. Duplicate PC numbers are not allowed. If the **-se=** argument is not specified, all the PC time series stored in the *input_eof_netcdf_file* are used in the orthogonal rotation.

- 4) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model. In this case, the duplicate points from the ORCA grid are removed before the rotated EOF analysis, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 3-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typemask*, *e1grid_type*, *e2grid_type* and *e3grid_type* respectively). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_4d* if the 3-D grid-mesh is regular or gaussian.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration of the NEMO ocean model
 - **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration of the NEMO ocean model.

- 6) If the NetCDF variable, *eof_netcdf_variable*, is from an experiment with the NEMO ocean model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

- 7) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are not specified, the geographical domain used in the rotated EOF analysis is determined from the attributes of the input mesh-mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit*, *lat2_Northern_limit*, *level1_First_level* and *level2_Last_level*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_4d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_ortho_rot_eof_4d*.

- 8) If the **-t=time1,time2** argument is missing, the whole time period associated with the original PC time series is used in the orthogonal rotation.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have *ntime = time2 - time1 + 1* observations.

- 9) The geographical shapes of the *eof_netcdf_variable* (in the *eof_netcdf_file*), the land-sea mask and scale factors (in the *input_mesh_mask_netcdf_file*) must agree.
- 10) The **-d=** argument specifies the metric and scalar product used in the rotated EOF analysis:
 - **-d=dist2** means that the orthogonal rotation is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid-point is weighted accordingly to the surface associated with it)
 - **-d=dist3** means that the projection is done with the diagonal distance associated with the whole 3D grid-mesh (e.g., each grid point is weighted accordingly to the volume or weight associated with it)

- **-d=ident** means that the orthogonal rotation is done with the identity metric, e.g., the usual Euclidean distance and scalar product are used in the rotated EOF analysis.

By default, the **-d=** argument is set to the distance used in the original EOF analysis.

- 11) The **-w=rotation_factor** argument specifies the exact form of the function criterion, which must be maximized over the set of `nrot` by `nrot` orthogonal rotation matrices (where `nrot` is the number of selected PCs as specified with the **-se=** argument). *rotation_factor* is a positive real scalar and may be set as follows:

- **-w=0.** is the quartimax method, which attempts to get each variable (e.g., grid-point) to load highly on only one (or a few) EOFs or factors.
- **-w=1.** is the varimax method, which attempts to load highly a relatively low number of variables on each EOF or factor. Varimax is the most widely used method of orthogonal rotation.
- **-w=nrot/2.**, where `nrot` is the number of rotated PCs, is the equamax method, which is a compromise of the above two.

More generally, *rotation_factor* can be any positive real number, but best values lie in the closed interval $[0., 5. * nrot]$. Generally, the larger the value of *rotation_factor* is, the more equal is the dispersion of the variance accounted for across the rotated EOFs or factors.

- 12) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_rot_eof_netcdf_file*.

If the **-mi=** argument is not specified *missing_value* is set to `1.e+20`.

- 13) The **-knorm** argument specifies that orthogonal rotation of the original PC time series must be performed using Kaiser's row normalization of the associated spatial loadings (e.g., the original scaled EOFs) and the computed spatial loadings (e.g., the rotated EOFs) associated with the rotated PC time series are rescaled after the orthogonal rotation.

- 14) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file, *output_rot_eof_netcdf_file*.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 15) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_rot_eof_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 18) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.

- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 20) For more details on EOF analysis, PCA and orthogonal rotation methods, see
 - “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300
 - “Principal Component Analysis”, by Jolliffe, I.T., Springer-Verlag, New York, USA, 487 pp., 2nd Ed, 2002. ISBN: 978-0-387-22440-4. <https://www.springer.com/gp/book/9780387954424>
 - “A user’s guide to principal components”, by Jackson, J.E., John Wiley and Sons, New York, USA, 592 pp., 2003. ISBN: 978-0-471-47134-9
 - “Quartic rotation criteria and algorithms” by Clarkson, D.B., and Jennrich, R.I., Psychometrika, Vol. 53, 251-259, 1988. <https://link.springer.com/article/10.1007/BF02294136>
 - “Rotation of principal components” by Richman, M.B., International Journal of Climatology, Vol. 6, 293-335, 1986. <https://doi.org/10.1002/joc.3370060305>

Outputs

`comp_ortho_rot_eof_4d` creates an output NetCDF file that contains the rotated standardized PC time series, their associated rotated EOF patterns and the variances described by the computed rotated PC time series.

The number of rotated PCs and EOFs, `nrot`, stored in the output NetCDF dataset is determined by the `-se=selected_eofs` argument. The number of observations in the output NetCDF dataset is determined from the `-t=time1,time2` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat`, `nlon` and `nlev` are the lengths of the geographical dimensions of the input NetCDF variable `eof_netcdf_variable` and `nrot` is the number of selected PC time series, which have been rotated as determined by the list specified with the `-se=selected_eofs` argument) :

- 1) `pc_number(nrot)` : the list of rotated PC time series as specified with the `-se=selected_eofs` argument.
- 2) `eof_netcdf_variable_ortho_rot_eof(nrot,nlev,nlat,nlon)` : the rotated EOF patterns. These rotated EOFs are scaled such that they give the scalar products (if `-a=scp` in the original EOF analysis), covariances (`-a=cov` in the original EOF analysis) or correlations (`-a=cor` in the original EOF analysis) between the original observed variables and the rotated PC time series.

The rotated EOF patterns are packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `eof_netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, this output NetCDF variable is filled with missing values.

- 3) `eof_netcdf_variable_ortho_rot_pc(ntime,nrot)` : the rotated standardized PC time series.

The rotated PC time series are always standardized to unit variance.

- 4) `eof_netcdf_variable_ortho_rot_std(nrot)` : the standard-deviations of the rotated PC time series.

The squares of these statistics are equal to the raw variance described by the rotated PC time series.

- 5) `eof_netcdf_variable_ortho_rot_var(nrot)` : the proportion of variance explained by each rotated PC time series (given between 0. and 1. with 1. corresponding to 100%).

Examples

- 1) For computing a varimax rotated 10-EOF analysis from a previous EOF analysis (stored in the file `eof_1m_197902_200501_votemper_oi.nc`) of a fourdimensional NetCDF variable named `votemper` and store the results of the rotated EOF analysis in a NetCDF file named `rot_eof_1m_197902_200501_votemper_oi.nc`, use the following command :

```
$ comp_ortho_rot_eof_4d \  
-f=eof_1m_197902_200501_votemper_oi.nc \  
-v=votemper \  
-se=1:10 \  
-m=oi_mask.nc \  
-o=rot_eof_1m_197902_200501_votemper_oi.nc
```

2.42 comp_project_eof_3d

2.42.1 Authors

Pascal Terray (LOCEAN/IPSL) and Eric Maisonnave (CERFACS)

2.42.2 Latest revision

30/04/2026

2.42.3 Purpose

Project a tridimensional NetCDF variable (or parts of it) extracted from a NetCDF dataset onto eigenvectors or singular vectors computed from a previous Empirical Orthogonal Function (EOF) or Singular Value Decomposition (SVD) analysis.

Using as input an EOF (or SVD) NetCDF file produced by `comp_eof_3d`, `comp_eof_miss_3d` or `comp_svd_3d`, this procedure computes the projection of a given tridimensional variable extracted from another NetCDF dataset onto the orthonormal basis formed by the eigenvectors or singular vectors of the EOF (or SVD) analysis.

The procedure first transforms the selected time steps of the input tridimensional NetCDF variable as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 2-D grid-mesh associated with the tridimensional NetCDF variable), does the same repacking transformation for the selected eigenvectors or singular vectors (which must have been computed on exactly the same selected cells of the 2-D grid-mesh associated with the input tridimensional NetCDF variable) and then computes the projections of the selected time steps onto the selected eigenvectors (or singular vectors) by performing the following matrix product:

$$\mathbf{A} = \mathbf{X}.\text{transpose}(\mathbf{B})$$

where

- **A** is the *ntime* by *k* matrix of the *k* selected principal components (or singular variables) time series to be computed
- **B** is the *k* by *nv* matrix of the *k* eigenvectors or singular vectors (stored rowwise) readed from an input NetCDF file produced by `comp_eof_3d`, `comp_eof_miss_3d` or `comp_svd_3d`.

If the NetCDF variable is fourdimensional use *comp_project_eof_4d* instead of *comp_project_eof_3d*.

An output NetCDF dataset containing the expansion coefficients for the selected time steps of the Principal Component (PC) or Singular Variable (SV) time series is created.

This procedure is parallelized if OpenMP is used.

2.42.4 Further Details

Usage

```
$ comp_project_eof_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -fe=eof_netcdf_file \
  -ve=eof_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -se=selected_eofs                (optional) \
  -g=grid_type                    (optional : n, t, u, v, w, f) \
  -r=resolution                  (optional : r2, r4) \
  -b=nlon_orca, nlat_orca        (optional) \
  -x=lon1,lon2                   (optional) \
  -y=lat1,lat2                   (optional) \
  -t=time1,time2                 (optional) \
  -l=selected_time_period        (optional) \
  -a=type_of_analysis            (optional : scp, cov, cor) \
  -c=input_climatology_netcdf_file (optional) \
  -d=distance                    (optional : dist2, ident) \
  -o=output_pc_netcdf_file       (optional) \
  -normpc                        (optional) \
  -svd                           (optional) \
  -double                        (optional) \
  -bigfile                       (optional) \
  -hdf5                          (optional) \
  -cdf5                          (optional) \
  -tlimited                       (optional)
```

By default

- se=** all the eigenvectors or singular vectors stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- l=** all the time steps from *time1* to *time2* as specified in the **-t=** argument

- a=** the *type_of_analysis* is set to `scp`. This means that the projection onto the eigenvectors is done on the raw data without centering or standardizing the input time series
- c=** an *input_climatology_netcdf_file* is not needed if the *type_of_analysis* is set to `scp`
- d=** the *distance* is set to `dist2`. This means that the scalar products for computing the projections are computed with the diagonal metric associated with the 2-D grid-mesh associated with the input NetCDF variable
- o=** the *output_pc_netcdf_file* is named `proj_netcdf_variable.nc`
- normpc** the computed PC (or SV) time series are not normalized in the output NetCDF file. If **-normpc** is activated, the computed PC (or SV) time series are normalized in the output NetCDF file
- svd** the *eof_netcdf_file* is assumed to be produced by `comp_eof_3d` or `comp_eof_miss_3d`. If **-svd** is activated, a file produced by `comp_svd_3d` is assumed, this means that the projection is done onto singular vectors of a previous SVD analysis
- double** the results of the projection analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a projection analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-ve=eof_netcdf_variable** argument specifies the NetCDF variable for which an EOF (or SVD) analysis was originally computed by `comp_eof_3d` (or `comp_svd_3d`) and the **-fe=input_eof_netcdf_file** argument specifies that the resulting EOF (or SVD) patterns must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by `comp_eof_3d` or `comp_svd_3d`. These EOF (or SVD) patterns will be used to compute the projections of the *netcdf_variable* specified by the **-v=** argument.
- 3) The **-m=input_mesh_mask_netcdf_file** argument specifies the land-sea mask to apply to the *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix before computing the EOF (or SVD) projection. The same land-sea mask is assumed and apply to the *eof_netcdf_variable* which contains the eigenvectors of the EOF (or SVD) analysis.

The scale factors associated with the 2-D grid-mesh of these NetCDF variables (needed if **-d=dist2** is specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.

- 4) The **-se=selected_eofs** argument allows the user to select the eigenvectors (or singular vectors) which must be included in the projection analysis. The list of selected vectors may be given in two formats:
 - **-se=1,3,...,nn** allows to include *eof1*, *eof3*,... and *eofnn* in the EOF (or SVD) projection
 - **-se=1 : 4** allows to include from *eof1* to *eof4* in the EOF (or SVD) projection.

The two forms of the **-se=** argument may be combined and repeated any number of times. Duplicate EOF or SVD numbers are not allowed. If the **-se=** argument is not specified, all the eigenvectors (or singular vectors) stored in the *input_eof_netcdf_file* are used in the projection analysis.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variables are from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before the EOF (or SVD) projection, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variables covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typemask*, *e1grid_type* and *e2grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by [comp_clim_3d](#) if the 2-D grid-mesh is regular or gaussian.

- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variables are from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:

- **-r=r2** the NetCDF variables are from an experiment with the ORCA R2 configuration
- **-r=r4** the NetCDF variables are from an experiment with the ORCA R4 configuration.

- 7) If the NetCDF variables are from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

- 8) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are not specified, the geographical domain used in the EOF projection is determined from the attributes of the input mesh mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_project_eof_3d](#).

- 9) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used in the projection analysis.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ observations if the **-l=** argument is missing.

- 10) The **-l=** argument lists the indices of the time steps which must be included in the output file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is not used). The list may be specified in two formats:

- **-l=n1,n2,...nn** allows to select for *n1*, *n2*, ... and *nn* time steps
- **-l=n1:n2** allows to select time steps from *n1* to *n2*.

The two forms of the **-l=** argument may be combined and repeated any number of times, but duplicate time steps are not allowed. Be careful with time period limits when specifying the **-l=** argument.

- 11) The **-a=** argument specifies if the observed variables have to be centered or standardized with an input climatology (specified with the **-c=** argument) before the projection analysis:

- **-a=sop** means that the projection analysis must be done on the raw data
 - **-a=cov** means that the projection analysis must be done on the anomalies
 - **-a=cor** means that the projection analysis must be done on the standardized anomalies.
- 12) The *input_climatology_netcdf_file* is needed only if **-a=cov** or **-a=cor**.
- 13) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 14) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the *eof_netcdf_variable* (in the *eof_netcdf_file*), the climatology (in the *input_climatology_netcdf_file*), the land-sea mask and the scale factors (in the *input_mesh_mask_netcdf_file*) must agree.
- 15) The **-d=** argument specifies the metric and scalar product used in the EOF or (SVD) projection:
- **-d=dist2** means that the projection is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident** means that the projection is done with the identity metric : the usual Euclidean distance and scalar product are used in the EOF (or SVD) projection.
- 16) The **-normpc** argument specifies that the computed PC (or SV) time series must be normalized with the reciprocal of the singular value of the associated EOF pattern stored in the *eof_netcdf_file* (or with the reciprocal of the previously computed standard-deviations of the SV time series stored in the *eof_netcdf_file* if the **-svd** argument is used).
- 17) The **-svd** argument specifies that the *eof_netcdf_file* is produced by *comp_svd_3d* instead of *comp_eof_3d*. This means that the projection will be done onto the singular vectors of a previous SVD analysis computed by *comp_svd_3d*.
- 18) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.
- By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 19) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.
- If this argument is specified, the *output_pc_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 20) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.
- If this argument is specified, the *output_pc_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 21) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.
- If this argument is specified, the *output_pc_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 22) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.

- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on EOF or SVD analysis in the climate literature, see
- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300

Outputs

`comp_project_eof_3d` creates an output NetCDF file that contains the new principal component (or singular variables) time series computed from the eigenvectors (or singular vectors) of a previous EOF (or SVD) analysis. The number of principal components stored in the output NetCDF dataset is determined by the `-se=selected_eofs` argument. The number of observations in the output NetCDF dataset is determined from the `-t=time1,time2` and `-l=selected_time_period` arguments. The output NetCDF dataset contains the following NetCDF variable :

- 1) `netcdf_variable_pc(ntime, number_of_eofs)` : the new principal component time series corresponding to the projection onto the selected eigenvectors or singular vectors.

The new principal component time series are standardized with the standard-deviations estimated from the previous EOF (or SVD) analysis if the `-normpc` argument is specified.

Examples

- 1) For computing a 10-EOF projection of a NetCDF variable named `sst` in the NetCDF file `ersst_2m_197902_200501_sst_oi.nc` on the 10 first EOF patterns of a previous EOF analysis stored in the file named `eof_HadISST1_2m_197902_200501_sst_oi.nc` and store the results in a NetCDF file named `ersst_2m_197902_200501_sst_oi_10pc.nc`, use the following command :

```
$ comp_project_eof_3d \
-f=ersst_2m_197902_200501_sst_oi.nc \
-v=sst \
-fe=eof_HadISST1_2m_197902_200501_sst_oi.nc \
-ve=sst \
-se=1:10 \
-m=ersst_mask.nc \
-o=ersst_2m_197902_200501_sst_oi_10pc.nc
```

2.43 comp_project_eof_4d

2.43.1 Authors

Pascal Terray (LOCEAN/IPSL) and Eric Maisonnave (CERFACS)

2.43.2 Latest revision

30/04/2026

2.43.3 Purpose

Project a fourdimensional NetCDF variable (or parts of it) extracted from a NetCDF dataset onto eigenvectors or singular vectors computed from a previous Empirical Orthogonal Function (EOF) or Singular Value Decomposition (SVD) analysis.

Using as input, an EOF (or SVD) NetCDF file produced by `comp_eof_4d` or `comp_svd_3d`, this procedure computes the projection of a given fourdimensional variable extracted from another NetCDF dataset onto the orthonormal basis formed by the eigenvectors or singular vectors of the EOF (or SVD) analysis.

The procedure first transforms the selected time steps of the input fourdimensional NetCDF variable as a *ntime* by *nv* rectangular matrix, **X**, of observed variables (e.g., the selected cells of the 3-D grid-mesh associated with the fourdimensional NetCDF variable), does the same repacking operation for the selected eigenvectors or singular vectors (which must have been computed on the same selected cells of the 3-D grid-mesh associated with the input fourdimensional NetCDF variable) and then computes the projections of the selected time steps onto the selected eigenvectors (or singular vectors) by performing the following matrix product

$$\mathbf{A} = \mathbf{X}.\text{transpose}(\mathbf{B})$$

where

- **A** is the *ntime* by *k* matrix of *k* selected principal components (or singular variables) time series to be computed
- **B** is the *k* by *nv* matrix of the *k* eigenvectors or singular vectors (stored rowwise) readed from an input NetCDF file produced by `comp_eof_4d` or `comp_svd_3d`.

If the NetCDF variable is tridimensional use `comp_project_eof_3d` instead of `comp_project_eof_4d`.

An output NetCDF dataset containing the expansion coefficients for the selected time steps of the Principal Component (PC) or Singular Variable (SV) time series is created.

This procedure is parallelized if OpenMP is used.

2.43.4 Further Details

Usage

```
$ comp_project_eof_4d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -fe=eof_netcdf_file \
  -ve=eof_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -se=selected_eofs                (optional) \
  -g=grid_type                     (optional : n, t, u, v, w, f) \
  -r=resolution                    (optional : r2, r4) \
  -b=nlon_orca, nlat_orca          (optional) \
  -x=lon1,lon2                    (optional) \
  -y=lat1,lat2                    (optional) \
  -z=level1,level2                (optional) \
  -t=time1,time2                  (optional) \
  -l=selected_time_period          (optional) \
  -a=type_of_analysis              (optional : scp, cov, cor) \
  -c=input_climatology_netcdf_file (optional) \
  -d=distance                      (optional : dist2, dist3, ident) \
  -o=output_pc_netcdf_file         (optional) \
  -normpc                         (optional) \
  -svd                            (optional) \
```

(continues on next page)

(continued from previous page)

-double	(optional) \
-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- se=** all the eigenvectors or singular vectors stored in the *eof_netcdf_file*
- g=** the *grid_type* is set to *n* which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-n=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this default choice with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- l=** all the time steps from *time1* to *time2* as specified in the **-t=** argument
- a=** the *type_of_analysis* is set to *scp*. This means that the projection onto the eigenvectors is done on the raw data without centering or standardizing the input time series
- c=** an *input_climatology_netcdf_file* is not needed if the *type_of_analysis* is set to *scp*
- d=** the *distance* is set to *dist3*. This means that the scalar products for computing the projections are computed with the diagonal metric associated with the 3-D grid-mesh associated with the input NetCDF variable
- o=** the *output_pc_netcdf_file* is named *proj_netcdf_variable.nc*
- normpc** the computed PC (or SV) time series are not normalized in the output NetCDF file. If **-normpc** is activated, the computed PC (or SV) time series are normalized in the output NetCDF file
- svd** the *eof_netcdf_file* is assumed to be produced by *comp_eof_4d* or *comp_eof_miss_3d*. If **-svd** is activated, a file produced by *comp_svd_3d* is assumed, this means that the projection is done onto singular vectors of a previous SVD analysis.
- double** the results of the projection analysis are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a projection analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-ve=eof_netcdf_variable** argument specifies the NetCDF variable for which an EOF (or SVD) analysis was originally computed by *comp_eof_4d* (or *comp_svd_3d*) and the **-fe=input_eof_netcdf_file** argument specifies that the resulting EOF (or SVD) patterns must be extracted from the NetCDF file, *eof_netcdf_file*. This NetCDF file must have exactly the same format as the files produced by *comp_eof_4d* or *comp_svd_3d*. These EOF (or SVD) patterns will be used to compute the projections of the *netcdf_variable* specified by the **-v=** argument.
- 3) The **-m=input_mesh_mask_netcdf_file** argument specifies the land-sea mask to apply to the *netcdf_variable* for transforming this fourdimensional NetCDF variable as a rectangular matrix before computing the EOF (or SVD) projection. The same land-sea mask is assumed and apply to the *eof_netcdf_variable* which contains the eigenvectors of the EOF (or SVD) analysis.

The scale factors associated with the 2-D grid-mesh of these NetCDF variables (needed if **-d=dist2** is specified when calling the procedure) are also read from the *input_mesh_mask_netcdf_file*.

- 4) The **-se=selected_eofs** argument allows the user to select the eigenvectors (or singular vectors) which must be included in the projection analysis. The list of selected vectors may be given in two formats:
 - **-se=1,3,...,nn** allows to include *eof1*, *eof3*, ... and *eofnn* in the EOF (or SVD) projection
 - **-se=1 : 4** allows to include from *eof1* to *eof4* in the EOF (or SVD) projection.

The two forms of the **-se=** argument may be combined and repeated any number of times. Duplicate EOF or SVD numbers are not allowed. If the **-se=** argument is not specified, all the eigenvectors (or singular vectors) stored in the *input_eof_netcdf_file* are used in the projection analysis.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variables are from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before the EOF (or SVD) projection, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variables covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 3-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typemask*, *e1grid_type*, *e2grid_type* and *e3grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by *comp_clim_4d* if the 3-D grid-mesh is regular or gaussian.

- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the input NetCDF variables are from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variables are from an experiment with the ORCA R2 configuration.
 - **-r=r4** the NetCDF variables are from an experiment with the ORCA R4 configuration.
- 7) If the NetCDF variables are from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 8) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are not specified, the geographical domain used in the EOF projection is determined from the attributes of the input mesh-mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit*, *lat2_Northern_limit*, *level1_First_level* and *level2_Last_level*), which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the

netcdf_variable is used in the EOF projection. These arguments must be set normally to the same values as used in the original EOF analysis, whose results are stored in the input NetCDF file, *eof_netcdf_file*.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_project_eof_4d*.

- 9) If the **-t=***time1,time2* argument is missing, the whole time period associated with the *netcdf_variable* is used in the projection analysis.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ observations if the **-l=** argument is missing.

- 10) The **-l=** argument lists the indices of the time steps which must be included in the output file. The indices of the time steps are counted from the start of the (selected) time period (e.g., *time1* in the **-t=***time1,time2* argument or 1 if this argument is not used). The list may be specified in two formats:

- **-l=***n1,n2,...nn* allows to select for *n1*, *n2*, ... and *nn* time steps
- **-l=***n1:n2* allows to select time steps from *n1* to *n2*.

The two forms of the **-l=** argument may be combined and repeated any number of times. Duplicate time steps are not allowed. Be careful with time period limits when specifying the **-l=** argument.

- 11) The **-a=** argument specifies if the observed variables have to be centered or standardized with an input climatology (specified with the **-c=** argument) before the projection analysis:

- **-a=***scp* means that the projection analysis must be done on the raw data.
- **-a=***cov* means that the projection analysis must be done on the anomalies.
- **-a=***cor* means that the projection analysis must be done on the standardized anomalies.

- 12) The *input_climatology_netcdf_file* is needed only if **-a=cov** or **-a=cor**.

- 13) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.

- 14) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the *eof_netcdf_variable* (in the *eof_netcdf_file*), the climatology (in the *input_climatology_netcdf_file*), the land-sea mask and the scale factors (in the *input_mesh_mask_netcdf_file*) must agree.

- 15) The **-d=** argument specifies the metric and scalar product used in the EOF or (SVD) projection:

- **-d=***dist2* means that the projection is done with the diagonal distance associated with the horizontal 2-D grid-mesh (e.g., each grid point is weighted accordingly to the surface associated with it)
- **-d=***dist3* means that the projection is done with the diagonal distance associated with the whole 3D grid-mesh (e.g., each grid point is weighted accordingly to the volume or weight associated with it)
- **-d=***ident* means that the projection is done with the identity metric : the usual Euclidean distance and scalar product are used in the EOF (or SVD) projection.

By default, the **-d=** argument is set to *dist3*.

- 16) The **-normpc** argument specifies that the computed PC (or SV) time series must be normalized with the reciprocal of the singular value of the associated EOF pattern stored in the *eof_netcdf_file* (or with the reciprocal of the

previously computed standard-deviations of the SV time series stored in the *eof_netcdf_file* if the **-svd** argument is used).

- 17) The **-svd** argument specifies that the *eof_netcdf_file* is produced by *comp_svd_3d* instead of *comp_eof_4d*. This means that the projection will be done onto the singular vectors of a previous SVD analysis computed by *comp_svd_3d*.
- 18) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 19) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_pc_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_pc_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 21) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_pc_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 22) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on EOF or SVD analysis in the climate literature, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: 9780521012300

Outputs

comp_project_eof_4d creates an output NetCDF file that contains the new principal component (or singular variables) time series computed from the eigenvectors (or singular vectors) of a previous EOF (or SVD) analysis. The number of principal components stored in the output NetCDF dataset is determined by the **-se=selected_eofs** argument. The number of observations in the output NetCDF dataset is determined from the **-t=time1,time2** and **-l=selected_time_period** arguments. The output NetCDF dataset contains the following NetCDF variable :

- 1) *netcdf_variable_pc*(*ntime*, *number_of_eofs*) : the new principal component time series corresponding to the projection onto the selected eigenvectors or singular vectors.

The new principal component time series are standardized with the standard-deviations estimated from the previous EOF (or SVD) analysis if the **-normpc** argument is specified.

Examples

- 1) For computing a 5-EOF projection of a NetCDF variable named `votemper` in the NetCDF file `ST7_1m_20101_30012_grid_T_votemper.nc` on the 5 first EOF patterns of a previous EOF analysis stored in the file named `eof_ST7_1m_0101_20012_grid_T_votemper.nc` and store the results in a NetCDF file named `ST7_1m_20101_30012_grid_T_votemper_10pc.nc`, use the following command :

```
$ comp_project_eof_4d \
-f=ST7_1m_20101_30012_grid_T_votemper.nc \
-v=votemper \
-fe=eof_ST7_1m_0101_20012_grid_T_votemper.nc \
-ve=votemper \
-se=1:5 \
-m=ST7_grid_T_votemper_mask.nc \
-o=ST7_1m_20101_20012_grid_T_votemper_10pc.nc
```

2.44 comp_reg_1d

2.44.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.44.2 Latest revision

29/04/2026

2.44.3 Purpose

Estimate polynomial trends and regression models from a time series extracted from a unidimensional variable in a NetCDF dataset and, optionally, remove the linear terms from the data by using linear least squares estimation and store the residuals and/or the predictions in a NetCDF dataset.

Optionally, regression diagnostics and statistical tests to diagnose the quality of the fitted regression models may also be computed and stored [[Rusta](#)] [[Rustb](#)] .

Finally, if the NetCDF variable is tridimensional use [comp_reg_3d](#) instead of `comp_reg_1d` and if the NetCDF variable is fourdimensional use [comp_reg_4d](#) instead of `comp_reg_1d`.

2.44.4 Further Details

Usage

```

$ comp_reg_1d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -t=time1,time2 (optional) \
  -p=periodicity (optional) \
  -a=type_of_analysis (optional : reg, residual, predict, all) \
  -o=output_netcdf_file (optional) \
  -to=time_origin (optional) \
  -dg=polynomial_degree (optional) \
  -fi=input_index_netcdf_file (optional) \
  -vi=index_netcdf_variable (optional) \
  -ti=itime1,itime2 (optional) \
  -pi=iperiodicity,istep (optional) \
  -ni=index_for_2d_index_netcdf_variable (optional) \
  -sm=smoothing_factor (optional) \
  -dv=dv_time1,dv_time2 (optional) \
  -mi=missing_value (optional) \
  -use_eps=tol (optional) \
  -comp_min_norm (optional) \
  -add_mean (optional) \
  -rsquare (optional) \
  -adjrsquare (optional) \
  -stderr (optional) \
  -ftest (optional) \
  -ttest (optional) \
  -double (optional) \
  -hdf5 (optional) \
  -cdf5 (optional) \
  -tlimited (optional)

```

By default

- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1
- a=** *type_of_analysis* is set to *reg*
- o=** *output_netcdf_file* is set to *reg_netcdf_variable.index_netcdf_variable.nc* if the **-vi=** argument is used and to *reg_netcdf_variable.poly_trend.nc* otherwise
- to=** If a polynomial trend is estimated, the origin for the time index is set to 0
- dg=** the *polynomial_degree* is set 1. This means that a linear trend is estimated if the **-vi=** argument is not used
- fi=** a polynomial trend is estimated if the **-vi=** argument is not used, otherwise the **-fi=** argument may be used to specified the NetCDF dataset for extracting the *index_netcdf_variable*
- vi=** a polynomial trend is estimated if the **-vi=** argument is not used
- ti=** the whole time period associated with the *index_netcdf_variable* if the **-vi=index_netcdf_variable** is specified
- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- sm=** no smoothing is applied to the *index_netcdf_variable* if the **-vi=** argument is used
- dv=** a dummy variable is not used

- mi=** the *missing_value* is set to $1.e+20$ for the NetCDF variables in the *output_netcdf_file*
- use_eps=** no tolerance is used for solving the linear least square problem associated with the specified regression model. If **-use_eps=** is used, the specified tolerance *tol* is used for solving the linear least square problem
- comp_min_norm** the minimal norm solution is not computed. If **-comp_min_norm** is activated, the minimal norm solution of the linear least square problem is computed
- add_mean** the means are not added to the residuals. If **-add_mean** is activated, the means are added to the residuals
- rsquare** the coefficient of determination is not computed. If **-rsquare** is activated, the coefficient of determination is computed
- adjrsquare** the adjusted coefficient of determination is not computed. If **-adjrsquare** is activated, the adjusted coefficient of determination is computed
- stderr** the standard errors of the estimates are not computed. If **-stderr** is activated, the standard errors of the estimates are computed
- ftest** a F-test for the regression model is not computed. If **-ftest** is activated, a F-test is computed
- ttest** Student-tests for the regression coefficients are not computed. If **-ttest** is activated, the Student-tests are computed
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a regression analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., $time2 - time1 + 1$) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.
- 3) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. If the **-p=periodicity** argument is specified, the regression models are computed by taking into account the periodicity of the data. This means that *periodicity* regression models are estimated for the input tridimensional NetCDF variable.
- 4) The **-a=**argument specifies the statistics which must be computed and if the residuals from the trend or regression models are stored in the output NetCDF file. If:
 - **-a=reg**, the default, the residuals or predictions are not computed and only the regression and intercept coefficients are computed and stored

- **-a=residual**, the residuals are computed and stored, in addition of the regression and intercept coefficients
- **-a=predict**, the predictions are computed and stored, in addition of the regression and intercept coefficients
- **-a=all**, the residuals and predictions are computed and stored, in addition of the regression and intercept coefficients.

- 5) The **-to=time_origin** argument specifies the origin for the time index if a polynomial regression model is used (e.g., when the **-vi=** argument is not used).

By default, the origin for the time index is set to 0. This shift of the zero point for the time index makes the estimate for the intercept(s) in the polynomial model equal to the model prediction(s) for the first observation at the beginning of the record.

- 6) The **-dg=polynomial_degree** argument specifies the degree of the polynomial if a polynomial trend regression model is used (e.g., when the **-vi=** argument is not used).

if **-dg=1** a linear trend is used, if **-dg=2** a quadratic trend is used, etc.

polynomial_degree must be a strictly positive integer.

- 7) The **-vi=index_netcdf_variable** specifies a predictor time series for the regression model (e.g., an independent variable).

If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset, which contains the *index_netcdf_variable*, is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

If the **-vi=index_netcdf_variable** is not specified, a regression model with a polynomial trend is assumed and the **-dg=** argument specifies the degree of the polynomial : if **-dg=1** a linear trend is used, if **-dg=2** a quadratic trend is used, etc.

- 8) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. If the **-vi=** argument is not present, this argument is not used.

- 9) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(\text{time2} - \text{time1} + 1) / \text{periodicity} = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$

otherwise, an error message will be issued and the program will stop.

- 10) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute regression models with the January monthly time series extracted from the *index_netcdf_variable*, which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.

If the **-vi=** argument is not present, this argument is not used.

- 11) The **-ni=** argument specifies the index (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable.

- 12) The **-sm=smoothing_factor** means that the time series associated with the *index_netcdf_variable* (e.g., the **-vi=** argument) must be smoothed with a moving average of approximately $2 \cdot \text{smoothing_factor} + 1$ terms before estimating the regression parameters for predicting the *netcdf_variable* (e.g., the **-v=** argument) from the *index_netcdf_variable*. *smoothing_factor* must be a strictly positive integer.

If the **-vi=** argument is not present, this argument is not used and has no effect.

- 13) If the **-dv=dv_time1,dv_time2** argument is specified, a dummy variable is also included in the regression model. The dummy variable is an absence/presence variable (e.g., with values 0 or 1) and the time observations where the dummy variable is equal to 1 is specified by the *dv_time1* and *dv_time2* integers. The *dv_time1* and *dv_time2* integers specify the first and last time observations of the selected time period in which the dummy variable is set to 1.

These time indices are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing) and must take into account the periodicity of the data if the **-p=** argument is specified.

- 14) The **-mi=missing_value** argument specifies the missing value indicator associated with the NETCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, *missing_value* is set to 1.e+20.

- 15) The **-use_eps=tol** argument is used to determine the effective rank of the linear least squares problem. *tol* must be set to the relative precision of the elements in the NETCDF variables specify by the **-v=** and **-vi=** arguments.

If each element is correct to, say, 5 digits then *tol* = 0.00001 should be used. *tol* must not be greater or equal to 1 or less than 0, otherwise an error message is printed and the program stops. If the **-use_eps=** argument is not used, the numerical rank is determined.

- 16) If **-comp_min_norm** is specified, a complete orthogonal factorization of the coefficient matrix and the minimum 2-norm solutions are computed.
- 17) If **-add_mean** is specified, the means are added to the residuals of the regression model. This option has an effect only if **-a=residual** or **-a=all** (e.g., if the residuals are computed and stored in the *output_netcdf_file*).

By default, the means of the residuals in the output NetCDF file are zero.

- 18) If **-rsquare** is specified, the coefficient of determination of the specified model is computed.
- 19) If **-adjrsquare** is specified, the adjusted coefficient of determination of the specified model is computed.
- 20) If **-stderr** is specified, the standard errors of the estimated regression coefficients are computed (unless the specified model is not of full rank).
- 21) If **-ftest** is specified, a F-test is performed to test the null hypothesis that all the regression coefficients are zero (excepted the intercept).
- 22) If **-ttest** is specified, Student-tests are performed to test the null hypothesis that the regression coefficients are zero (independently of the other regression coefficients).
- 23) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 24) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 25) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 26) The **-tlimited** argument specifies that the time dimension must be defined as limited in the output NetCDF file. By default, this time dimension is defined as unlimited in the output NetCDF file.
- 27) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 28) It is assumed that the data has no missing values.
- 29) For more details on regression analysis in the climate literature, see
- “Fitting nature s basic functions Part I: polynomials and linear least squares”, by Rust, B.W., Computing in Science and Engineering, Vol. 3, no 5, 84-89, 2001. doi: [10.1109/MCISE.2001.947111](https://doi.org/10.1109/MCISE.2001.947111)
 - “Fitting nature s basic functions Part II: estimating uncertainties and testing hypotheses”, by Rust, B.W., Computing in Science and Engineering, Vol. 3, no 6, 60-64, 2001. doi: [10.1109/5992.963429](https://doi.org/10.1109/5992.963429)
 - “Fitting nature s basic functions Part III: exponentials, sinusoids, and nonlinear least squares”, by Rust, B.W., Computing in Science and Engineering, Vol. 4, no 4, 72-77, 2002. doi: [10.1109/MCISE.2002.1014982](https://doi.org/10.1109/MCISE.2002.1014982)
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: [9780521012300](https://www.amazon.com/Statistical-Analysis-Climate-Research/dp/0521012300)

Outputs

`comp_reg_1d` creates an output NetCDF file that contains the regression statistics and statistical tests associated with these coefficients, taking into account eventually the periodicity of the data as determined by the **-p=periodicity** argument.

If the **-vi=index_netcdf_variable** is specified, the output NetCDF data set will have *periodicity* time observations and may contain the following NetCDF variables depending on the arguments used in calling the procedure :

- 1) `netcdf_variable_index_netcdf_variable_reg0(periodicity)` : the intercept coefficient in the regression model for predicting the time series of the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 2) `netcdf_variable_index_netcdf_variable_reg1(periodicity)` : the regression coefficient in the regression model for predicting the time series of the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.

By default, the regression coefficient is expressed in units of the input NetCDF variable `netcdf_variable` by unit of the `index_netcdf_variable` time series.

- 3) `netcdf_variable_index_netcdf_variable_stderr0(periodicity)` : the standard-error of the intercept coefficient in the regression model for predicting the time series of the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 4) `netcdf_variable_index_netcdf_variable_stderr1(periodicity)` : the standard-error of the regression coefficient in the regression model for predicting the time series of the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 5) `netcdf_variable_index_netcdf_variable_tprob0(periodicity)` : the Student t-test probability associated with the intercept coefficient in the regression model for predicting the time series of the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 6) `netcdf_variable_index_netcdf_variable_tprob1(periodicity)` : the Student t-test probability associated with the regression coefficient in the regression model for predicting the time series of the the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.

- 7) *netcdf_variable_index_netcdf_variable_dv*(periodicity) : the regression coefficient associated with the dummy variable time series if a dummy variable is included in the regression model.

This variable is stored only if the **-dv=** argument has been specified when calling `comp_reg_1d`.

- 8) *netcdf_variable_index_netcdf_variable_dv_stderr*(periodicity) : the standard-error of the regression coefficient associated with the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the **-dv=** argument has been specified when calling `comp_reg_1d`.

- 9) *netcdf_variable_index_netcdf_variable_dv_tprob*(periodicity) : the Student t-test probability associated with the dummy variable time series if a dummy variable is included in the regression model.

This variable is stored only if the **-dv=** argument has been specified when calling `comp_reg_1d`.

- 10) *netcdf_variable_index_netcdf_variable_r2*(periodicity) : the r-square statistic associated with the regression model for predicting the time series associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.

- 11) *netcdf_variable_index_netcdf_variable_adj_r2*(periodicity) : the adjusted r-square statistic associated with the regression model for predicting the time series associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.

- 12) *netcdf_variable_index_netcdf_variable_fprob*(periodicity) : the F-test probability associated with the regression model for predicting the time series associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.

- 13) *netcdf_variable_index_netcdf_variable_predict*(ntime) : the predictions associated with the regression model for predicting the time series associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.

- 14) *netcdf_variable_index_netcdf_variable_resid*(ntime) : the residuals associated with the regression model for predicting the time series associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.

If the **-vi=index_netcdf_variable** is not specified when calling the procedure, similar statistics will be produced and each component of the selected polynomial trend regression model will have regression coefficients, standard-errors and Student t-test probabilities associated with it. As an illustration, the intercept and the regression coefficients (e.g., the slope) in a linear trend regression model will be stored in NetCDF variables *netcdf_variable_poly_trend_reg0* and *netcdf_variable_poly_trend_reg1*, respectively. For a quadratic trend regression model, in addition to these variables, the regression coefficients associated with the quadratic component will be stored in a NetCDF variable *netcdf_variable_poly_trend_reg2*. The same naming conventions are used for the standard-errors and Student t-test probabilities associated with each component of the selected polynomial trend regression model.

Examples

- 1) For polynomial detrending of a unidimensional NetCDF variable *sosstsst* in the NetCDF file *ST7_1m_sst_nino34.nc* and store the results in a NetCDF file named *ST7_1m_sst_nino34_detrended.nc*, use the following command (note that quadratic detrending is used since **-dg=2** is specified and cyclostationarity is assumed for the *sosstsst* variable since **-p=12** is also specified) :

```
$ comp_reg_1d \
-f=ST7_1m_sst_nino34.nc \
-v=sosstsst \
```

(continues on next page)

(continued from previous page)

```
-dg=2 \
-p=12 \
-a=residual \
-o=ST7_1m_sst_nino34_detrended.nc
```

2.45 comp_reg_3d

2.45.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.45.2 Latest revision

29/04/2026

2.45.3 Purpose

Estimate polynomial trends and regression models from time series extracted from a tridimensional variable in a NetCDF dataset and, optionally, remove the linear terms from the data by using linear least squares estimation and store the residuals and/or the predictions in a NetCDF dataset.

Optionally, regression diagnostics and statistical tests to diagnose the quality of the fitted regression models may also be computed and stored [[Rusta](#)] [[Rustb](#)] .

Finally, if the NetCDF variable is fourdimensional use [comp_reg_4d](#) instead of `comp_reg_3d` and if the NetCDF variable is unidimensional use [comp_reg_1d](#) instead of `comp_reg_3d`.

This procedure is parallelized if OpenMP is used.

2.45.4 Further Details

Usage

```
$ comp_reg_3d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file           (optional) \
-g=grid_type                             (optional : n, t, u, v, w, f) \
-x=lon1,lon2                             (optional) \
-y=lat1,lat2                             (optional) \
-t=time1,time2                           (optional) \
-p=periodicity                           (optional) \
-a=type_of_analysis                      (optional : reg, residual, predict, all) \
-o=output_netcdf_file                    (optional) \
-to=time_origin                          (optional) \
-dg=polynomial_degree                    (optional) \
-fi=input_index_netcdf_file               (optional) \
-vi=index_netcdf_variable                 (optional) \
-ti=itime1,itime2                        (optional) \
-pi=iperiodicity,istep                    (optional) \
```

(continues on next page)

(continued from previous page)

```

-ni=index_for_2d_index_netcdf_variable    (optional) \
-sm=smoothing_factor                      (optional) \
-dv=dv_time1,dv_time2                    (optional) \
-mi=missing_value                        (optional) \
-use_eps=tol                             (optional) \
-comp_min_norm                           (optional) \
-add_mean                                (optional) \
-rsquare                                 (optional) \
-adjrsquare                              (optional) \
-stderr                                  (optional) \
-ftest                                   (optional) \
-ttest                                   (optional) \
-double                                  (optional) \
-bigfile                                 (optional) \
-hdf5                                    (optional) \
-cdf5                                    (optional) \
-tlimited                                 (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1
- a=** *type_of_analysis* is set to *reg*
- o=** *output_netcdf_file* is set to *reg_netcdf_variable.index_netcdf_variable.nc* if the **-vi=** argument is used and to *reg_netcdf_variable.poly_trend.nc* otherwise
- to=** If a polynomial trend is estimated, the origin for the time index is set to 0
- dg=** the *polynomial_degree* is set 1. This means that a linear trend is estimated if the **-vi=** argument is not used
- fi=** a polynomial trend is estimated if the **-vi=** argument is not used, otherwise the **-fi=** argument may be used to specified the NetCDF dataset for extracting the *index_netcdf_variable*
- vi=** a polynomial trend is estimated if the **-vi=** argument is not used
- ti=** the whole time period associated with the *index_netcdf_variable* if the **-vi=index_netcdf_variable** is specified
- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- sm=** no smoothing is applied to the *index_netcdf_variable* if the **-vi=** argument is used
- dv=** a dummy variable is not used
- mi=** the *missing_value* is set to *1.e+20* for the NetCDF variables in the *output_netcdf_file*

- use_eps=** no tolerance is used for solving the linear least square problem associated with the specified regression model. If **-use_eps=** is used, the specified tolerance *tol* is used for solving the linear least square problem
- comp_min_norm** the minimal norm solution is not computed. If **-comp_min_norm** is activated, the minimal norm solution of the linear least square problem is computed
- add_mean** the means are not added to the residuals. If **-add_mean** is activated, the means are added to the residuals
- rsquare** the coefficient of determination is not computed. If **-rsquare** is activated, the coefficient of determination is computed
- adjrsquare** the adjusted coefficient of determination is not computed. If **-adjrsquare** is activated, the adjusted coefficient of determination is computed
- stderr** the standard errors of the estimates are not computed. If **-stderr** is activated, the standard errors of the estimates are computed
- ftest** a F-test for the regression model is not computed. If **-ftest** is activated, a F-test is computed
- ttest** Student-tests for the regression coefficients are not computed. If **-ttest** is activated, the Student-tests are computed
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a regression analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this tridimensional NetCDF variable as a rectangular matrix of observed variables before estimating the regression model. By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to *comp_clim_3d* or *comp_mask_3d* for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_reg_3d*.

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f*, it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the mesh_mask variable if an *input_mesh_mask_netcdf_file* is used.

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon + lon1 + 1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_reg_3d*.

- 5) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., $time2 - time1 + 1$) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.

- 6) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc.

If the **-p=periodicity** argument is specified, the regression models are computed by taking into account the periodicity of the data. This means that *periodicity* regression models are estimated for each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variable.

- 7) The **-a=**argument specifies the statistics which must be computed and if the residuals from the trend or regression models are stored in the output NetCDF file. If:

- **-a=reg**, the default, the residuals or predictions are not computed and only the regression and intercept coefficients are computed and stored
- **-a=residual**, the residuals are computed and stored, in addition of the regression and intercept coefficients
- **-a=predict**, the predictions are computed and stored, in addition of the regression and intercept coefficients
- **-a=all**, the residuals and predictions are computed and stored, in addition of the regression and intercept coefficients.

- 8) The **-to=time_origin** argument specifies the origin for the time index if a polynomial regression model is used (e.g., when the **-vi=** argument is not used).

By default, the origin for the time index is set to 0. This shift of the zero point for the time index makes the estimate for the intercept(s) in the polynomial model equal to the model prediction(s) for the first observation at the beginning of the record.

- 9) The **-dg=polynomial_degree** argument specifies the degree of the polynomial if a polynomial trend regression model is used (e.g., when the **-vi=** argument is not used).

If **-dg=1** a linear trend is used, if **-dg=2** a quadratic trend is used, etc.

polynomial_degree must be a strictly positive integer.

- 10) The **-vi=index_netcdf_variable** specifies a predictor time series for the regression model (e.g., an independent variable).

If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset, which contains the *index_netcdf_variable*, is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

If the **-vi=index_netcdf_variable** is not specified a regression model with a polynomial trend is assumed and the **-dg=** argument specifies the degree of the polynomial : if **-dg=1** a linear trend is used, if **-dg=2** a quadratic trend is used, etc.....

- 11) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. If the **-vi=** argument is not present, this argument is not used.

- 12) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(\text{time2} - \text{time1} + 1) / \text{periodicity} = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$

otherwise, an error message will be issued and the program will stop.

- 13) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute regression models with the January monthly time series extracted from the *index_netcdf_variable*, which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.

If the **-vi=** argument is not present, this argument is not used.

- 14) The **-ni=** argument specifies the index (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable.
- 15) The **-sm=smoothing_factor** means that the time series associated with the *index_netcdf_variable* (e.g., the **-vi=** argument) must be smoothed with a moving average of approximately $2 \cdot \text{smoothing_factor} + 1$ terms before estimating the regression parameters for predicting the *netcdf_variable* (e.g., the **-v=** argument) from the *index_netcdf_variable*. *smoothing_factor* must be a strictly positive integer.

If the **-vi=** argument is not present, this argument is not used.

- 16) If the **-dv=dv_time1,dv_time2** argument is specified, a dummy variable is also included in the regression model. The dummy variable is an absence/presence variable (e.g., with values 0 or 1) and the time observations where the dummy variable is equal to 1 is specified by the *dv_time1* and *dv_time2* integers. The *dv_time1* and *dv_time2* integers specify the first and last time observations of the selected time period in which the dummy variable is set to 1.

These time indices are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing) and must take into account the periodicity of the data if the **-p=** argument is specified.

- 17) The **-mi=missing_value** argument specifies the missing value indicator associated with the NETCDF variables in the *output_netcdf_file*. If the **-mi=** argument is not specified, *missing_value* is set to $1 \cdot e + 20$.
- 18) The **-use_eps=tol** argument is used to determine the effective rank of the linear least squares problem. *tol* must be set to the relative precision of the elements in the NETCDF variables specify by the **-v=** and **-vi=** arguments. If each element is correct to, say, 5 digits then $tol = 0.00001$ should be used. *tol* must not be greater or equal to 1 or less than 0, otherwise an error message is printed and the program stops. If the **-use_eps=** argument is not used, the numerical rank is determined.
- 19) If **-comp_min_norm** is specified, a complete orthogonal factorization of the coefficient matrix and the minimum 2-norm solutions are computed.
- 20) If **-add_mean** is specified, the means are added to the residuals of the regression model. This option has an effect only if **-a=residual** or **-a=all** (e.g., if the residuals are computed and stored in the *output_netcdf_file*).
By default, the means of the residuals in the output NetCDF file are zero.
- 21) If **-rsquare** is specified, the coefficient of determination of the specified model is computed.

- 22) If **-adjrsquare** is specified, the adjusted coefficient of determination of the specified model is computed.
- 23) If **-stderr** is specified, the standard errors of the estimated regression coefficients are computed (unless the specified model is not of full rank).
- 24) If **-ftest** is specified, a F-test is performed to test the null hypothesis that all the regression coefficients are zero (excepted the intercept).
- 25) If **-ttest** is specified, Student-tests are performed to test the null hypothesis that the regression coefficients are zero (independently of the other regression coefficients).
- 26) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 27) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 28) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 29) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 30) The **-tlimited** argument specifies that the time dimension must be defined as limited in the output NetCDF file.

By default, this time dimension is defined as unlimited in the output NetCDF file.

- 31) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 32) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask if the **-m=** argument is used.
- 33) For more details on regression analysis in the climate literature see

- “Fitting nature s basic functions Part I: polynomials and linear least squares”, by Rust, B.W., Computing in Science and Engineering, Vol. 3, no 5, 84-89, 2001. doi: [10.1109/MCISE.2001.947111](https://doi.org/10.1109/MCISE.2001.947111)
- “Fitting nature s basic functions Part II: estimating uncertainties and testing hypotheses”, by Rust, B.W., Computing in Science and Engineering, Vol. 3, no 6, 60-64, 2001. doi: [10.1109/5992.963429](https://doi.org/10.1109/5992.963429)
- “Fitting nature s basic functions Part III: exponentials, sinusoids, and nonlinear least squares”, by Rust, B.W., Computing in Science and Engineering, Vol. 4, no 4, 72-77, 2002. doi: [10.1109/MCISE.2002.1014982](https://doi.org/10.1109/MCISE.2002.1014982)
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: [9780521012300](https://www.isbn-international.org/number/9780521012300)

Outputs

`comp_reg_3d` creates an output NetCDF file that contains the regression statistics and statistical tests associated with these coefficients, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument.

If the `-vi=index_netcdf_variable` is specified, the output NetCDF data set will have *periodicity* time observations and may contain the following NetCDF variables depending on the arguments used in calling the procedure (in the description below, `nlat` and `nlon` are the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_index_netcdf_variable_reg0(periodicity,nlat,nlon)` : the intercept coefficients in the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 2) `netcdf_variable_index_netcdf_variable_reg1(periodicity,nlat,nlon)` : the regression coefficients between each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the `index_netcdf_variable` time series.

By default, the regression coefficients are expressed in units of the input NetCDF variable `netcdf_variable` by unit of the `index_netcdf_variable` time series.

- 3) `netcdf_variable_index_netcdf_variable_stderr0(periodicity,nlat,nlon)` : the standard-errors of the intercept coefficients in the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 4) `netcdf_variable_index_netcdf_variable_stderr1(periodicity,nlat,nlon)` : the standard-errors of the regression coefficients between each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the `index_netcdf_variable` time series.
- 5) `netcdf_variable_index_netcdf_variable_tprob0(periodicity,nlat,nlon)` : the Student t-test probabilities associated with the intercept coefficients in the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 6) `netcdf_variable_index_netcdf_variable_tprob1(periodicity,nlat,nlon)` : the Student t-test probabilities associated with the regression coefficients between each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the `index_netcdf_variable` time series.
- 7) `netcdf_variable_index_netcdf_variable_dv(periodicity,nlat,nlon)` : the regression coefficients between each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the `-dv=` argument has been specified when calling `comp_reg_3d`.

- 8) `netcdf_variable_index_netcdf_variable_dv_stderr(periodicity,nlat,nlon)` : the standard-errors of the regression coefficients between each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the `-dv=` argument has been specified when calling `comp_reg_3d`.

- 9) `netcdf_variable_index_netcdf_variable_dv_tprob(periodicity,nlat,nlon)` : the Student t-test probabilities associated with the regression coefficients between each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the **-dv=** argument has been specified when calling `comp_reg_3d`.

- 10) `netcdf_variable_index_netcdf_variable_r2(periodicity,nlat,nlon)` : the r-square statistics associated with the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 11) `netcdf_variable_index_netcdf_variable_adj_r2(periodicity,nlat,nlon)` : the adjusted r-square statistics associated with the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 12) `netcdf_variable_index_netcdf_variable_fprob(periodicity,nlat,nlon)` : the F-test probabilities associated with the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 13) `netcdf_variable_index_netcdf_variable_predict(ntime,nlat,nlon)` : the predictions associated with the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 14) `netcdf_variable_index_netcdf_variable_resid(ntime,nlat,nlon)` : the residuals associated with the regression models for predicting each grid-point in the time series of the 2-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.

All these statistics are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable`, even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

If the **-vi=index_netcdf_variable** is not specified when calling the procedure, similar statistics will be produced and each component of the selected polynomial trend regression model will have regression coefficients, standard-errors and Student t-test probabilities associated with it. As an illustration, the intercept and the regression coefficients (e.g., the slope) in a linear trend regression model will be stored in NetCDF variables `netcdf_variable_poly_trend_reg0` and `netcdf_variable_poly_trend_reg1`, respectively. For a quadratic trend regression model, in addition to these variables, the regression coefficients associated with the quadratic component will be stored in a NetCDF variable `netcdf_variable_poly_trend_reg2`. The same naming conventions are used for the standard-errors and Student t-test probabilities associated with each component of the selected polynomial trend regression model.

Examples

- 1) For linear detrending bimonthly data from a tridimensional NetCDF variable `mslp` in the NetCDF file `mslp.seas.mean.nc` and store the results in a NetCDF file named `reg_mslp_seas_ncep2.nc`, use the following command (note that cyclostationarity is assumed for the `mslp` variable since **-p=6** is specified) :

```
$ comp_reg_3d \
-f=mslp.seas.mean.nc \
-v=mslp \
-m=mesh_mask_mslp_ncep2.nc \
-p=6 \
-a=residual \
-o=reg_mslp_seas_ncep2.nc
```

- 2) For computing bimonthly lag regressions and residuals from a tridimensional NetCDF variable `mslp` in the NetCDF file `mslp.seas.mean.nc` and a February-March Nino34 SST index in the NetCDF file `sst_nino34_7901_seas.nc` and store the results in a NetCDF file named

reg_mslp_seas_ncep2_nino34_23.nc, use the following command (note that cyclostationarity is assumed for both the mslp variable since **-p=6** is specified, and the index variable since **-pi=6, 1**) :

```
$ comp_reg_3d \  
-f=mslp.seas.mean.nc \  
-v=mslp \  
-m=mesh_mask_mslp_ncep2.nc \  
-p=6 \  
-fi=sst_nino34_7901_seas.nc \  
-vi=sst \  
-pi=6,1 \  
-a=residual \  
-o=reg_mslp_seas_ncep2_nino34_23.nc
```

2.46 comp_reg_4d

2.46.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.46.2 Latest revision

29/04/2026

2.46.3 Purpose

Estimate polynomial trends and regression models from time series extracted from a fourdimensional variable in a NetCDF dataset and, optionally, remove the linear terms from the data by using linear least squares estimation and store the residuals and/or the predictions in a NetCDF dataset.

Optionally, regression diagnostics and statistical tests to diagnose the quality of the fitted regression models may also be computed and stored [[Rusta](#)] [[Rustb](#)] .

Finally, if the NetCDF variable is tridimensional use [comp_reg_3d](#) instead of comp_reg_4d and if the NetCDF variable is unidimensional use [comp_reg_1d](#) instead of comp_reg_4d.

This procedure is parallelized if OpenMP is used.

2.46.4 Further Details

Usage

```
$ comp_reg_4d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-m=input_mesh_mask_netcdf_file (optional) \  
-g=grid_type (optional : n, t, u, v, w, f) \  
-x=lon1,lon2 (optional) \  
-y=lat1,lat2 (optional) \  
-z=level1,level2 (optional) \  
-t=time1,time2 (optional) \  

```

(continues on next page)

(continued from previous page)

```

-p=periodicity                (optional) \
-a=type_of_analysis           (optional : reg, residual, predict, all) \
-o=output_netcdf_file         (optional) \
-to=time_origin               (optional) \
-dg=polynomial_degree         (optional) \
-fi=input_index_netcdf_file   (optional) \
-vi=index_netcdf_variable     (optional) \
-ti=itime1,itime2             (optional) \
-pi=iperiodicity,istep        (optional) \
-ni=index_for_2d_index_netcdf_variable (optional) \
-sm=smoothing_factor          (optional) \
-dv=dv_time1,dv_time2        (optional) \
-mi=missing_value            (optional) \
-use_eps=tol                  (optional) \
-comp_min_norm                (optional) \
-add_mean                     (optional) \
-rsquare                      (optional) \
-adjrsquare                   (optional) \
-stderr                       (optional) \
-ftest                        (optional) \
-ttest                        (optional) \
-double                       (optional) \
-bigfile                      (optional) \
-hdf5                         (optional) \
-cdf5                         (optional) \
-tlimited                      (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input *netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- p=** the *periodicity* is set to 1
- a=** *type_of_analysis* is set to *reg*
- o=** *output_netcdf_file* is set to *reg_netcdf_variable.index_netcdf_variable.nc* if the **-vi=** argument is used and to *reg_netcdf_variable.poly_trend.nc* otherwise
- to=** If a polynomial trend is estimated, the origin for the time scale is set to 0
- dg=** the *polynomial_degree* is set 1. this means that a linear trend is estimated if the **-vi=** argument is not used
- fi=** a polynomial trend is estimated if the **-vi=** argument is not used, otherwise the **-fi=** argument may be used to specified the NetCDF dataset for extracting the *index_netcdf_variable*
- vi=** a polynomial trend is estimated if the **-vi=** argument is not used

- ti=** the whole time period associated with the *index_netcdf_variable* if the **-vi=index_netcdf_variable** is specified.
- pi=** this parameter is not used
- ni=** if the *index_netcdf_variable* is bidimensional, the first time series is used
- sm=** no smoothing is applied to the *index_netcdf_variable* if the **-vi=** argument is used
- dv=** a dummy variable is not used
- mi=** the *missing_value* is set to $1.e+20$ for the NetCDF variables in the *output_netcdf_file*
- use_eps=** no tolerance is used for solving the linear least square problem associated with the specified regression model. If **-use_eps=** is used, the specified tolerance *tol* is used for solving the linear least square problem
- comp_min_norm** the minimal norm solution is not computed. If **-comp_min_norm** is activated, the minimal norm solution of the linear least square problem is computed
- add_mean** the means are not added to the residuals. If **-add_mean** is activated, the means are added to the residuals
- rsquare** the coefficient of determination is not computed. If **-rsquare** is activated, the coefficient of determination is computed
- adjrsquare** the adjusted coefficient of determination is not computed. If **-adjrsquare** is activated, the adjusted coefficient of determination is computed
- stderr** the standard errors of the estimates are not computed. If **-stderr** is activated, the standard errors of the estimates are computed
- ftest** a F-test is not computed. If **-ftest** is activated, a F-test is computed
- ttest** Student-tests are not computed. If **-ttest** is activated, the Student-tests are computed
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a regression analysis must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to *netcdf_variable* for transforming this fourdimensional NetCDF variable as a rectangular matrix of observed variables before computing the regression analysis. By default, it is assumed that each cell in the 3-D grid-mesh associated with the input fourdimensional NetCDF variable is a valid time series (e.g., missing values are not present).

The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to [comp_clim_4d](#) or [comp_mask_4d](#) for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using [comp_reg_4d](#).

- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f*, it is assumed that the input NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used.
- 4) If the **-x=lon1,lon2** , **-y=lat1,lat2** **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon + lon1 + 1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_reg_4d](#).

- 5) If the **-t=time1,time2** argument is missing, data in the whole time period associated with the *netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

The selected time period (e.g., $time2 - time1 + 1$) must be a whole multiple of the *periodicity* if the **-p=** argument is specified.

- 6) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc.

If the **-p=periodicity** argument is specified, the regression models are computed by taking into account the periodicity of the data. This means that *periodicity* regression models are estimated for each cell in the 3-D grid-mesh associated with the input fourdimensional NetCDF variable.

- 7) The **-a=**argument specifies the statistics which must be computed and if the residuals from the trend or regression models are stored in the output NetCDF file. If:
 - **-a=reg**, the default, the residuals or predictions are not computed and only the regression and intercept coefficients are computed and stored
 - **-a=residual**, the residuals are computed and stored, in addition of the regression and intercept coefficients
 - **-a=predict**, the predictions are computed and stored, in addition of the regression and intercept coefficients
 - **-a=all**, the residuals and predictions are computed and stored, in addition of the regression and intercept coefficients.

- 8) The **-to=time_origin** argument specifies the origin for the time scale if a polynomial regression model is used (e.g., when the **-vi=** argument is not used).

By default, the origin for the time scale is set to 0. This shift of the zero point for the time scale makes the estimate for the intercept(s) in the polynomial model equal to the model prediction(s) for the first observation at the beginning of the record.

- 9) The **-dg=polynomial_degree** argument specifies the degree of the polynomial if a polynomial trend regression model is used.

if **-dg=1** a linear trend is used, if **-dg=2** a quadratic trend is used, etc.

polynomial_degree must be a strictly positive integer.

- 10) The **-vi=index_netcdf_variable** specifies a time series for the regression model (e.g., an independent variable).

If the **-vi=index_netcdf_variable** is present, the **-fi=** argument must also be present and this argument specifies the NetCDF dataset which contains the *index_netcdf_variable*. However, if the NetCDF dataset, which contains the *index_netcdf_variable*, is the same as the NetCDF dataset specified by the **-f=** argument, it is not necessary to specify the **-fi=** argument.

If the **-vi=index_netcdf_variable** is not specified a regression model with a polynomial trend is assumed and the **-dg=** argument specifies the degree of the polynomial : if **-dg=1** a linear trend is used, if **-dg=2** a quadratic trend is used, etc.....

- 11) If the **-ti=itime1,itime2** argument is missing, data in the whole time period associated with the *index_netcdf_variable* is taken into account. The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. If the **-vi=** argument is not present, this argument is not used.
- 12) The selected time periods for the *netcdf_variable* and *index_netcdf_variable* must agree. This means that the following equality must be verified

$$(\text{time2} - \text{time1} + 1) / \text{periodicity} = \text{ceiling}((\text{itime2} - \text{itime1} - \text{istep} + 2) / \text{iperiodicity}),$$

otherwise, an error message will be issued and the program will stop.

- 13) The **-pi=** argument gives the periodicity and selects the time step for the *index_netcdf_variable*. For example, to compute regression models with the January monthly time series extracted from the *index_netcdf_variable*, which is assumed to be sampled every month, **-pi=12, 1** should be specified, with yearly data **-pi=1, 1** may be used, etc.

If the **-vi=** argument is not present, this argument is not used.

- 14) The **-ni=** argument specifies the index (e.g., an integer) for selecting the time series if the *index_netcdf_variable* specified in the **-vi=** argument is a 2D NetCDF variable.
- 15) The **-sm=smoothing_factor** means that the time series associated with the *index_netcdf_variable* (e.g., the **-vi=** argument) must be smoothed with a moving average of approximately $2 \cdot \text{smoothing_factor} + 1$ terms before estimating the regression parameters for predicting the *netcdf_variable* (e.g., the **-v=** argument) from the *index_netcdf_variable*. *smoothing_factor* must be a strictly positive integer.

If the **-vi=** argument is not present, this argument is not used.

- 16) If the **-dv=dv_time1,dv_time2** argument is specified, a dummy variable is also included in the regression model. The dummy variable is an absence/presence variable (e.g., with values 0 or 1) and the time observations where the dummy variable is equal to 1 is specified by the *dv_time1* and *dv_time2* integers. The *dv_time1* and *dv_time2* integers specify the first and last time observations of the selected time period in which the dummy variable is set to 1.

These time indices are counted from the start of the (selected) time period (e.g., *time1* in the **-t=time1,time2** argument or 1 if this argument is missing) and must take into account the periodicity of the data if the **-p=** argument is specified.

- 17) The **-mi=missing_value** argument specifies the missing value indicator associated with the NETCDF variables in the *output_netcdf_file*. If the **-mi=** argument is not specified *missing_value* is set to 1.e+20.
- 18) The **-use_eps=tol** argument is used to determine the effective rank of the linear least squares problem. *tol* must be set to the relative precision of the elements in the NETCDF variables specify by the **-v=** and **-vi=** arguments. If each element is correct to, say, 5 digits then *tol*=0.00001 should be used. *tol* must not be greater or equal to 1 or less than 0, otherwise an error message is printed and the program stops. If the **-use_eps=** argument is not used, the numerical rank is determined.
- 19) If **-comp_min_norm** is specified, a complete orthogonal factorization of the coefficient matrix and the minimum 2-norm solutions are computed.

- 20) If **-add_mean** is specified, the means are added to the residuals of the regression model. This option has an effect only if **-a=residual** or **-a=all** (e.g., if the residuals are computed and stored in the `output_netcdf_file`).

By default, the means of the residuals in the output NetCDF file are zero.

- 21) If **-rsquare** is specified, the coefficient of determination of the specified model is computed.
- 22) If **-adjrsquare** is specified, the adjusted coefficient of determination of the specified model is computed.
- 23) If **-stderr** is specified, the standard errors of the estimated regression coefficients are computed (unless the specified model is not of full rank).
- 24) If **-ftest** is specified, a F-test is performed to test the null hypothesis that all the regression coefficients are zero (excepted the intercept).
- 25) If **-ttest** is specified, Student-tests are performed to test the null hypothesis that the regression coefficients are zero (independently of the other regression coefficients).
- 26) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 27) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the `output_netcdf_file` will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 28) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 29) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 30) The **-tlimited** argument specifies that the time dimension must be defined as limited in the output NetCDF file.

By default, this time dimension is defined as unlimited in the output NetCDF file.

- 31) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 32) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask if the **-m=** argument is used.

- 33) For more details on regression analysis in the climate literature, see

- “Fitting nature s basic functions Part I: polynomials and linear least squares”, by Rust, B.W., Computing in Science and Engineering, Vol. 3, no 5, 84-89, 2001. doi: [10.1109/MCISE.2001.947111](https://doi.org/10.1109/MCISE.2001.947111)
- “Fitting nature s basic functions Part II: estimating uncertainties and testing hypotheses”, by Rust, B.W., Computing in Science and Engineering, Vol. 3, no 6, 60-64, 2001. doi: [10.1109/5992.963429](https://doi.org/10.1109/5992.963429)

- “Fitting nature s basic functions Part III: exponentials, sinusoids, and nonlinear least squares”, by Rust, B.W., Computing in Science and Engineering, Vol. 4, no 4, 72-77, 2002. doi: [10.1109/MCISE.2002.1014982](https://doi.org/10.1109/MCISE.2002.1014982)
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 8, 484 pp., 2002. ISBN: [9780521012300](https://www.amazon.com/Statistical-Analysis-Climate-Research/dp/0521012300)

Outputs

`comp_reg_4d` creates an output NetCDF file that contains the regression statistics and statistical tests associated with these coefficients, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument.

If the `-vi=index_netcdf_variable` is specified, the output NetCDF data set will have *periodicity* time observations and may contain the following NetCDF variables depending on the arguments used in calling the procedure (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_index_netcdf_variable_reg0` (`periodicity`, `nlev`, `nlat`, `nlon`) : the intercept coefficients in the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 2) `netcdf_variable_index_netcdf_variable_reg1` (`periodicity`, `nlev`, `nlat`, `nlon`) : the regression coefficients between each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the `index_netcdf_variable` time series.

By default, the regression coefficients are expressed in units of the input NetCDF variable `netcdf_variable` by unit of the `index_netcdf_variable` time series.
- 3) `netcdf_variable_index_netcdf_variable_stderr0` (`periodicity`, `nlev`, `nlat`, `nlon`) : the standard-errors of the intercept coefficients in the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 4) `netcdf_variable_index_netcdf_variable_stderr1` (`periodicity`, `nlev`, `nlat`, `nlon`) : the standard-errors of the regression coefficients between each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the `index_netcdf_variable` time series.
- 5) `netcdf_variable_index_netcdf_variable_tprob0` (`periodicity`, `nlev`, `nlat`, `nlon`) : the Student t-test probabilities associated with the intercept coefficients in the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` by the `index_netcdf_variable` time series.
- 6) `netcdf_variable_index_netcdf_variable_tprob1` (`periodicity`, `nlev`, `nlat`, `nlon`) : the Student t-test probabilities associated with the regression coefficients between each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the `index_netcdf_variable` time series.
- 7) `netcdf_variable_index_netcdf_variable_dv` (`periodicity`, `nlev`, `nlat`, `nlon`) : the regression coefficients between each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable `netcdf_variable` and the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the `-dv=` argument has been specified when calling `comp_reg_4d`.

- 8) `netcdf_variable_index_netcdf_variable_dv_stderr` (`periodicity`, `nlev`, `nlat`, `nlon`) : the standard-errors of the regression coefficients between each grid-point in the time series of the 3-

D grid-mesh associated with the input NetCDF variable *netcdf_variable* and the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the **-dv=** argument has been specified when calling `comp_reg_4d`.

- 9) *netcdf_variable_index_netcdf_variable_dv_tprob*(periodicity,nlev,nlat,nlon) : the Student t-test probabilities associated with the regression coefficients between each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable* and the dummy variable time series if a dummy variable is included in the regression models.

This variable is stored only if the **-dv=** argument has been specified when calling `comp_reg_4d`.

- 10) *netcdf_variable_index_netcdf_variable_r2*(periodicity,nlev,nlat,nlon) : the r-square statistics associated with the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.
- 11) *netcdf_variable_index_netcdf_variable_adj2*(periodicity,nlev,nlat,nlon) : the adjusted r-square statistics associated with the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.
- 12) *netcdf_variable_index_netcdf_variable_fprob*(periodicity,nlev,nlat,nlon) : the F-test probabilities associated with the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.
- 13) *netcdf_variable_index_netcdf_variable_predict*(ntime,nlev,nlat,nlon) : the predictions associated with the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.
- 14) *netcdf_variable_index_netcdf_variable_resid*(ntime,nlev,nlat,nlon) : the residuals associated with the regression models for predicting each grid-point in the time series of the 3-D grid-mesh associated with the input NetCDF variable *netcdf_variable* by the *index_netcdf_variable* time series.

All these statistics are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable*, even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

If the **-vi=index_netcdf_variable** is not specified when calling the procedure, similar statistics will be produced and each component of the selected polynomial trend regression model will have regression coefficients, standard-errors and Student t-test probabilities associated with it. As an illustration, the intercept and the regression coefficients (e.g., the slope) in a linear trend regression model will be stored in NetCDF variables *netcdf_variable_poly_trend_reg0* and *netcdf_variable_poly_trend_reg1*, respectively. For a quadratic trend regression model, in addition to these variables, the regression coefficients associated with the quadratic component will be stored in a NetCDF variable *netcdf_variable_poly_trend_reg2*. The same naming conventions are used for the standard-errors and Student t-test probabilities associated with each component of the selected polynomial trend regression model.

Examples

- 1) For quadratic detrending bimonthly data from a fourdimensional NetCDF variable *uwind* in the NetCDF file *uwind.seas.mean.nc* and store the results in a NetCDF file named *reg_uwind_seas_ncep2.nc*, use the following command (note that cyclostationarity is assumed for the *uwind* variable since **-p=6** is specified and that computations are done only for levels 1 and 3) :

```
$ comp_reg_4d \  
-f=uwind.seas.mean.nc \  
-v=uwind \  
-z=1,3 \  
-p=6 \  
-dg=2 \  
-m=mesh_mask_wind_ncep2.nc \  
-a=residual \  
-o=reg_uwind_seas_ncep2.nc
```

2.47 comp_season_3d

2.47.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.47.2 Latest revision

30/04/2026

2.47.3 Purpose

Compute time (i.e., seasonal, daily, ...) averages from a tridimensional variable extracted from a NetCDF dataset and write the results in an output NetCDF dataset.

The time averages are computed for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable and the time series are assumed to be strictly periodic with a periodicity given by the **-p=** argument (this means, in particular, that `comp_season_3d` cannot be used to compute monthly time series from observed daily data). Note that the date information associated with each time average in the output NetCDF dataset is the date of the first contributing input observation in each time average (if this date information is available in the input NetCDF dataset).

If your data contains missing values (excepted those associated with a constant land-sea mask) use [*comp_season_miss_3d*](#) instead of `comp_season_3d` to estimate time averages from your gappy dataset.

Finally, if the NetCDF variable is fourdimensional use [*comp_season_4d*](#) instead of `comp_season_3d`.

2.47.4 Further Details

Usage

```
$ comp_season_3d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-p=periodicity_of_data \  
-s=length_of_season \  
-o=output_seasonal_netcdf_file \  
-x=lon1,lon2 (optional) \  
-y=lat1,lat2 (optional) \  
-t=time1,time2 (optional) \  

```

(continues on next page)

(continued from previous page)

```

-m=input_mesh_mask_netcdf_file      (optional) \
-g=grid_type                        (optional : n, t, u, v, w, f) \
-compact                            (optional) \
-bigfile                           (optional) \
-hdf5                               (optional) \
-cdf5                               (optional) \
-tlimited                            (optional)

```

By default

- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- m=** an *input_mesh_mask_netcdf_file* is not used. In this case, it is assumed that the *netcdf_variable* does not have any missing values
- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- compact** the output NetCDF dataset is not compacted
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be averaged and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity_of_data** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified.

The **-p=periodicity_of_data** argument is a strictly positive integer.

- 3) The **-s=length_of_season** argument gives the length of the season or the time interval for averaging the input data (i.e., the number of time observations used to compute the time averages in the output NetCDF file).

The specified number for the **-s=** argument should be an exact divisor of the number specified in the **-p=** argument. For example, to construct a seasonal (averages of four months) dataset from a monthly dataset, **-p=12** and **-s=4** should be specified.

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used to construct the output NetCDF file.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_season_3d*.

- 5) If the **-t=***time1,time2* argument is missing the whole time period associated with the *netcdf_variable* is used to construct the output NetCDF file. The selected time period is not necessarily a multiple of the number of observations used to compute the time averages (e.g., the **-s=** argument). If this is not the case, the last time average is computed with less observations (i.e., less observations than the number specified in the **-s=** argument).
- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

The **-g=** argument is used only if a mesh-mask NetCDF dataset is specified with the **-m=** argument. This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument.

- 7) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 8) It is assumed that the data have no missing values, excepted those associated with a constant land-sea mask (if a mesh-mask NetCDF file is used). If it is the case, use *comp_season_miss_3d* instead of *comp_season_3d*.
- 9) If the **-compact** argument is specified and a domain is selected (with the **-x=** and **-y=** arguments) then only the data for the selected domain will be written in the output NetCDF file.

By default, the whole grid is stored (with missing values outside the selected domain) in the output NetCDF file.

- 10) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 11) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 12) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 13) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

comp_season_3d creates an output NetCDF file that contains time averages computed from the time series associated with the input NetCDF variable and the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file*. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, *nlat* and *nlon* are the lengths of the

spatial dimensions of the input NetCDF variable) and *nseason* time observations (corresponding to the number of time averages which have been computed from the input time series) :

- 1) *netcdf_variable* (*nseason*, *nlat*, *nlon*) : the computed time averages for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

By default, the whole grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used, the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=** and **-y=** arguments (e.g., in this case *nlat*=*lat2-lat1*+1 and *nlon*=*lon2-lon1*+1).

The number of time steps written in the output NetCDF file (e.g., *ntime*) is determined from the **-t=**, **-s=** and **-p=** arguments.

Examples

- 1) For computing seasonal averages (i.e., time averages of 4 months) from a monthly tridimensional NetCDF variable *sosstsst* in the NetCDF file *ST7_1m_00101_20012_grid_T_sosstsst.nc* and store the results in a NetCDF file named *ST7_4m_00101_20012_grid_T_sosstsst.nc* use the following command (note that cyclostationarity is assumed for the *sosstsst* variable since **-p=12** is specified) :

```
$ comp_season_3d \
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \
-v=sosstsst \
-p=12 \
-s=4 \
-m=mask_orca2.nc \
-o=sST7_4m_0101_20012_grid_T_sosstsst.nc
```

2.48 comp_season_4d

2.48.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.48.2 Latest revision

30/04/2026

2.48.3 Purpose

Compute time (i.e., seasonal, daily, ...) averages from a fourdimensional variable extracted from a NetCDF dataset and write the results in an output NetCDF dataset.

The time averages are computed for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable and the time series are assumed to be strictly periodic with a periodicity given by the **-p=** argument (this means, in particular, that *comp_season_4d* cannot be used to compute monthly time series from observed daily data). Note that the date information associated with each time average in the output NetCDF dataset is the date of the first contributing input observation in each time average (if this date information is available in the input NetCDF dataset).

If your data contains missing values (excepted those associated with a constant land-sea mask) use *comp_season_miss_4d* instead of *comp_season_4d* to estimate time averages from your gappy dataset.

Finally, if the NetCDF variable is tridimensional use *comp_season_3d* instead of *comp_season_4d*.

2.48.4 Further Details

Usage

```
$ comp_season_4d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-p=periodicity_of_data \  
-s=length_of_season \  
-o=output_seasonal_netcdf_file \  
-x=lon1,lon2 (optional) \  
-y=lat1,lat2 (optional) \  
-z=level1,level2 (optional) \  
-t=time1,time2 (optional) \  
-m=input_mesh_mask_netcdf_file (optional) \  
-g=grid_type (optional : n, t, u, v, w, f) \  
-compact (optional) \  
-bigfile (optional) \  
-hdf5 (optional) \  
-cdf5 (optional) \  
-tlimited (optional)
```

By default

- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- m=** an *input_mesh_mask_netcdf_file* is not used. In this case, it is assumed that the *netcdf_variable* does not have any missing values
- g=** the *grid_type* is set to *n*, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- compact** the output NetCDF dataset is not compacted
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be averaged and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified.
The **-p=periodicity_of_data** argument is a strictly positive integer.
- 3) The **-s=length_of_season** argument gives the length of the season or the time interval for averaging the input data (i.e., the number of time observations used to compute the time averages in the output NetCDF file).
The specified number for the **-s=** argument should be an exact divisor of the number specified in the **-p=** argument. For example, to construct a seasonal (averages of four months) dataset from a monthly dataset, **-p=12** and **-s=4** should be specified.
- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used.
The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.
Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_season_4d*.
- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to construct the output NetCDF file. The selected time period is not necessarily a multiple of the number of observations used to compute the time averages (e.g., the **-s=** argument). If this is not the case, the last time average is computed with less observations (i.e., less observations than the number specified in the **-s=** argument).
- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).
The **-g=** argument is used only if a mesh-mask NetCDF dataset is specified with the **-m=** argument. This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument.
- 7) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 8) It is assumed that the data has no missing values, excepted those associated with a constant land-sea mask (if a mesh-mask NetCDF file is used).
- 9) If the **-compact** argument is specified and a domain is selected (with the **-x=**, **-y=** and **-z=** arguments) then only data for the selected domain will be output. By default, the whole 3-D grid is stored (with missing values outside the selected domain) in the output NetCDF file.
- 10) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.
If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 11) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 12) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 13) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_season_4d` creates an output NetCDF file that contains time averages computed from the time series associated with the input NetCDF variable and the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file*. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable) and *nseason* time observations (corresponding to the number of time averages which have been computed from the input time series):

- 1) *netcdf_variable* (`nseason`, `nlev`, `nlat`, `nlon`): the computed time averages for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.

By default, the whole grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=**, **-y=** and **-z=** arguments (e.g., in this case `nlev=level2-level1+1`, `nlat=lat2-lat1+1` and `nlon=lon2-lon1+1`). The number of time steps written in the output NetCDF file (e.g., `ntime`) is determined from the **-t=**, **-s=** and **-p=** arguments.

Examples

- 1) For computing seasonal averages (i.e., time averages of 4 months) from a monthly fourdimensional NetCDF variable `votemper` in the NetCDF file `ST7_1m_00101_20012_grid_T_votemper.nc` and store the results in a NetCDF file named `ST7_4m_00101_20012_grid_T_votemper.nc` use the following command (note that cyclostationarity is assumed for the `votemper` variable since **-p=12** is specified):

```
$ comp_season_4d \  
-f=ST7_1m_0101_20012_grid_T_votemper.nc \  
-v=votemper \  
-p=12 \  
-s=4 \  
-m=mask_orca2.nc \  
-o=sST7_4m_0101_20012_grid_T_votemper.nc
```

2.49 comp_season_miss_3d

2.49.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.49.2 Latest revision

30/04/2026

2.49.3 Purpose

Compute time (i.e., seasonal, daily, ...) averages from a tridimensional variable with missing values extracted from a NetCDF dataset and write the results in an output NetCDF dataset.

It is assumed that the input NetCDF variable has a scalar *missing_value* or *_FillValue* attribute, which allows a proper identification of the missing values in the dataset, otherwise the procedure will exit with an error message.

The time averages are computed for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable and the time series are assumed to be strictly periodic with a periodicity given by the **-p=** argument (this means, in particular, that `comp_season_miss_3d` cannot be used to compute monthly time series from observed daily data). Note that the date information associated with each time average in the output NetCDF dataset is the date of the first contributing input observation in each time average (if this date information is available in the input NetCDF dataset).

If your data does not contain missing values (in addition to those associated with a constant land-sea mask) use `comp_season_3d` instead of `comp_season_miss_3d` to estimate time averages from your dataset.

Finally, if the NetCDF variable is fourdimensional use `comp_season_miss_4d` instead of `comp_season_miss_4d`.

2.49.4 Further Details

Usage

```
$ comp_season_miss_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -p=periodicity_of_data \
  -s=length_of_season \
  -o=output_seasonal_netcdf_file \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -t=time1,time2              (optional) \
  -m=input_mesh_mask_netcdf_file (optional) \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -ns=nobs_limit_by_season    (optional) \
  -compact                    (optional) \
  -bigfile                     (optional) \
  -hdf5                        (optional) \
  -cdf5                        (optional) \
  -tlimited                     (optional)
```

By default

- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- m=** an *input_mesh_mask_netcdf_file* is not used

- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- ns=** the *nobs_limit_by_season* is set to 1, which means that a time average is set to missing only if all time observations averaged are missing
- compact** the output NetCDF dataset is not compacted
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be averaged and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity_of_data** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified.
The **-p=periodicity_of_data** argument is a strictly positive integer.
- 3) The **-s=length_of_season** argument gives the length of the season or the time interval for averaging the input data (i.e., the number of time observations used to compute the time averages in the output NetCDF file).
The specified number for the **-s=** argument should be an exact divisor of the number specified in the **-p=** argument. For example, to construct a seasonal (averages of four months) dataset from a monthly dataset, **-p=12** and **-s=4** should be specified.
- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used to construct the output NetCDF file.
The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.
Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using [comp_season_miss_3d](#).
- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to construct the output NetCDF file. The selected time period is not necessarily a multiple of the number of observations used to compute the time averages (e.g., the **-s=** argument). If this is not the case, the last time average is computed with less observations (i.e., less observations than the number specified in the **-s=** argument).
- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO (R2, R4 or R05 resolutions) model.
The **-g=** argument is used only if a mesh-mask NetCDF dataset is specified with the **-m=** argument. This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument.
- 7) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.

- 8) It is assumed that the data have missing values in addition to those associated with a constant land-sea mask. If it is not the case, use `comp_season_3d` instead of `comp_season_miss_3d`.

It is further assumed that the specified *netcdf_variable* has a scalar `missing_value` or `_FillValue` attribute and that missing values in the data are identified by the value of this `missing_value` or `_FillValue` attribute.

- 9) The `-ns=nobs_limit_by_season` argument specifies a limit for computing a seasonal mean or time average. *nobs_limit_by_season* is a positive integer less than or equal to the length of the season as specified by the `-s=` argument. If the number of observations by season or for a time interval is less than *nobs_limit_by_season*, the corresponding seasonal mean or time average is set to missing in the output NetCDF file. The default value for *nobs_limit_by_season* is 1, which means that a time average is set to missing only if all time observations averaged are missing.
- 10) If the `-compact` argument is specified and a domain is selected (with the `-x=` and `-y=` arguments) then only the data for the selected domain will be written in the output NetCDF file.

By default, the whole grid is stored (with missing values outside the selected domain) in the output NetCDF file.

- 11) The `-bigfile` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 12) The `-hdf5` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 13) The `-cdf5` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 14) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_season_miss_3d` creates an output NetCDF file that contains time averages computed from the time series associated with the input NetCDF variable and the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file*. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the input NetCDF variable) and *nseason* time observations (corresponding to the number of time averages which have been computed from the input time series) :

- 1) *netcdf_variable* (*nseason*, *nlat*, *nlon*) : the computed time averages for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

By default, the whole grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument `-compact` is used, the geographical

dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=** and **-y=** arguments (e.g., in this case `nlat=lat2-lat1+1` and `nlon=lon2-lon1+1`).

The number of time steps written in the output NetCDF file (e.g., `ntime`) is determined from the **-t=**, **-s=** and **-p=** arguments.

Examples

- 1) For computing seasonal averages (i.e., time averages of 4 months) from a monthly tridimensional NetCDF variable `sosstsst` in the NetCDF file `ST7_1m_00101_20012_grid_T_sosstsst.nc` and store the results in a NetCDF file named `ST7_4m_00101_20012_grid_T_sosstsst.nc` use the following command (note that cyclostationarity is assumed for the `sosstsst` variable since **-p=12** is specified) :

```
$ comp_season_miss_3d \  
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \  
-v=sosstsst \  
-p=12 \  
-s=4 \  
-m=mask_orca2.nc \  
-o=sST7_4m_0101_20012_grid_T_sosstsst.nc
```

2.50 comp_season_miss_4d

2.50.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.50.2 Latest revision

30/04/2026

2.50.3 Purpose

Compute time (i.e., seasonal, daily, ...) averages from a fourdimensional variable with missing values extracted from a NetCDF dataset and write the results in an output NetCDF dataset.

It is assumed that the input NetCDF variable has a scalar *missing_value* or *_FillValue* attribute, which allows a proper identification of the missing values in the dataset, otherwise the procedure will exit with an error message.

The time averages are computed for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable and the time series are assumed to be strictly periodic with a periodicity given by the **-p=** argument (this means, in particular, that `comp_season_miss_4d` cannot be used to compute monthly time series from observed daily data). Note that the date information associated with each time average in the output NetCDF dataset is the date of the first contributing input observation in each time average (if this date information is available in the input NetCDF dataset).

If your data does not contain missing values (in addition to those associated with a constant land-sea mask) use *comp_season_4d* instead of `comp_season_miss_4d` to estimate time averages from your dataset.

Finally, if the NetCDF variable is tridimensional use *comp_season_miss_3d* instead of `comp_season_miss_4d`.

2.50.4 Further Details

Usage

```
$ comp_season_miss_4d \
-f=input_netcdf_file \
-v=netcdf_variable \
-p=periodicity_of_data \
-s=length_of_season \
-o=output_seasonal_netcdf_file \
-x=lon1,lon2 (optional) \
-y=lat1,lat2 (optional) \
-z=level1,level2 (optional) \
-t=time1,time2 (optional) \
-m=input_mesh_mask_netcdf_file (optional) \
-g=grid_type (optional : n, t, u, v, w, f) \
-ns=nobs_limit_by_season (optional) \
-compact (optional) \
-bigfile (optional) \
-hdf5 (optional) \
-cdf5 (optional) \
-tlimited (optional)
```

By default

- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n*, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- ns=** the *nobs_limit_by_season* is set to 1, which means that a time average is set to missing only if all time observations averaged are missing
- compact** the output NetCDF dataset is not compacted
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be averaged and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.

- 2) The **-p=periodicity** argument gives the periodicity of the input data for the *netcdf_variable*. For example, with monthly data **-p=12** should be specified.

The **-p=periodicity_of_data** argument is a strictly positive integer.

- 3) The **-s=length_of_season** argument gives the length of the season or the time interval for averaging the input data (i.e., the number of time observations used to compute the time averages in the output NetCDF file).

The specified number for the **-s=** argument should be an exact divisor of the number specified in the **-p=** argument. For example, to construct a seasonal (averages of four months) dataset from a monthly dataset, **-p=12** and **-s=4** should be specified.

- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using [comp_season_miss_4d](#).

- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to construct the output NetCDF file. The selected time period is not necessarily a multiple of the number of observations used to compute the time averages (e.g., the **-s=** argument). If this is not the case, the last time average is computed with less observations (i.e., less observations than the number specified in the **-s=** argument).

- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

The **-g=** argument is used only if a mesh-mask NetCDF dataset is specified with the **-m=** argument. This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument.

- 7) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.

- 8) It is assumed that the data have missing values in addition to those associated with a constant land-sea mask. If it is not the case, use [comp_season_4d](#) instead of [comp_season_miss_4d](#).

It is further assumed that the specified *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the value of this *missing_value* or *_FillValue* attribute.

- 9) The **-ns=nobs_limit_by_season** argument specifies a limit for computing a seasonal mean or time average. *nobs_limit_by_season* is a positive integer less than or equal to the length of the season as specified by the **-s=** argument. If the number of observations by season or for a time interval is less than *nobs_limit_by_season*, the corresponding seasonal mean or time average is set to missing in the output NetCDF file. The default value for *nobs_limit_by_season* is 1, which means that a time average is set to missing only if all time observations averaged are missing.

- 10) If the **-compact** argument is specified and a domain is selected (with the **-x=**, **-y=** and **-z=** arguments) then only data for the selected domain will be output. By default, the whole 3-D grid is stored (with missing values outside the selected domain) in the output NetCDF file.

- 11) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 12) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 13) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 14) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_season_miss_4d` creates an output NetCDF file that contains time averages computed from the time series associated with the input NetCDF variable and the coordinate NetCDF variables of the input NetCDF dataset *input_netcdf_file*. This NetCDF variable will have the same dimensions and name as the input NetCDF variable in the file *input_netcdf_file* (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable) and *nseason* time observations (corresponding to the number of time averages which have been computed from the input time series) :

- 1) *netcdf_variable* (`nseason`, `nlev`, `nlat`, `nlon`) : the computed time averages for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.

By default, the whole grid associated with the input NetCDF variable is stored (with missing values outside the selected domain). Note, however, that if the argument **-compact** is used the geographical dimensions of the output NetCDF variable will be reduced to the selected domain as specified by the **-x=**, **-y=** and **-z=** arguments (e.g., in this case `nlev=level2-level1+1`, `nlat=lat2-lat1+1` and `nlon=lon2-lon1+1`). The number of time steps written in the output NetCDF file (e.g., `ntime`) is determined from the **-t=**, **-s=** and **-p=** arguments.

Examples

- 1) For computing seasonal averages (i.e., time averages of 4 months) from a monthly fourdimensional NetCDF variable `votemper` in the NetCDF file `ST7_1m_00101_20012_grid_T_votemper.nc` and store the results in a NetCDF file named `ST7_4m_00101_20012_grid_T_votemper.nc` use the following command (note that cyclostationarity is assumed for the `votemper` variable since **-p=12** is specified) :

```
$ comp_season_miss_4d \
  -f=ST7_1m_00101_20012_grid_T_votemper.nc \
  -v=votemper \
```

(continues on next page)

(continued from previous page)

```
-p=12 \
-s=4 \
-m=mask_orca2.nc \
-o=sST7_4m_0101_20012_grid_T_votemper.nc
```

2.51 comp_section_3d

2.51.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.51.2 Latest revision

30/04/2026

2.51.3 Purpose

Compute a longitude-time or latitude-time section (see the description of the **-s=** argument below) from a tridimensional variable extracted from a NetCDF dataset.

Different options are available for computing the cross-section from raw data, anomalies after the removal of a climatology or standardized anomalies (see the description of the **-a=** argument below). Different options are also available for averaging the grid-point time series in the selected domain (see the description of the **-d=** argument below).

The computed section is stored in an output NetCDF dataset.

If your data contains missing values, use [comp_section_miss_3d](#) instead of `comp_section_3d` to compute the cross-section from your gappy dataset.

If the NetCDF variable is fourdimensional use [comp_section_4d](#) instead of `comp_section_3d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the cross-section with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.51.4 Further Details

Usage

```
$ comp_section_3d \
-f=input_netcdf_file \
-v=input_netcdf_variable \
-m=input_mesh_mask_netcdf_file \
-o=output_netcdf_file \
-n=output_netcdf_variable \
-s=section                (optional : lon, lat) \
-g=grid_type              (optional : n, t, u, v, w, f) \
-r=resolution            (optional : r2, r4) \
-b=nlon_orca, nlat_orca  (optional) \
-x=lon1,lon2             (optional) \
```

(continues on next page)

(continued from previous page)

```

-y=lat1,lat2                (optional) \
-t=time1,time2              (optional) \
-c=input_climatology_netcdf_file (optional) \
-a=type_of_analysis         (optional : scp, cov, cor) \
-d=type_of_distance         (optional : dist2, ident) \
-sm=smoothing_factor        (optional) \
-mi=missing_value           (optional) \
-3d                          (optional)
-double                     (optional) \
-bigfile                    (optional) \
-hdf5                       (optional) \
-cdf5                       (optional) \
-tlimited                    (optional)

```

By default

- s=** by default a latitude-time section is computed, which is equivalent to use `lon` for the **-s=** argument
- g=** the *grid_type* is set to `n`, which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to `n`) the resolution is assumed to be `r2`
- b=** if **-g=** is not set to `n`, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to `scp`. This means that the section is computed from the raw data
- d=** the *type_of_distance* is set to `dist2`. This means that the section is computed as a weighted average and that the weight associated with each grid-point time series in the selected domain is proportional to the surface associated with it
- sm=** no time smoothing is applied to the section
- mi=** by default, the *missing_value* in the output NetCDF variable is set to `1.e+20`
- 3d** the *output_netcdf_variable* is defined as an bidimensional NetCDF variable. However, if **-3d** is activated, the *output_netcdf_variable* is defined as a tridimensional NetCDF variable, but with one dummy dimension (e.g., with a length equal to 1)
- double** the section is stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the section is stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The `-v=netcdf_variable` argument specifies the NetCDF variable from which the section must be computed and the `-f=input_netcdf_file` argument specifies that this NetCDF variable must be extracted from the NetCDF file, `input_netcdf_file`.
- 2) If the `-x=lon1,lon2` and `-y=lat1,lat2` arguments are missing the whole geographical domain associated with the `netcdf_variable` is used for computing the section. The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for `lon1`. In this case the longitude domain is from `nlon+lon1+1` to `lon2` where `nlon` is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using `comp_section_3d`.

- 3) If the `-t=time1,time2` argument is missing the whole time period associated with the `netcdf_variable` is used for computing the section.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.

- 4) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask. If your dataset has missing values, use [comp_section_miss_3d](#) instead of `comp_section_3d`.
- 5) The `-s=` argument determines if a latitude-time or longitude-time section is computed. If:

- `-s=lon`, a latitude-time section is computed
- `-s=lat`, a longitude-time section is computed.

- 6) If `-g=` is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before computing the section, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If `-g=` is set to `n`, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

- 7) If `-g=` is set to `t`, `u`, `v`, `w` or `f` (e.g., if the NetCDF variable is from an experiment with the NEMO model), the `-r=` argument gives the resolution used. If:

- `-r=r2`, the NetCDF variable is from an experiment with the ORCA R2 configuration
- `-r=r4`, the NetCDF variable is from an experiment with the ORCA R4 configuration.

- 8) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the `-b=` argument.

- 9) The `-a=` argument specifies if the grid-point time series are centered or standardized with an input climatology (specified with the `-c=` argument) before computing the cross-section for the selected domain. If:

- `-a=scp`, the section is computed from the raw data
- `-a=cov`, the section is computed from the anomalies
- `-a=cor`, the section is computed from the standardized anomalies.

- 10) The `input_climatology_netcdf_file` specified with the `-c=` argument is needed only if `-a=cov` or `-a=cor`.

- 11) If `-a=cov` or `-a=cor`, the selected time period must agree with the climatology. This means that the first selected time observation (`time1` if the `-t=` argument is present) must correspond to the first day, month, season of the climatology specified with the `-c=` argument.

- 12) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (*input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 13) The **-d=** argument specifies the weighting method for computing the cross-section. If:
 - **-d=dist2**, the section is computed with the diagonal distance associated with the horizontal 2-D grid-mesh (each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident**, the section is computed with the identity metric (each grid point has the same weight when computing the averaged time series).
- 14) **-sm=smoothing_factor** means that the section must be smoothed in time with a moving average of approximately $2 * \text{smoothing_factor} + 1$ terms. *smoothing_factor* must be an integer greater than 0.
- 15) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contain the computed section in the output NetCDF file, *output_netcdf_file*, specified by the **-o=** argument.
- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *output_netcdf_variable* in the *output_netcdf_file*.
If the **-mi=** argument is not specified, *missing_value* is set to $1.e+20$ in the output NetCDF dataset.
- 17) The **-3d** argument specifies that the latitude-time or longitude-time section must be stored as a tridimensional NetCDF variable with a dummy dimension in the output NetCDF file.
By default, the section is stored as a bidimensional NetCDF variable.
- 18) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.
By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 19) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.
If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 20) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.
If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 21) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.
If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 22) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.

Outputs

`comp_section_3d` creates an output NetCDF file that contains the computed cross-section. The output NetCDF dataset contains the following NetCDF variable (in the description below, `lat3` and `lon3` are, respectively, equal to `lat2 - lat1 + 1` and `lon2 - lon1 + 1` or `lon2 - nlon - lon1 - 1` if `lon1` is negative) :

If `-s=lon` is specified:

- 1) `output_netcdf_variable(ntime, lat3)` : the computed latitude-time section.

Or if `-s=lon` and `-3d` arguments have been specified :

- 1) `output_netcdf_variable(ntime, lat3, 1)` : the computed latitude-time section defined as a tridimensional variable with one dummy dimension.

If `-s=lat` is specified:

- 1) `output_netcdf_variable(ntime, lon3)` : the computed longitude-time section.

Or if `-s=lat` and `-3d` arguments have been specified :

- 1) `output_netcdf_variable(ntime, 1, lon3)` : the computed longitude-time section defined as a tridimensional variable with one dummy dimension.

Examples

- 1) For computing a monthly longitude-time section from the file `ST7_1m_0101_20012_grid_T_sosstsst.nc`, which includes a NetCDF variable `sosstsst`, and store the results in a NetCDF variable named `sst` in the file `section_sst.orca2.nc`, use the following command (note that the cross-section is computed from monthly anomalies since `-a=cov` is specified) :

```
$ comp_section_3d \  
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \  
-v=sosstsst \  
-m=meshmask.orca2.nc \  
-o=section_sst.orca2.nc \  
-n=sst \  
-s=lat \  
-g=t \  
-c=clim_sosstsst_grid_T.nc \  
-a=cov \  
-d=dist2
```

2.52 comp_section_4d

2.52.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.52.2 Latest revision

30/04/2026

2.52.3 Purpose

Compute a longitude-level-time, latitude-level-time or longitude-latitude-time section (see the description of the **-s=** argument below) from a fourdimensional variable extracted from a NetCDF dataset.

Different options are available for computing the cross-section from raw data, anomalies after the removal of a climatology or standardized anomalies (see the description of the **-a=** argument below). Different options are also available for averaging the grid-point time series in the selected domain (see the description of the **-d=** argument below).

The computed cross-section is stored in an output NetCDF dataset.

If the NetCDF variable is tridimensional use `comp_section_3d` instead of `comp_section_4d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP flag. Moreover, this procedure computes the cross-section with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.52.4 Further Details

Usage

```
$ comp_section_4d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -s=section                (optional : lon, lat, dep) \
  -g=grid_type              (optional : n, t, u, v, w, f) \
  -r=resolution            (optional : r2, r4) \
  -b=nlon_orca, nlat_orca, nlevel_orca (optional) \
  -x=lon1, lon2            (optional) \
  -y=lat1, lat2            (optional) \
  -z=level1, level2        (optional) \
  -t=time1, time2          (optional) \
  -c=input_climatology_netcdf_file    (optional) \
  -a=type_of_analysis      (optional : scp, cov, cor) \
  -d=type_of_distance      (optional : dist2, dist3, ident) \
  -sm=smoothing_factor     (optional) \
  -mi=missing_value        (optional) \
  -double                  (optional) \
  -bigfile                 (optional) \
  -hdf5                    (optional) \
  -cdf5                    (optional) \
  -tlimited                 (optional)
```

By default

- s=** by default a latitude-time section is computed, which is equivalent to use `lon` for the **-s=** argument
- g=** the `grid_type` is set to `n`, which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input `netcdf_variable` is from the NEMO model (e.g., if **-g=** argument is not set to `n`) the resolution is assumed to be `r2`

- b=** if **-g=** is not set to `n`, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to `scp`. This means that the section is computed from the raw data
- d=** the *type_of_distance* is set to `dist3`. This means that the section is computed as a weighted average and that the weight associated with each grid-point time series in the selected domain is proportional to the volume or the weight associated with it
- sm=** no time smoothing is applied to the section
- mi=** by default, the *missing_value* in the output NetCDF variable is set to `1.e+20`
- double** the section is stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the section is stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the section must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain associated with the *netcdf_variable* is used for computing the section. The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using `comp_section_4d`.

- 3) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used for computing the section.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.

- 4) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.

- 5) The **-s=** argument determines how the cross-section is computed. If:

- **-s=lon**, a latitude-level-time section is computed

- **-s=lat**, a longitude-level-time section is computed
 - **-s=dep**, a longitude-latitude-time section is computed.
- 6) If **-g=** is set to **t**, **u**, **v**, **w** or **f**, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before computing the section, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variable covers the whole globe.
- If **-g=** is set to **n**, it is assumed that the 3-D grid-mesh is regular or Gaussian and as such has no duplicate points.
- 7) If **-g=** is set to **t**, **u**, **v**, **w** or **f** (e.g., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:
- **-r=r2**, the NetCDF variable is from an experiment with the ORCA R2 configuration
 - **-r=r4**, the NetCDF variable is from an experiment with the ORCA R4 configuration.
- 8) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 9) The **-a=** argument specifies if the grid-point time series are centered or standardized with an input climatology (specified with the **-c=** argument) before computing the cross-section for the selected domain. If:
- **-a=scp**, the section is computed from the raw data
 - **-a=cov**, the section is computed from the anomalies
 - **-a=cor**, the section is computed from the standardized anomalies.
- 10) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.
- 11) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 12) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (*input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 13) The **-d=** argument specifies the weighting method for computing the cross-section. If:
- **-d=dist3**, the section is computed with the diagonal distance associated with the whole 3-D grid-mesh (each grid point is weighted accordingly to the volume or weight associated with it)
 - **-d=dist2**, the section is computed with the diagonal distance associated with the horizontal 2-D grid-mesh (each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident**, the section is computed with the identity metric (each grid point has the same weight when computing the cross-section).
- 14) **-sm=smoothing_factor** means that the section must be smoothed in time with a moving average of approximately $2 * \text{smoothing_factor} + 1$ terms. *smoothing_factor* must be an integer greater than 0.
- 15) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contains the computed section in the output NetCDF file, *output_netcdf_file*, specified by the **-o=** argument.
- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *output_netcdf_variable* in the *output_netcdf_file*.
- If the **-mi=** argument is not specified, *missing_value* is set to $1.e+20$ in the output NetCDF dataset.
- 17) The **-double** argument specify that the results are stored as double-precision floating point numbers in the output NetCDF file.
- By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 18) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.

Outputs

`comp_section_4d` creates an output NetCDF file that contains the computed cross-section. The output NetCDF dataset contains the following NetCDF variable (in the description below, `lat3`, `lon3` and `lev3` are, respectively, equal to `lat2 - lat1 + 1`, `lon2 - lon1 + 1` and `lev2 - lev1 + 1`):

If **-s=lon** is specified:

- 1) *output_netcdf_variable*(`ntime`, `lev3`, `lat3`) : the computed latitude-level-time section.

If **-s=lat** is specified:

- 1) *output_netcdf_variable*(`ntime`, `lev3`, `lon3`) : the computed longitude-level-time section.

If **-s=dep** is specified:

- 1) *output_netcdf_variable*(`ntime`, `lat3`, `lon3`) : the computed longitude-latitude-time section.

Examples

- 1) For computing a monthly longitude-level-time section from the file `ST7_1m_0101_20012_grid_T_votemper.nc`, which includes a NetCDF variable `votemper`, and store the results in a NetCDF variable named `temp` in the file `section_temp.orca2.nc`, use the following command (note that the cross-section is computed from monthly anomalies) :

```
$ comp_section_4d \
-f=ST7_1m_0101_20012_grid_T_votemper.nc \
-v=votemper \
-m=meshmask.orca2.nc \
-o=section_temp.orca2.nc \
-n=temp \
-s=lat \
-g=t \
```

(continues on next page)

(continued from previous page)

```
-c=clim_votemper_grid_T.nc \
-a=cov \
-d=dist3
```

2.53 comp_section_miss_3d

2.53.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.53.2 Latest revision

30/04/2026

2.53.3 Purpose

Compute a longitude-time or latitude-time section (see the description of the **-s=** argument below) from a tridimensional variable with missing values extracted from a NetCDF dataset.

Different options are available for computing the cross-section from raw data, anomalies after the removal of a climatology or standardized anomalies (see the description of the **-a=** argument below). Different options are also available for averaging the grid-point time series in the selected domain (see the description of the **-d=** argument below).

The computed section is stored in an output NetCDF dataset.

If your data has no missing values, excepted for a constant land-sea mask, use [comp_section_3d](#) instead of `comp_section_miss_3d` to compute the cross-section from your dataset.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the cross-section with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.53.4 Further Details

Usage

```
$ comp_section_miss_3d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -s=section                      (optional : lon, lat) \
  -g=grid_type                    (optional : n, t, u, v, w, f) \
  -r=resolution                   (optional : r2, r4) \
  -b=nlon_orca, nlat_orca        (optional) \
  -x=lon1,lon2                   (optional) \
  -y=lat1,lat2                   (optional) \
  -t=time1,time2                 (optional) \
  -c=input_climatology_netcdf_file (optional) \
```

(continues on next page)

(continued from previous page)

-a=type_of_analysis	(optional : scp, cov, cor) \
-d=type_of_distance	(optional : dist2, ident) \
-sm=smoothing_factor	(optional) \
-mi=missing_value	(optional) \
-3d	(optional)
-double	(optional) \
-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- s= by default a latitude-time section is computed, which is equivalent to use lon for the -s= argument
- g= the *grid_type* is set to n, which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r= if the input netcdf_variable is from the NEMO model (e.g., if -g= argument is not set to n) the resolution is assumed to be r2
- b= if -g= is not set to n, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca* are determined from the -r= argument. However, you may override this choice by default with the -b= argument
- x= the whole longitude domain associated with the *netcdf_variable*
- y= the whole latitude domain associated with the *netcdf_variable*
- t= the whole time period associated with the *netcdf_variable*
- a= the *type_of_analysis* is set to scp. This means that the section is computed from the raw data
- d= the *type_of_distance* is set to dist2. This means that the section is computed as a weighted average and that the weight associated with each point time series in the selected domain is proportional to the surface associated with it
- sm= no time smoothing is applied to the section
- mi= by default, the *missing_value* in the output NetCDF variable is set to 1.e+20
- 3d the *output_netcdf_variable* is defined as an bidimensional NetCDF variable. However, if -3d is activated, the *output_netcdf_variable* is defined as a tridimensional NetCDF variable, but with one dummy dimension (e.g., with a length equal to 1)
- double the section is stored as single-precision floating point numbers in the output NetCDF file. If -double is activated, the section is stored as double-precision floating point numbers
- bigfile a NetCDF classical format file is created. If -bigfile is activated, the output NetCDF file is a 64-bit offset format file
- hdf5 a NetCDF classical format file is created. If -hdf5 is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5 a NetCDF classical format file is created. If -cdf5 is activated, the output NetCDF file is a CDF5 format file
- tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if -tlimited is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The `-v=netcdf_variable` argument specifies the NetCDF variable from which the section must be computed and the `-f=input_netcdf_file` argument specifies that this NetCDF variable must be extracted from the NetCDF file, `input_netcdf_file`.
- 2) It is assumed that the specified NetCDF variable has a scalar `missing_value` or `_FillValue` attribute and that missing values in the data are identified by the value of this missing attribute.
- 3) It is assumed that the data has missing values in addition to those associated with a constant land-sea mask. If your dataset has no missing values, use `comp_section_3d` instead of `comp_section_miss_3d`.
- 4) If the `-x=lon1,lon2` and `-y=lat1,lat2` arguments are missing the whole geographical domain associated with the `netcdf_variable` is used for computing the section. The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for `lon1`. In this case the longitude domain is from `nlon+lon1+1` to `lon2` where `nlon` is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to `comp_mask_3d` for transforming geographical coordinates as indices before using `comp_section_miss_3d`.
- 5) If the `-t=time1,time2` argument is missing the whole time period associated with the `netcdf_variable` is used for computing the section.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.
- 6) The `-s=` argument determines if a latitude-time or longitude-time section is computed. If:
 - `-s=lon`, a latitude-time section is computed
 - `-s=lat`, a longitude-time section is computed.
- 7) If `-g=` is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before computing the section, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If `-g=` is set to `n`, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.
- 8) If `-g=` is set to `t`, `u`, `v`, `w` or `f` (e.g., if the NetCDF variable is from an experiment with the NEMO model), the `-r=` argument gives the resolution used. If:
 - `-r=r2`, the NetCDF variable is from an experiment with the ORCA R2 configuration
 - `-r=r4`, the NetCDF variable is from an experiment with the ORCA R4 configuration.
- 9) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the `-b=` argument.
- 10) The `-a=` argument specifies if the grid-point time series are centered or standardized with an input climatology (specified with the `-c=` argument) before computing the cross-section for the selected domain. If:
 - `-a=scp`, the section is computed from the raw data
 - `-a=cov`, the section is computed from the anomalies
 - `-a=cor`, the section is computed from the standardized anomalies.
- 11) The `input_climatology_netcdf_file` specified with the `-c=` argument is needed only if `-a=cov` or `-a=cor`.

- 12) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 13) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (*input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 14) The **-d=** argument specifies the weighting method for computing the section. If:
 - **-d=dist2**, the section is computed with the diagonal distance associated with the horizontal 2-D grid-mesh (each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident**, the section is computed with the identity metric (each grid point has the same weight when computing the averaged time series).
- 15) **-sm=smoothing_factor** means that the section must be smoothed in time with a moving average of approximately $2 * \text{smoothing_factor} + 1$ terms. *smoothing_factor* must be an integer greater than 0. However, if missing values are present in the computed time section, smoothing is not allowed.
- 16) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contains the computed section in the output NetCDF file, *output_netcdf_file*, specified by the **-o=** argument.
- 17) The **-mi=missing_value** argument specifies the missing value indicator associated with the *output_netcdf_variable* in the *output_netcdf_file*.

If the **-mi=** argument is not specified, *missing_value* is set to $1.e+20$ in the output NetCDF dataset.
- 18) The **-3d** argument specifies that the latitude-time or longitude-time section must be stored as a tridimensional NetCDF variable with a dummy dimension in the output NetCDF file.

By default, the section is stored as a bidimensional NetCDF variable.
- 19) The **-double** argument specify that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 20) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations.

Outputs

`comp_section_miss_3d` creates an output NetCDF file that contains the computed cross-section. The output NetCDF dataset contains the following NetCDF variable (in the description below, `lat3` and `lon3` are, respectively, equal to `lat2 - lat1 + 1` and `lon2 - lon1 + 1` or `lon2 - nlon - lon1 - 1` if `lon1` is negative) :

If `-s=lon` is specified:

- 1) `output_netcdf_variable(ntime, lat3)` : the computed latitude-time section.

Or if `-s=lon` and `-3d` arguments have been specified :

- 1) `output_netcdf_variable(ntime, lat3, 1)` : the computed latitude-time section defined as a tridimensional variable with one dummy dimension.

If `-s=lat` is specified:

- 1) `output_netcdf_variable(ntime, lon3)` : the computed longitude-time section.

Or if `-s=lat` and `-3d` arguments have been specified :

- 1) `output_netcdf_variable(ntime, 1, lon3)` : the computed longitude-time section defined as a tridimensional variable with one dummy dimension.

Examples

- 1) For computing a monthly longitude-time section from the file `ST7_1m_0101_20012_grid_T_sosstsst.nc`, which includes a NetCDF variable `sosstsst`, and store the results in a NetCDF variable named `sst` in the file `section_sst.orca2.nc`, use the following command (note that the cross-section is computed from monthly anomalies since `-a=cov` is specified) :

```
$ comp_section_miss_3d \
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \
-v=sosstsst \
-m=meshmask.orca2.nc \
-o=section_sst.orca2.nc \
-n=sst \
-s=lat \
-g=t \
-c=clim_sosstsst_grid_T.nc \
-a=cov \
-d=dist2
```

2.54 comp_serie_3d

2.54.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.54.2 Latest revision

30/04/2026

2.54.3 Purpose

Compute a time series from a tridimensional variable extracted from a NetCDF dataset.

Different options are available for computing the time series from raw data, anomalies after the removal of a climatology or standardized anomalies (see the description of the **-a=** argument below). Different options are also available for averaging the different pointwise time series in the selected domain (see the description of the **-d=** argument below).

The computed time series is stored in an output NetCDF dataset.

If your data contains missing values, use `comp_serie_miss_3d` instead of `comp_serie_3d` to compute the time series from your gappy dataset.

If the NetCDF variable is fourdimensional, use `comp_serie_4d` instead of `comp_serie_3d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the time series with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.54.4 Further Details

Usage

```
$ comp_serie_3d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -r=resolution                (optional : r2, r4) \
  -b=nlon_orca,nlat_orca      (optional) \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -t=time1,time2              (optional) \
  -c=input_climatology_netcdf_file (optional) \
  -a=type_of_analysis          (optional : scp, cov, cor) \
  -d=type_of_distance          (optional : dist2, ident) \
  -sm=smoothing_factor        (optional) \
  -3d                          (optional) \
  -double                      (optional) \
  -hdf5                        (optional) \
  -cdf5                        (optional) \
  -tlimited                     (optional)
```

By default

- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO or ORCA model (eg if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca*, and *nlat_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*

- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *scp*. This means that the averaged time series is computed from the raw data
- d=** the *type_of_distance* is set to *dist2*. This means that the averaged time series is computing as a weighted average and that the weight associated with each point time series in the selected domain is proportional to the surface associated with it
- sm=** no smoothing is applied to the computed time series
- 3d** the *output_netcdf_variable* is defined as an unidimensional NetCDF variable. However, if **-3d** is activated, the *output_netcdf_variable* is defined as an tridimensional NetCDF variable but with two dummy dimensions (e.g., with a length equal to 1)
- double** the time series is stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the time series is stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the time series must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to construct the time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_serie_3d*.

- 3) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 4) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask. If your dataset has missing values, use [comp_serie_miss_3d](#) instead of *comp_serie_3d*.
- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f*, it is assumed that the NetCDF variable is from an experiment with the NEMO or ORCA model. In this case, the duplicate points from the ORCA grid are removed before computing the time series, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

- 6) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (e.g., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:
 - **-r=r2**, the NetCDF variable is from an experiment with the ORCA R2 configuration
 - **-r=r4**, the NetCDF variable is from an experiment with the ORCA R4 configuration.
- 7) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.
- 8) The **-a=** argument specifies if the point time series are centered or standardized with an input climatology (specified with the **-c=** argument) before computing the averaged time series for the selected domain. If:
 - **-a=scp**, the averaged time series is computed from the raw data
 - **-a=cov**, the averaged time series is computed from the anomalies
 - **-a=cor**, the averaged time series is computed from the standardized anomalies.
- 9) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.
- 10) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.
- 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.
- 12) The **-d=** argument specifies the weighting method for computing the averaged time series. If:
 - **-d=dist2**, the averaged time series is computed with the diagonal distance associated with the horizontal 2-D grid-mesh (each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident**, the time series is computed with the identity metric (each grid point has the same weight when computing the averaged time series).
- 13) **-sm=smoothing_factor** means that the averaged time series must be smoothed with a moving average of approximately $2 * \text{smoothing_factor} + 1$ terms. *smoothing_factor* must be an integer greater than 0.
- 14) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contains the computed time series in the output NetCDF file, *output_netcdf_file*, specified by the **-o=output_netcdf_file** argument.
- 15) The **-3d** argument specifies that the averaged time series must be stored as a tridimensional NetCDF variable with two dummy dimensions in the output NetCDF file.

By default, the time series is stored as an unidimensional NetCDF variable.
- 16) The **-double** argument specifies that, the time series is stored as double-precision floating point numbers in the output NetCDF file.

By default, the time series is stored as single-precision floating point numbers in the output NetCDF file.
- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_serie_3d` creates an output NetCDF file that contains the computed time series. The output NetCDF dataset contains the following NetCDF variable (in the description below, *ntime* is the number of time steps selected with the `-t=` argument) :

- 1) *output_netcdf_variable* (*ntime*) : the averaged time series defined as an unidimensional variable.

or if the `-3d` argument has been specified :

- 1) *output_netcdf_variable* (*ntime*, 1, 1) : the averaged time series defined as a tridimensional variable with two dummy dimensions.

Examples

- 1) For computing a monthly time series from the file `HadISST1_1m_187001_200702_sst_reg1m.nc`, which includes a NetCDF variable `sst`, and store the results in a NetCDF variable named `nino34_sst` in the file `HadISST1_1m_187001_200702_nino34sst.nc`, use the following command :

```
$ comp_serie_3d \
-f=HadISST1_1m_187001_200702_sst_reg1m.nc \
-v=sst \
-x=11,60 \
-y=86,95 \
-m=mask_HadISST1_sst.nc \
-n=nino34_sst \
-o=HadISST1_1m_187001_200702_nino34sst.nc
```

- 2) For computing a monthly time series from the file `F31_1m_000101_011012_sosstsst_grid_T.nc`, which includes a NetCDF variable `sosstsst` (from the NEMO model), and store the results in a NetCDF variable `nino34_sosstsst` in the file `F31_1m_000101_011012_nino34sst.nc`, use the following command :

```
$ comp_serie_3d \
-f=F31_1m_000101_011012_sosstsst_grid_T.nc \
-v=sosstsst \
-x=236,335 \
-y=240,260 \
-m=F31_mesh_mask.nc \
-b=722,511 \
-g=t \
-n=nino34_sosstsst \
-o=F31_1m_000101_011012_nino34sst.nc
```

2.55 comp_serie_4d

2.55.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.55.2 Latest revision

30/04/2026

2.55.3 Purpose

Compute a time series from a fourdimensional variable extracted from a NetCDF dataset.

Different options are available for computing the time series from raw data, anomalies after the removal of a climatology or standardized anomalies (see the description of the **-a=** argument below). Different options are also available for averaging the different point time series in the selected domain (see the description of the **-d=** argument below).

The computed time series is stored in an output NetCDF dataset.

If the NetCDF variable is tridimensional, use *comp_serie_3d* instead of *comp_serie_4d*.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the time series with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.55.4 Further Details

Usage

```
$ comp_serie_4d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -r=resolution                (optional : r2, r4) \
  -b=nlon_orca,nlat_orca,nlevel_orca (optional) \
  -x=lon1,lon2                (optional) \
  -y=lat1,lat2                (optional) \
  -z=level1,level2            (optional) \
  -t=time1,time2              (optional) \
  -c=input_climatology_netcdf_file (optional) \
  -a=type_of_analysis          (optional : scp, cov, cor) \
  -d=type_of_distance          (optional : dist2, dist3, ident) \
  -sm=smoothing_factor        (optional) \
  -3d                          (optional) \
  -double                      (optional) \
  -hdf5                        (optional) \
  -cdf5                        (optional) \
  -tlimited                     (optional)
```

By default

- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 3-D grid-mesh, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument by default, *nlon_orca*, *nlat_orca* and *nlevel_orca* are determined from the **-r=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *scp*. This means that the averaged time series is computed from the raw data
- d=** the *type_of_distance* is set to *dist3*. This means that the averaged time series is computing as a weighted average and that the weight associated with each point time series in the selected domain is proportional to the volume or weight associated with it
- sm=** no smoothing is applied to the time series
- 3d** the *output_netcdf_variable* is defined as an unidimensional NetCDF variable. However, if **-3d** is activated, the *output_netcdf_variable* is defined as an tridimensional NetCDF variable but with two dummy dimensions (e.g., with a length equal to 1)
- double** the time series is stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the time series is stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the time series must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used for computing the time series.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_serie_4d*.

- 3) If the **-t=***time1,time2* argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.

- 4) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f`, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before computing the time series, as far as possible, and, in particular, if the 3-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to `n`, it is assumed that the 3-D grid-mesh is regular or Gaussian and as such has no duplicate points.

- 6) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (e.g., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If

- **-r=r2**, the NetCDF variable is from an experiment with the ORCA R2 configuration
- **-r=r4**, the NetCDF variable is from an experiment with the ORCA R4 configuration.

- 7) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

- 8) The **-a=** argument specifies if the point time series are centered or standardized with an input climatology (specified with the **-c=** argument) before computing the averaged time series for the selected domain. If:

- **-a=scp**, the averaged time series is computed from the raw data
- **-a=cov**, the averaged time series is computed from the anomalies
- **-a=cor**, the averaged time series is computed from the standardized anomalies.

- 9) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.

- 10) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.

- 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.

- 12) The **-d=** argument specifies the weighting method for computing the averaged time series. If:

- **-d=dist3**, the averaged time series is computed with the diagonal distance associated with the whole 3-D grid-mesh (each grid point is weighted accordingly to the volume or weight associated with it)
- **-d=dist2**, the averaged time series is computed with the diagonal distance associated with the horizontal 2-D grid-mesh (each grid point is weighted accordingly to the surface associated with it)
- **-d=ident**, the time series is computed with the identity metric (each grid point has the same weight when computing the averaged time series).

- 13) **-sm=smoothing_factor** means that the averaged time series must be smoothed with a moving average of approximately $2 * \text{smoothing_factor} + 1$ terms. *smoothing_factor* must be an integer greater than 0.

- 14) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contains the computed time series in the output NetCDF file, *output_netcdf_file*, specified by the **-o=output_netcdf_file** argument.

- 15) The **-3d** argument specifies that the averaged time series must be stored as a tridimensional NetCDF variable with two dummy dimensions in the output NetCDF file.

By default, the time series is stored as an unidimensional NetCDF variable.

- 16) The **-double** argument specifies that, the time series is stored as double-precision floating point numbers in the output NetCDF file.

By default, the time series is stored as single-precision floating point numbers in the output NetCDF file.

- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_serie_4d` creates an output NetCDF file that contains the computed time series. The output NetCDF dataset contains the following NetCDF variable (in the description below, `ntime` is the number of time steps selected with the **-t=** argument) :

- 1) *output_netcdf_variable*(`ntime`) : the averaged time series defined as an unidimensional variable.

or if the **-3d** argument has been specified :

- 1) *output_netcdf_variable*(`ntime, 1, 1`) : the averaged time series defined as a tridimensional variable with two dummy dimensions.

Examples

- 1) For computing a monthly anomaly time series from the file `ST7_1m_0101_20012_grid_T_votemper.nc`, which includes a NetCDF variable `votemper` (from the NEMO model), and store the results in a NetCDF variable `pacific_votemper` in the file `pacific_votemper.orca2.nc`, use the following command :

```
$ comp_serie_4d \
  -f=ST7_1m_0101_20012_grid_T_votemper.nc \
  -v=votemper \
  -m=meshmask.orca2.nc \
  -o=pacific_votemper.orca2.nc \
  -n=pacific_votemper \
  -g=t \
  -c=clim_grid_T_votemper.nc \
  -a=cov \
  -d=dist3
```

2.56 comp_serie_miss_3d

2.56.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.56.2 Latest revision

30/04/2026

2.56.3 Purpose

Compute a time series from a tridimensional variable with missing values extracted from a NetCDF dataset.

Different options are available for computing the time series from raw data, anomalies after the removal of a climatology or standardized anomalies (see the description of the **-a=** argument below). Different options are also available for averaging the different point time series in the selected domain (see the description of the **-d=** argument below).

The computed time series is stored in an output NetCDF dataset.

If your data does not contain missing values, use *comp_serie_3d* instead of *comp_serie_miss_3d* to compute the time series from your dataset.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. Moreover, this procedure computes the time series with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

2.56.4 Further Details

Usage

```
$ comp_serie_miss_3d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -n=output_netcdf_variable \
  -g=grid_type                (optional : n, t, u, v, w, f) \
  -r=resolution                (optional : r2, r4) \
  -b=nlon_orca, nlat_orca      (optional) \
  -x=lon1,lon2                 (optional) \
  -y=lat1,lat2                 (optional) \
  -t=time1,time2               (optional) \
  -c=input_climatology_netcdf_file (optional) \
  -a=type_of_analysis          (optional : scp, cov, cor) \
  -d=type_of_distance          (optional : dist2, ident) \
  -sm=smoothing_factor         (optional) \
  -mi=missing_value            (optional) \
  -3d                          (optional) \
  -double                      (optional) \
  -hdf5                        (optional) \
  -cdf5                        (optional) \
  -tlimited                     (optional)
```


By default

- g=** the *grid_type* is set to `n` which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- r=** if the input *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to `n`) the resolution is assumed to be `r2`
- b=** if **-g=** is not set to `n`, the dimensions of the 2-D grid-mesh, *nlon_orca*, and *nlat_orca* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to `scp`. This means that the averaged time series is computed from the raw data
- d=** the *type_of_distance* is set to `dist2`. This means that the averaged time series is computing as a weighted average and that the weight associated with each point time series in the selected domain is proportional to the surface associated with it
- sm=** no smoothing is applied to the computed time series
- mi=** the *missing_value* attribute for the *output_netcdf_variable* is set to `1.e+20`
- 3d** the *output_netcdf_variable* is defined as an unidimensional NetCDF variable. However, if **-3d** is activated, the *output_netcdf_variable* is defined as an tridimensional NetCDF variable but with two dummy dimensions (e.g., with a length equal to 1)
- double** the time series is stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the time series is stored as double-precision floating point numbers by default, the results are stored as single-precision floating point numbers in the output NetCDF file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the time series must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to compute the time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using `comp_serie_miss_3d`.

- 3) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 4) It is assumed that the data has missing values in addition to those associated with a constant land-sea mask. If your dataset has no missing values, use *comp_serie_3d* instead of *comp_serie_miss_3d*.
- 5) It is assumed that the specified input *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the value of this missing attribute.
- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f*, it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed before computing the time series, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

- 7) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:

- **-r=r2**, the NetCDF variable is from an experiment with the ORCA R2 configuration
- **-r=r4**, the NetCDF variable is from an experiment with the ORCA R4 configuration.

- 8) If the NetCDF variable is from an experiment with the NEMO model, but the resolution is not R2 or R4, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

- 9) The **-a=** argument specifies if the point time series are centered or standardized with an input climatology (specified with the **-c=** argument) before computing the averaged time series for the selected domain. If:

- **-a=scp**, the averaged time series is computed from the raw data
- **-a=cov**, the averaged time series is computed from the anomalies
- **-a=cor**, the averaged time series is computed from the standardized anomalies.

- 10) The *input_climatology_netcdf_file* specified with the **-c=** argument is needed only if **-a=cov** or **-a=cor**.

- 11) If **-a=cov** or **-a=cor**, the selected time period must agree with the climatology. This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument.

- 12) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.

- 13) The **-d=** argument specifies the weighting method for computing the averaged time series. If:

- **-d=dist2**, the averaged time series is computed with the diagonal distance associated with the horizontal 2-D grid-mesh (each grid point is weighted accordingly to the surface associated with it)
- **-d=ident**, the time series is computed with the identity metric (each grid point has the same weight when computing the averaged time series).

- 14) **-sm=smoothing_factor** means that the averaged time series must be smoothed with a moving average of approximately $2 * smoothing_factor + 1$ terms. *smoothing_factor* must be an integer greater than 0. However, if missing values are present in the computed averaged time series, smoothing is not allowed.

- 15) The **-n=output_netcdf_variable** argument specifies the NetCDF variable which will contains the computed time series in the output NetCDF file, *output_netcdf_file*, specified by the **-o=output_netcdf_file** argument.

- 16) The **-mi=missing_value** argument specifies the missing value indicator associated with the *netcdf_variable* (specified by the **-n=** argument) in the *output_netcdf_file*. *missing_value* must be a real number outside of the range of the *netcdf_variable*.

If the **-mi=** argument is not specified *missing_value* is set to 1.e+20.

- 17) The **-3d** argument specifies that the averaged time series must be stored as a tridimensional NetCDF variable with two dummy dimensions in the output NetCDF file.

By default, the time series is stored as an unidimensional NetCDF variable.

- 18) The **-double** argument specifies that, the time series is stored as double-precision floating point numbers in the output NetCDF file.

By default, the time series is stored as single-precision floating point numbers in the output NetCDF file.

- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Outputs

`comp_serie_miss_3d` creates an output NetCDF file that contains the computed time series. The output NetCDF dataset contains the following NetCDF variable (in the description below, *ntime* is the number of time steps selected with the **-t=** argument) :

- 1) *output_netcdf_variable*(*ntime*) : the averaged time series defined as an unidimensional variable.

or if the **-3d** argument has been specified :

- 1) *output_netcdf_variable*(*ntime*, 1, 1) : the averaged time series defined as a tridimensional variable with two dummy dimensions.

Examples

- 1) For computing a monthly time series from the file `HadISST2_1m_187001_200702_sst_reg1m.nc`, which includes a NetCDF variable `sst` with missing values, and store the results in a NetCDF variable named `nino34_sst` in the file `HadISST2_1m_187001_200702_nino34sst.nc`, use the following command :

```
$ comp_serie_miss_3d \
-f=HadISST2_1m_187001_200702_sst_reg1m.nc \
-v=sst \
-x=11,60 \
```

(continues on next page)

(continued from previous page)

```
-y=86,95 \  
-m=mask_HadISST2_sst.nc \  
-n=nino34_sst \  
-o=HadISST2_1m_187001_200702_nino34sst.nc
```

- 2) For computing a monthly global time series from the file `hadcrut2v.nc`, which includes a NetCDF variable `temanom` with missing values, and store the results in a NetCDF variable named `glob_temp` in the file `hadcrut2v_1m_glob_tempanom.nc`, use the following command :

```
$ comp_serie_miss_3d \  
-f=hadcrut2v.nc \  
-v=temanom \  
-m=mesh_mask_hadcrut2v.nc \  
-g=n \  
-a=scp \  
-n=glob_temp \  
-o=hadcrut2v_1m_glob_tempanom.nc
```

2.57 comp_spectrum_1d

2.57.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.57.2 Latest revision

30/04/2026

2.57.3 Purpose

Compute Fast Fourier Transform (FFT) estimates of the power spectrum of a real time series extracted from a uni- or bidimensional variable readed from a NetCDF dataset.

Spectral analysis amounts to estimate the power spectrum of a signal (e.g., a time series) from its time representation. The Power Spectral Density (PSD) estimates, which constitute the power spectrum, characterize the frequency content of the signal and decompose the variance of the signal into the different frequencies and allow to identify the periodicities present in the signal.

`comp_spectrum_1d` is based on a non-parametric method, which divides the time series into segments (see the `-sl=` argument), applies FFT on each segment, computes the PSD estimates from the FFTs and, finally, averages these estimates from each segment. Such non-parametric method is based on the observed time series and does not require prior knowledge about the data or a model generating the data. This method is known as the Welch's averaged, modified periodogram method for estimating the power spectrum of a time series [Welch] .

At the user option, the time series can be detrended before computing the PSD estimates (see the `-tr=` and `-tr2=` arguments) in order to ensure that the analyzed time series remain stationary.

In order to obtain stable PSD estimates (e.g., reduce the variance of the estimates) and minimize the loss of information, the real time series can also be segmented, and PSD estimates on, eventually overlapping, segments can be averaged at the user option. This is known as the Welch's method. The stability of the PSD estimates is increased by this averaging process because the Welch's method uses PSD estimates from different segments of the time series, which represent

approximately uncorrelated estimates of the true PSD estimates. That is, the greater the number of segments, the more stable, the resulting PSD estimates as their variance is reduced by the averaging.

The user can also select different data windows for the computations of the PSD estimates (see the **-win=** argument) on each segment. The segments are multiplied by a data window (selected with the **-win=** argument) before performing the FFTs, so that the Welch's method used in `comp_spectrum_1d` computes a modified periodogram on each segment and then averaged these estimates to produce the final PSD estimates. Because the segments usually overlap (see the **-nooverlap** argument), data values at the beginning and end of each segment tapered by the data window on one segment, occur away from the ends of adjacent segments. In this way, the overlap guards against the loss of information caused by windowing, but allows to increase the number of segments and, finally, PSD estimates on each segment can still be considered as uncorrelated.

Optionally, the PSD estimates may also be smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument).

Additionally, the spectral analysis can be done separately for different segments of equal length of the selected time series if these time series are not continuous in time (see the **-p=** argument).

The PSD estimates are returned in units, which are the square of the data if the **-nonormpsd** argument is used (meaning that the sum of the PSD estimates is equal to the variance of the time series) or in PSD units otherwise (meaning that the total area under the power spectrum is equal to the variance of the time series).

Finally, again at the user option, statistical confidence interval and equivalent number of degrees of freedom of the PSD estimates can be computed (see the **-conflim=** argument).

All these power spectrum statistics are stored in an output NetCDF dataset.

If the NetCDF variable is tridimensional use `comp_cross_spectrum_3d` or `comp_cross_spectrum2_3d` instead of `comp_spectrum_1d` to estimate the power spectra of the multichannel time series. If the time series has a seasonal (or diurnal) cycle, use `comp_stl_1d` in order to estimate and remove the harmonic components of the time series before using `comp_spectrum_1d`.

If you want to compute a cross-power spectrum analysis between two time series use `comp_cross_spectrum_1d` instead of `comp_spectrum_1d`.

Finally, use `comp_spectvar_1d` to compute variance or PSD estimates in a selected frequency band for the real time series before or after computing the power spectrum of the real time series.

For more information on spectral analysis, see the references below [Welch] [Bloomfield] [Diggle].

2.57.4 Further Details

Usage

```
$ comp_spectrum_1d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -t=time1,time2                (optional) \
  -p=period_length              (optional) \
  -o=output_netcdf_file        (optional) \
  -ni=index_for_2d_netcdf_variable (optional) \
  -sl=segment_length           (optional : 16,32,64,128,...) \
  -tr=trend_removal            (optional : 0,1,2,3) \
  -tr2=trend_removal_on_segment (optional : 0,1,2,3) \
  -win=window_choice           (optional : 1,2,3,4,5,6) \
  -tap=tapered_percentage      (optional : 0.0 > 1.) \
  -nsm=Daniell_filter_length   (optional) \
  -pro=critical_probability    (optional : 0.0 > 1.) \
```

(continues on next page)

(continued from previous page)

-mi=missing_value	(optional) \
-nonormpsd	(optional) \
-nooverlap	(optional) \
-conflim	(optional) \
-double	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- t=** the whole time period associated with the *netcdf_variable*
- p=** the *period_length* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time interval
- o=** the *output_netcdf_file* is named *spectrum_netcdf_variable.nc*
- ni=** if the *netcdf_variable* is bidimensional, the first time series is used
- sl=** the *segment_length* is set to 0, which means that the time series (or each time interval if the **-p=** argument is used) is not segmented to stabilize the the PSD estimates
- tr=** the *trend_removal* is set to 1, which means that the mean of the whole time series is removed before the spectral computations
- tr2=** the *trend_removal_on_segment* is set to 0, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_choice* is set to 1, which means that the Bartlett window is used in the spectral computations
- tap=** the *tapered_percentage* is set to 0.2, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to 6 with the **-win=** argument), 20% of the data are tapered
- nsm=** the *Daniell_filter_length* is set to 0, which means that the PSD estimates are not smoothed in the frequency domain by a moving average
- pro=** the *critical_probability* is set to 0.05, which means that if a confidence interval for the spectrum is requested with the **-conflim** argument, the lower and upper 95% confidence limit factors are computed
- mi=** the *missing_value* for the output NetCDF variables is set to $1.e+20$
- nonormpsd** the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series. If **-nonormpsd** is specified, the PSD estimates are returned in units which are the square of the data
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- conflim** normally, the equivalent number of degrees of freedom and the confidence limit factors of the power spectrum estimates are not computed. However, if **-conflim** is specified, these statistics are computed and stored in the output NetCDF dataset
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

-hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

-cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension (e.g., the segment axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable (e.g., the time series) for which a power spectrum must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to estimate the power spectrum.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

$time2 - time1 + 1$ must be an even integer if the **-sl=** argument is not used and of any length if this argument is used.

- 3) If the **-p=** argument is specified, the power spectrum estimates are computed separately for each time period of length *period_length* (as determined by the value of the **-p=** argument).

The whole selected time period (e.g., $time2 - time1 + 1$) must be a multiple of the *period_length*. The *period_length* must also be of even length if the **-sl=** argument is not used.

- 4) The **-ni=index_for_2d_netcdf_variable** argument specifies the index for selecting the time series if the *netcdf_variable* is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *index_for_2d_netcdf_variable* to 1.

- 5) The **-sl=segment_length** argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., $time2 - time1 + 1$) or the *period_length* if the **-p=** argument is used. Spectral computations are at $(segment_length/2) + 1$ Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies $(i-1) / segment_length$ for $i=1, 2, \dots, (segment_length/2) + 1$). By default, *segment_length* is set to 0, e.g., the time series is not segmented and the segment length is equal to $time2 - time1 + 1$ or *period_length* if the **-p=** argument is used.

- 6) The **-tr=** argument specifies pre-filtering processing of the real time series before estimating the power spectrum of the time series. If:

- **-tr=+1**, the mean of the time series is removed before computing the power spectrum
- **-tr=+2**, the drift from the time series is removed before computing the power spectrum. The drift from the time series (say *tseries*) is estimated using the formula: $drift = (tseries(ntime) - tseries(1)) / (ntime - 1)$
- **-tr=+3**, the least-squares line from the time series is removed before computing the power spectrum.

For other values of the **-tr=** argument, nothing is done before estimating the spectrum.

If the **-p=** argument is present, the pre-filtering processing is applied to each time interval of length *period_length*, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 1, e.g., the mean of the time series is removed before the spectrum computations.

- 7) The **-tr2=** argument specifies pre-filtering processing of the real time series before estimating the power spectrum on each time segment. If:
- **-tr2=+1**, the mean of the time segment is removed before computing the power spectrum on the segment
 - **-tr2=+2**, the drift from the time segment is removed before computing the power spectrum on the segment
 - **-tr2=+3**, the least-squares line from the time segment is removed before computing the power spectrum on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the spectrum on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 8) The **-win=window_choice** argument specifies the form of the data window used in the computations of the modified periodogram on each segment. If:
- **-win=1** The Bartlett window is used
 - **-win=2** The square window is used
 - **-win=3** The Welch window is used
 - **-win=4** The Hann window is used
 - **-win=5** The Hamming window is used
 - **-win=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6. The default value for the *window_choice* is 1, e.g., the Bartlett window is used for the spectral computations on each segment.

- 9) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 10) The **-nsm=Daniell_filter_length** argument specified that the PSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average). On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD estimates are not smoothed in the frequency domain.

- 11) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, comp_spectrum_1d segments the time series without any overlapping (in each time period if the **-p=** argument is used).

By default, comp_spectrum_1d overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting (modified if the **-win=** argument is used) periodograms will be averaged together to obtain PSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 12) If the **-nonormpsd** argument is specified, the sum of the PSD estimates is equal to the variance of the time series. By default, the PSD estimates are normalized in such a way that the total area under the power spectrum is equal to the variance of the time series.

- 13) If the **-conflim** argument is specified, the equivalent number of degrees of freedom, the lower and upper $(1 - critical_probability) * 100$ % confidence limit factors of the PSD estimates are computed. Multiply the

PSD estimates by the lower and upper confidence limit factors to get the lower and upper limits of a $(1 - \text{critical_probability}) * 100$ % confidence interval for the PSD estimates.

By default, these statistics are not computed.

- 14) The **-pro=***critical_probability* argument specifies the critical probability which is used to determine the lower and upper confidence limit factors. *critical_probability* must be greater than 0. and less than 1.. The default value is 0.05.
- 15) The **-mi=***missing_value* argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 16) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file. By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) It is assumed that the real time series has no missing values.
- 20) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 21) For more details on power spectrum analysis and examples of use in the climate literature, see
 - “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
 - “Fourier analysis of time series- An introduction”, by P. Bloomfield, John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
 - “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
 - “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
 - “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
 - “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_spectrum_1d` creates an output NetCDF file that contains the power spectrum of the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nfreq` is the number of frequencies where the PSD estimates are computed as determined by the `-t=`, `-p=` and `-sl=` arguments and `nperiod` is 1 if the `-p=` argument is not used or equal to `ntime / period_length` if it is used, where `ntime = time2 - time1 + 1`) :

- 1) `frequency(nfreq)` : the frequencies at which the PSD estimates are computed.
- 2) `periodicity(nfreq)` : the corresponding periodicities in number of time observations.
- 3) `netcdf_variable_spectrum(nperiod, nfreq)` : the corresponding PSD estimates for each time interval of the time series associated with the input NetCDF variable.
- 4) `edof(nfreq)` : the equivalent number of degrees of freedom of the PSD estimates of the time series associated with the input NetCDF variable.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_spectrum_1d`.

- 5) `lower_factor(nfreq)` : the lower confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding lower confidence limit factors to get the lower limits of the $(1 - critical_probability) * 100$ % confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_spectrum_1d`.

- 6) `upper_factor(nfreq)` : the upper confidence limit factors of the PSD estimates. Multiply the PSD estimates by the corresponding upper confidence limit factors to get the upper limits of the $(1 - critical_probability) * 100$ % confidence intervals for the PSD estimates.

This variable is stored only if the **-conflim** argument has been specified when calling `comp_spectrum_1d`.

Examples

- 1) For computing the power spectrum of a real monthly time series from a unidimensional NetCDF variable called `sst` extracted from the file `sst.monthly.nino34.nc` and store the results in the NetCDF file `spectrum_sst_nino34.nc`, use the following command :

```
$ comp_spectrum_1d \  
-f=sst.monthly.nino34.nc \  
-v=sst \  
-sl=128 \  
-nonormpsd \  
-nooverlap \  
-o=spectrum_sst_nino34.nc
```

2.58 comp_spectrum_diff_3d

2.58.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.58.2 Latest revision

30/04/2026

2.58.3 Purpose

Test if two multi-channel power spectra, computed from two real multi-channel time series with the help of *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d*, are significantly different or have a different shape using various nonparametric statistical tests based on the average of the squares of the (Log) spectral ratios of the two estimated multi-channel power spectra (e.g., a CHI2 statistic), the minimum and maximum values of these spectral ratios or the differences between the two (normalized) cumulative periodograms derived from the two estimated multi-channel power spectra [Diggle] [Priestley] [Jenkins_Watts] [Diggle_Fisher] [Coates_Diggle] [Potscher_Reschenhofer] [Potscher_Reschenhoferb] .

The critical probabilities associated with these different spectral statistics are estimated or approximated under the assumptions (e.g., the null hypothesis) that the two “true” underlying multi-channel power spectra are the same or have the same shape depending on the selected test statistic.

See below for more details on the different test statistics available in *comp_spectrum_diff_3d* and their associated null hypothesis, e.g., same underlying “true” power spectrum or same shape of the two “true” power spectra.

The two input multi-channel power spectra are read from two tridimensional variables (as specified by the **-nv=** and **-dv=** arguments) readed from one or two NetCDF datasets (as specified by the **-nf=** and **-df=** arguments).

It is assumed that the two estimated multi-channel power spectra have been computed by the Welch’s method [Welch] by calls to *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* with the **-conflim** argument. This optional argument is required for computing the DOFs of the Power Spectral Density (PSD) estimates, which are used by *comp_spectrum_diff_3d*, excepted when **-a=ks_perm** or **chi2_perm** is specified in the call to *comp_spectrum_diff_3d*. Note also that the methods used to estimate the two multi-channel power spectra must be identical and that the two multi-channel power spectra must not have been smoothed in the frequency domain.

The multi-channel test statistics and their associated critical probabilities are stored in an output NetCDF dataset.

If you want to get a pointwise tolerance interval of the given (Log) spectral ratios for two estimated multi-channel power spectra, use *comp_spectrum_ratio_3d* instead of *comp_spectrum_diff_3d*.

For additional details on these spectral computations, see [Welch] [Diggle] [Priestley] [Jenkins_Watts] [Coates_Diggle] [Diggle_Fisher] [Potscher_Reschenhoferb] .

2.58.4 Further Details

Usage

```
$ comp_spectrum_diff_3d \
  -nf=numerator_netcdf_file \
  -nv=numerator_netcdf_variable \
  -dv=denominator_netcdf_variable \
  -df=denominator_netcdf_file      (optional) \
  -m=input_mesh_mask_netcdf_file   (optional) \
  -g=grid_type                      (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                     (optional) \
  -y=lat1,lat2                     (optional) \
  -np=numerator_period              (optional) \
  -dp=denominator_period           (optional) \
  -t=freq1,freq2                   (optional) \
```

(continues on next page)

(continued from previous page)

```

-a=type_of_statistic          (optional : minmax, range, chi2, chi2_perm, ↵
↵ks_perm) \
-nb=number_of_shuffles        (optional) \
-o=output_netcdf_file         (optional) \
-mi=missing_value             (optional) \
-ksnonorm                     (optional) \
-double                       (optional) \
-bigfile                      (optional) \
-hdf5                         (optional) \
-cdf5                         (optional) \
-tlimited                      (optional)

```

By default

- df=** the *denominator_netcdf_file* is the same as the *numerator_netcdf_file*
- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input *numerator_netcdf_variable* and *denominator_netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *numerator_netcdf_variable* and *denominator_netcdf_variable*
- y=** the whole latitude domain associated with the *numerator_netcdf_variable* and *denominator_netcdf_variable*
- np=** the *numerator_period* argument is not used
- dp=** the *denominator_period* argument is not used
- t=** the whole set of frequencies included in the two input NetCDF files, but excluding the zero and Nyquist frequencies
- a=** the *type_of_statistic* is set to *chi2*, e.g., a CHI2 statistic is used as the spectral test statistic
- nb=** the *number_of_shuffles* is set to 999, meaning that 999 random interchanges are performed if a randomization test is requested (e.g., if **-a=chi2_perm** or **-a=ks_perm**). For other values of the **-a=** argument, the **-nb=** argument has no effect
- o=** *output_netcdf_file* name is set to *spectrum_diff_numerator_netcdf_variable.denominator_netcdf_variable.nc* where *numerator_netcdf_variable* and *denominator_netcdf_variable* are specified with the **-nv=** and **-dv=** arguments.
- mi=** the *missing_value* for the output NetCDF variables is set to 1.e+20
- ksnonorm** the Kolmogorov-Smirnov statistic is computed from the normalized cumulative periodograms if **-a=ks_perm** is specified. However, if **-ksnonorm** is also specified, the cumulative periodograms are not normalized before computing the Kolmogorov-Smirnov statistic. For other values of the **-a=** argument, the **-ksnonorm** argument has no effect
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. However, if **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. However, if **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the frequency axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the frequency dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-nf=numerator_netcdf_file** argument specifies the NetCDF file containing the multi-channel PSD estimates and their associated degrees of freedom for the numerator of the spectral ratio or for the first real multi-channel time series if a Kolmogorov-Smirnov statistic is used.

It is assumed that this NetCDF file has been created by NCSTAT procedures performing spectral computations like *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* with the **-conflim** argument activated (e.g., this NetCDF file must contain two NetCDF variables named edof and freq).

- 2) The **-df=denominator_netcdf_file** argument specifies the NetCDF file containing the multi-channel PSD estimates and their associated degrees of freedom for the denominator of the spectral ratio or for the second real multi-channel time series if a Kolmogorov-Smirnov statistic is used.

If this argument is not specified, it is assumed that the *denominator_netcdf_file* is the same as the *numerator_netcdf_file*.

It is also assumed that this NetCDF file has been created by NCSTAT procedures performing spectral computations like *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* with the **-conflim** argument activated (e.g., this NetCDF file must contain two NetCDF variables named edof and freq).

- 3) The **-nv=numerator_netcdf_variable** and **-dv=denominator_netcdf_variable** arguments specify the NetCDF variables containing the multi-channel PSD estimates for the numerator and denominator of the spectral ratio, respectively (or the first and second PSD estimates when a Komogorov-Smirnov statistic is used).

The geographical shapes of the *numerator_netcdf_variable* and *denominator_netcdf_variable* must agree.

In addition, the multi-channel PSD estimates stored in *numerator_netcdf_variable* and *denominator_netcdf_variable* must have been normalized in the same way and not been smoothed in the frequency domain (for details, see the arguments **-nsm=** and **-nonormpsd** in the NCSTAT procedures *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* in order to verify the conditions of use of the statistical tests available in *comp_spectrum_diff_3d*).

Finally, if **-a=chi2_perm** or **-a=ks_perm**, the two estimated multi-channel power spectra must have been obtained by averaging exactly the same number of (modified) periodograms.

- 4) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to the *numerator_netcdf_variable* and *denominator_netcdf_variable* for transforming these tridimensional NetCDF variables as rectangular matrices before computing the spectral ratio of the multi-channel PSD estimates or the differences between their cumulative periodograms.

By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variables is a valid power spectrum (e.g., missing values are not present in the PSD estimates).

The geographical shapes of the *numerator_netcdf_variable* (in the *numerator_netcdf_file*), the *denominator_netcdf_variable* (in the *denominator_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to *comp_clim_3d* or *comp_mask_3d* for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_spectrum_diff_3d*.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the input NetCDF variables are from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the `mesh_mask` variable if an `input_mesh_mask_netcdf_file` is used.
- 6) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the `numerator_netcdf_variable` and `denominator_netcdf_variable` is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for `lon1`. In this case the longitude domain is from `nlon+lon1+1` to `lon2` where `nlon` is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using `comp_spectrum_diff_3d`.

- 7) If the **-np=numerator_period** argument is specified, the multi-channel PSD estimates in the `numerator_netcdf_variable` have been computed for different periods separately (as determined by the value of the **-p=period_length** argument in [comp_cross_spectrum_3d](#)). In that case, the **-np=** argument selects the period in the `numerator_netcdf_variable` which must be used for computing the spectral ratio or the difference between the cumulative periodograms of the two multi-channel PSD estimates.
- 8) If the **-dp=denominator_period** argument is specified, the multi-channel PSD estimates in the `denominator_netcdf_variable` have been computed for different periods separately (as determined by the value of the **-p=period_length** argument in [comp_cross_spectrum_3d](#)). In that case, the **-dp=** argument selects the period in the `denominator_netcdf_variable` which must be used for computing the spectral ratio or the difference between the cumulative periodograms of the two multi-channel PSD estimates.
- 9) If the **-t=freq1,freq2** argument is missing, the spectral ratios or differences between the cumulative periodograms of the two multi-channel PSD estimates are computed for the whole set of frequencies stored in the `numerator_netcdf_variable` and `denominator_netcdf_variable` (e.g., the multi-channel power spectra must have been estimated for the same set of frequencies).

The selected frequency interval is a vector of two integers specifying the first and last frequencies for which the spectral ratios or differences between the cumulative periodograms must be computed. The indices are relative to 1.

However, in all cases, the zero and Nyquist frequencies are excluded from the computations of the statistics.

- 10) The **-a=** argument selects the type of test statistic and/or the method for computing critical probabilities associated with the test statistic:
 - If **-a=minmax**, the minimum and maximum values of the spectral ratios of the two estimated multi-channel spectra are used as a test statistic. The critical probabilities are derived from the associated probabilities of obtaining, respectively, a value less (for the minimum ratio) and higher (for the maximum ratio) than attained under the null hypothesis of a common underlying population spectrum for the two multi-channel time series. The computed critical probabilities are only approximate and are conservative. The null hypothesis is that the two “true” underlying multi-channel spectra are the same. For more details, see Chapter 4 of [\[Diggle\]](#).
 - If **-a=range**, the range of the Log spectral ratios of the two estimated multi-channel spectra is used as a test statistic. The null hypothesis is that of a common shape of the two multi-channel power spectra of the two multi-channel time series. For more details and theory, see [\[Coates_Diggle\]](#) [\[Potscher_Reschenhofer\]](#) [\[Potscher_Reschenhofer\]](#).
 - If **-a=chi2**, a standard CHI2 statistic computed from the Log spectral ratios of the two multi-channel power spectra is used as a test statistic. The null hypothesis is that of a common spectrum of the two multi-channel series and asymptotic theory is used to derive the associated critical probabilities. For more details and theory, see [\[Priestley\]](#) [\[Jenkins_Watts\]](#).

- If **-a=chi2_perm**, a CHI2 statistic computed from the Log spectral ratios is also used, but the distribution of this statistic under the null hypothesis of a common power spectrum for the two multi-channel series is approximated by recalculating the CHI2 statistic for some large number (e.g., see the description of the **-nb=** argument below) of random interchanges of periodogram or PSD ordinates at each frequency for the two estimated multi-channel power spectra. This statistical randomization test relies on the assumptions that the different spectral ordinates have the same sampling distribution and are independent of each other [Priestley] . This means, in particular, that the PSD ordinates corresponding to the zero and Nyquist frequencies must be excluded and that the two estimated multi-channel power spectra have not been obtained by smoothing the periodograms in the frequency domain. For more details and theory, see [Priestley] [Diggle_Fisher] .
- If **-a=ks_perm**, a Kolmogorov-Smirnov statistic is used, but the distribution of this statistic under the null hypothesis of a common shape for the two “true” power spectra of the two multi-channel series is approximated by calculating the Kolmogorov-Smirnov statistic for some large number (e.g., as determined by the **-nb=** argument) of random interchanges of periodogram or PSD ordinates at each frequency for the two estimated multi-channel spectra. For more details and theory, see [Priestley] [Diggle_Fisher] .

If **-a=chi2_perm** or **ks_perm**, the number of frequencies used (e.g., as determined by the **-t=** argument) must be greater or equal to 10.

The default is **-a=chi2**.

- 11) The **-nb=number_of_shuffles** argument specifies the number of random interchanges of periodogram or PSD ordinates at each frequency in order to approximate the randomization distribution of the Kolmogorov-Smirnov or CHI2 statistics under the null hypothesis if **-a=chi2_perm** or **ks_perm**.

The default value is 999.

- 12) The **-ksnonorm** argument specifies that the Kolmogorov-Smirnov statistics must be computed from non-normalized cumulative periodograms if **-a=ks_perm**. In that case the null hypothesis which is tested is that the “true” underlying population power spectra of the two multi-channel time series are the same instead of having the same shape.

By default, the Kolmogorov-Smirnov statistics are calculated from normalized cumulative periodograms and the null hypothesis which is tested is that of a common shape for the two “true” underlying multi-channel spectra (see the description of the **-a=** argument for more details).

- 13) The **-mi=missing_value** argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* in the output NetCDF variables is set to $1.e+20$.

- 14) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 15) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has

been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The `-cdf5` argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 18) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 19) For more details on power spectrum analysis, theory for the statistical tests used in `comp_spectrum_diff_3d` and examples of use in the climate literature, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
- “Tests for comparing two estimated spectral densities”, by D.S. Coates and P.J. Diggle, Journal of Time series Analysis, 7:1, 7-20, 1986. DOI: <https://doi.org/10.1111/j.1467-9892.1986.tb00482.x>
- “Nonparametric comparison of cumulative periodograms”, by P.J. Diggle and N.I. Fisher, Applied Statistics, 40:3, 423-434, 1991. DOI: <https://doi.org/10.2307/2347522>
- “Spectral Analysis and its Applications”, by G.M. Jenkins and D.G., San Francisco: Holden-Day. ISBN-10: 0816244642, 1968.
- “Spectral Analysis and Time Series”, by M.B. Priestley, London: Academic Press, 1981. ISBN-10: 0125649223
- “Discriminating between two spectral densities in case of replicated observations”, by B.M. Potscher and E. Reschenhofer, Journal of Time series Analysis, 9:3, 221-224, 1988. DOI: <https://doi.org/10.1111/j.1467-9892.1988.tb00466.x>
- “Distribution of the Coates-Diggle test statistic in case of replicated observations”, by B.M. Potscher and E. Reschenhofer, Statistics, 20:3, 417-421, 1989. DOI: <https://doi.org/10.1080/02331888908802190>
- “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
- “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
- “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_spectrum_diff_3d` creates an output NetCDF file that contains the selected test statistics for the two input multi-channel power spectra and the associated critical probabilities under the assumption that the two “true” underlying multi-channel power spectra are the same or have the same shape depending on the selected test statistic (see the description of the `-a=` argument above for more details on the available test statistics and the null hypothesis used to derive the critical probabilities for each `type_of_statistic`).

The output NetCDF dataset contains the following NetCDF variables (in the description below, *nlat* and *nlon* are the lengths of the spatial dimensions of the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*, and *nfreq* is the number of frequencies at which the two input multi-channel power spectra have been estimated) :

- 1) frequency (*nfreq*) : the frequencies at which the input PSD estimates have been computed.
- 2) periodicity (*nfreq*) : the corresponding periods in number of time observations.
- 3) *numerator_netcdf_variable_denominator_netcdf_variable_stat* (*nlat*, *nlon*) : the multi-channel test statistics computed from the two input multi-channel power spectra stored in the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*. The type of the computed statistic, as determined by the **-a=** argument, is stored as a global attribute in the output NetCDF dataset.
- 4) *numerator_netcdf_variable_denominator_netcdf_variable_prob* (*nlat*, *nlon*) : the multi-channel critical probabilities associated with the computed multi-channel test statistics. These critical probabilities are computed or estimated (depending on the type of statistic) under the null hypothesis that the two “true” multi-channel power spectra are identical or have the same shape.

If the **-a=minmax** argument is used, these critical probabilities are conservative and are simply computed as two times the minimum of the critical probabilities associated with the minimum and maximum values of the spectral ratios computed from the two input multi-channel power spectra. The minimum and maximum values of the spectral ratios and their associated critical probabilities are also stored in the output NetCDF dataset.

- 5) *numerator_netcdf_variable_denominator_netcdf_variable_minratio_stat* (*nlat*, *nlon*) : the multi-channel minimum values of the spectral ratios computed from the two input multi-channel power spectra stored in the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*.

This variable is stored only if the **-a=minmax** argument has been specified when calling `comp_spectrum_diff_3d`.

- 6) *numerator_netcdf_variable_denominator_netcdf_variable_minratio_prob* (*nlat*, *nlon*) : the multi-channel critical probabilities associated with the minimum values of the multi-channel spectral ratios. These critical probabilities give the probabilities of obtaining minimum values less than attained under the null hypothesis of a common power spectra for the two (multi-channel) time series.

This variable is stored only if the **-a=minmax** argument has been specified when calling `comp_spectrum_diff_3d`.

- 7) *numerator_netcdf_variable_denominator_netcdf_variable_maxratio_stat* (*nlat*, *nlon*) : the multi-channel maximum values of the spectral ratios computed from the two input multi-channel power spectra stored in the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*.

This variable is stored only if the **-a=minmax** argument has been specified when calling `comp_spectrum_diff_3d` and is used to derive the conservative critical probabilities stored in the NetCDF variable *numerator_netcdf_variable_denominator_netcdf_variable_prob*.

- 8) *numerator_netcdf_variable_denominator_netcdf_variable_maxratio_prob* (*nlat*, *nlon*) : the multi-channel critical probabilities associated with the maximum values of the multi-channel spectral ratios. These critical probabilities give the probabilities of obtaining maximum values higher than attained under the null hypothesis of a common power spectra for the two (multi-channel) time series.

This variable is stored only if the **-a=minmax** argument has been specified when calling `comp_spectrum_diff_3d` and is used to derive the conservative critical probabilities stored in the

NetCDF variable *numerator_netcdf_variable_denominator_netcdf_variable_prob*.

All the multi-channel spectral statistics and their associated critical probabilities involving the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable* are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with these input NetCDF variables even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For testing that the two multi-channel power spectra stored in a NetCDF variable called *sst* extracted, respectively, from the files *spectrum_sst.monthly.1979_2005.nc* and *spectrum_sst.monthly.1950_1978.nc* are the same (more precisely that the two “true” underlying power spectra are identical) using a standard CHI2 statistic, use the following command :

```
$ comp_spectrum_diff_3d -fn=spectrum_sst.monthly.1979_2005.nc \  
                        -fd=spectrum_sst.monthly.1950_1978.nc \  
                        -nv=sst \  
                        -dv=sst \  
                        -o=spectrum_diff_sst_1979_2005.1950_1978.nc
```

2.59 comp_spectrum_ratio_1d

2.59.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.59.2 Latest revision

30/04/2026

2.59.3 Purpose

Compute the ratio of two Power Spectral Density (PSD) estimates (computed from two real time series with the help of *comp_spectrum_1d* or *comp_cross_spectrum_1d*) and a pointwise tolerance interval for this spectral ratio under the assumption that the two “true” underlying power spectra are the same. The two PSD estimates are readed from two unidimensional variables (as specified by the **-nv=** and **-dv=** arguments) readed from one or two NetCDF datasets (as specified by the **-nf=** and **-df=** arguments).

How to compare quantitatively two estimated power spectra is an important question solved by *comp_spectrum_ratio_1d*. Suppose that $f1(w)$ and $f2(w)$ are PSD estimates (where w is a Fourier frequency) with $df1$ and $df2$ Degrees Of Freedom (DOF), respectively. To answer this question, spectral analysis theory [Diggle] suggests to examine the spectral ratio

$$R(w) = f1(w) / f2(w)$$

, which may be assumed to follow an F-distribution with numerator and denominator degrees of freedom $df1$ and $df2$, respectively. This result can be used to calculate pointwise confidence (or tolerance) intervals for the spectral ratios at the Fourier frequencies [Diggle] [Masson_etal] [Terray_etalb].

It is assumed that the two estimated power spectra have been computed by the Welch's method [Welch] by calls to *comp_spectrum_1d* or *comp_cross_spectrum_1d* with the **-conflim** argument (this argument is required for computing the DOFs of the PSD estimates,

which are required by *comp_spectrum_ratio_1d*).

The pointwise tolerance interval and the spectral ratio of the two estimated power spectra (if both the **-nv=** and **-dv=** arguments are specified in the call to *comp_spectrum_ratio_1d*) are stored in an output NetCDF dataset.

For additional details on these spectral computations, see [Welch] [Bloomfield] [Diggle] .

2.59.4 Further Details

Usage

```
$ comp_spectrum_ratio_1d \
  -nf=numerator_netcdf_file \
  -df=denominator_netcdf_file      (optional) \
  -nv=numerator_netcdf_variable    (optional) \
  -dv=denominator_netcdf_variable (optional) \
  -np=numerator_period              (optional) \
  -dp=denominator_period           (optional) \
  -t=freq1,freq2                   (optional) \
  -o=output_netcdf_file             (optional) \
  -pro=probability_interval         (optional : 0.0 > 1.) \
  -mi=missing_value                 (optional) \
  -logratio                         (optional) \
  -double                           (optional) \
  -hdf5                             (optional) \
  -cdf5                             (optional) \
  -tlimited                          (optional)
```

By default

- df=** the *denominator_netcdf_file* is the same as the *numerator_netcdf_file*
- nv=** a *numerator_netcdf_variable* is not used and only the upper and lower critical ratios for each frequency are computed
- dv=** a *denominator_netcdf_variable* is not used and only the upper and lower critical ratios for each frequency are computed
- np=** the *numerator_period* argument is not used
- dp=** the *denominator_period* argument is not used
- t=** the whole set of frequencies included in the two input NetCDF files
- o=** *output_netcdf_file* name is set to *spectrum_ratio_numerator_netcdf_variable.denominator_netcdf_variable.nc* if *numerator_netcdf_variable* and *denominator_netcdf_variable* are both specified with the **-nv=** and **-dv=** arguments, and to *spectrum_ratio.nc* if the **-nv=** and **-dv=** arguments are not used.
- pro=** the *probability_interval* is set to 0.90, which means that a 90% tolerance interval for the ratio of the estimated power spectra is computed
- mi=** the *missing_value* for the output NetCDF variables is set to 1.e+20

- logratio** the spectral ratios and their associated upper and lower critical ratios are not Log transformed. If **-logratio** is specified, the spectral ratios, upper and lower critical ratios are Log transformed
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. However, if **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the frequency axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the frequency dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-nf=numerator_netcdf_file** argument specifies the NetCDF file containing the power spectrum estimates and their associated degrees of freedom for the numerator of the spectral ratio.

It is assumed that this NetCDF file has been created by NCSTAT procedures performing spectral computations like *comp_spectrum_1d* or *comp_cross_spectrum_1d* with the **-conflim** argument activated (e.g., this NetCDF file must contain two NetCDF variables named edof and freq).

- 2) The **-df=denominator_netcdf_file** argument specifies the NetCDF file containing the spectrum estimates and their associated degrees of freedom for the denominator of the spectral ratio. If this argument is not specified, it is assumed that the *denominator_netcdf_file* is the same as the *numerator_netcdf_file*.

It is assumed that this NetCDF file has been created by NCSTAT procedures performing spectral computations like *comp_spectrum_1d* or *comp_cross_spectrum_1d* with the **-conflim** argument activated (e.g., this NetCDF file must contain two NetCDF variables named edof and freq).

- 3) The **-nv=numerator_netcdf_variable** and **-dv=denominator_netcdf_variable** arguments specify the NetCDF variables containing the PSD estimates for the numerator and denominator of the spectral ratio, respectively.

If one or two of these optional arguments are absent, *comp_spectrum_ratio_1d* computes only the upper and lower critical ratios associated with the (*probability_interval* * 100)% tolerance interval (*probability_interval* is the value of the **-pro=** argument described below).

- 4) If the **-np=numerator_period** argument is specified, the PSD estimates in the *numerator_netcdf_variable* have been computed for different periods separately (as determined by the value of the **-p=period_length** argument in *comp_spectrum_1d* or *comp_cross_spectrum_1d*). In that case, the **-np=** argument selects the period in the *numerator_netcdf_variable* which must be used for computing the spectral ratio of the two PSD estimates (e.g., this argument is only used and useful, if both the **-nv=** and **-dv=** arguments are also specified).

- 5) If the **-dp=denominator_period** argument is specified, the PSD estimates in the *denominator_netcdf_variable* have been computed for different periods separately (as determined by the value of the **-p=period_length** argument in *comp_spectrum_1d* or *comp_cross_spectrum_1d*). In that case, the **-dp=** argument selects the period in the *denominator_netcdf_variable* which must be used for computing the spectral ratio of the two PSD estimates (e.g., this argument is only used and useful if both the **-nv=** and **-dv=** arguments are also specified).

- 6) If the **-t=freq1,freq2** argument is missing, the spectral ratios are computed for the whole set of frequencies stored in the *numerator_netcdf_variable* and *denominator_netcdf_variable* (e.g., the power spectra must have been estimated for the same set of frequencies).

The selected frequency interval is a vector of two integers specifying the first and last frequencies for which the spectral ratios and the associated tolerance intervals must be computed. The indices are relative to 1.

- 7) The **-pro=probability_interval** argument specifies that a (*probability_interval* * 100)% tolerance interval must be computed. This argument is used to determine the upper and lower critical spectral ratios of the computed tolerance interval. *probability_interval* must be greater than 0. and less than 1..

The default value is 0.90, e.g., a 90)% tolerance interval is estimated.

- 8) The **-mi=missing_value** argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* in the output NetCDF variables is set to 1.e+20.

- 9) The **-logratio** argument specifies that the logarithms of the spectral ratios must be computed and stored in the output NetCDF file.

- 10) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 11) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 12) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 13) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 14) For more details on power spectrum analysis, theory for computing a pointwise tolerance interval for the ratio of two estimated power spectra and examples in the climate literature, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
- “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
- “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
- “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_spectrum_ratio_1d` creates an output NetCDF file that contains the upper and lower critical spectral ratios associated with the specified pointwise tolerance interval and, eventually, the spectral ratios of the two input estimated power spectra, if both the **-nv=** and **-dv=** arguments have been used in the call to `comp_spectrum_ratio_1d`.

The output NetCDF dataset contains the following NetCDF variables (in the description below, `nfreq` is the number of frequencies for which the upper and lower critical (and eventually estimated) spectral ratios are computed as determined by the **-t=** and **-p=** arguments) :

- 1) `frequency(nfreq)` : the frequencies at which the spectral ratios are computed.
- 2) `periodicity(nfreq)` : the corresponding periods in number of time observations.
- 3) `lower_critical_ratio(nfreq)` : the lower critical spectral ratios associated with the specified pointwise tolerance interval.

This variable is stored only if the **-logratio** argument has not been specified when calling `comp_spectrum_ratio_1d`.

- 4) `upper_critical_ratio(nfreq)` : the upper critical spectral ratios associated with the specified pointwise tolerance interval.

This variable is stored only if the **-logratio** argument has not been specified when calling `comp_spectrum_ratio_1d`.

- 5) `numerator_netcdf_variable_denominator_netcdf_variable_ratio(nfreq)` : the spectral ratios computed from the two power spectra stored in the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*.

This variable is stored only if the two **-nv=** and **-dv=** arguments have been specified and if the **-logratio** argument has not been specified when calling `comp_spectrum_ratio_1d`.

- 6) `lower_critical_logratio(nfreq)` : the lower critical spectral ratios associated with the specified pointwise tolerance interval.

This variable is stored only if the **-logratio** argument has been specified when calling `comp_spectrum_ratio_1d`.

- 7) `upper_critical_logratio(nfreq)` : the upper critical spectral ratios associated with the specified pointwise tolerance interval.

This variable is stored only if the **-logratio** argument has been specified when calling `comp_spectrum_ratio_1d`.

- 8) `numerator_netcdf_variable_denominator_netcdf_variable_logratio(nfreq)` : the spectral ratios computed from the two spectra specified with the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*.

This variable is stored only if both the **-nv=** and **-dv=** arguments have been specified and if the **-logratio** argument has been specified when calling `comp_spectrum_ratio_1d`.

Examples

- 1) For computing the upper and lower critical spectral ratios associated with a 90% confidence interval of two power spectra stored, respectively, in the files `spectrum_sst.monthly.nino34.1979_2005.nc` and `spectrum_sst.monthly.nino34.1950_1978.nc`, use the following command :

```
$ comp_spectrum_ratio_1d -fn=spectrum_sst.monthly.nino34.1979_2005.nc \
                        -fd=spectrum_sst.monthly.nino34.1950_1978.nc \
                        -o=spectrum_ratio_sst_nino34.nc
```

2.60 comp_spectrum_ratio_3d

2.60.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.60.2 Latest revision

30/04/2026

2.60.3 Purpose

Compute the ratio of two multi-channel Power Spectral Density (PSD) estimates (computed from two real multi-channel time series with the help of *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d*) and a pointwise tolerance interval for this spectral ratio under the assumption that the two “true” underlying multi-channel power spectra are the same. The two multi-channel PSD estimates are readed from two tridimensional variables (as specified by the **-nv=** and **-dv=** arguments) readed from one or two estimates NetCDF datasets (as specified by the **-nf=** and **-df=** arguments).

How to compare quantitatively two estimated power spectra is an important question solved by *comp_spectrum_ratio_3d*. Suppose that $f1(w)$ and $f2(w)$ are (multi-channel) PSD estimates (where w is a Fourier frequency) with $df1$ and $df2$ Degrees Of Freedom (DOF), respectively. To answer this question, spectral analysis theory [Diggle] suggests to examine the spectral ratio

$$R(w) = f1(w) / f2(w)$$

, which may be assumed to follow an F-distribution with numerator and denominator degrees of freedom $df1$ and $df2$, respectively. This result can be used to calculate pointwise confidence (or tolerance) intervals for the spectral ratios at the Fourier frequencies [Diggle] [Masson_etal] [Terray_etalb].

It is assumed that the two estimated multi-channel power spectra have been computed by the Welch’s method [Welch] by calls to *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* with the **-conflim** argument (this argument is required for computing the DOFs of the PSD estimates, which are required by *comp_spectrum_ratio_3d*).

The pointwise tolerance interval and the spectral ratio of the two estimated multi-channel power spectra are stored in an output NetCDF dataset.

If you want to test if two estimated (multi-channel) power spectra have the same underlying population spectrum or if the underlying population spectra have the same shape for a given set of frequencies, use *comp_spectrum_diff_3d* instead of *comp_spectrum_ratio_3d*.

For additional details on these spectral computations, see [Welch] [Bloomfield] [Diggle] .

2.60.4 Further Details

Usage

```
$ comp_spectrum_ratio_3d \
  -nf=numerator_netcdf_file \
  -nv=numerator_netcdf_variable \
  -dv=denominator_netcdf_variable \
  -df=denominator_netcdf_file      (optional) \
  -m=input_mesh_mask_netcdf_file   (optional) \
  -g=grid_type                      (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                     (optional) \
  -y=lat1,lat2                     (optional) \
  -np=numerator_period              (optional) \
  -dp=denominator_period            (optional) \
  -t=freq1,freq2                   (optional) \
  -o=output_netcdf_file             (optional) \
  -pro=probability_interval         (optional : 0.0 > 1.) \
  -mi=missing_value                (optional) \
  -logratio                         (optional) \
  -double                           (optional) \
  -bigfile                          (optional) \
  -hdf5                             (optional) \
  -cdf5                             (optional) \
  -tlimited                          (optional)
```

By default

- df=** the *denominator_netcdf_file* is the same as the *numerator_netcdf_file*
- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n*, which means that the 2-D grid-mesh associated with the input *numerator_netcdf_variable* and *denominator_netcdf_variable* is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *numerator_netcdf_variable* and *denominator_netcdf_variable*
- y=** the whole latitude domain associated with the *numerator_netcdf_variable* and *denominator_netcdf_variable*
- np=** the *numerator_period* argument is not used
- dp=** the *denominator_period* argument is not used
- t=** the whole set of frequencies included in the two input NetCDF files
- o=** *output_netcdf_file* name is set to *spectrum_ratio_numerator_netcdf_variable.denominator_netcdf_variable.nc* where *numerator_netcdf_variable* and *denominator_netcdf_variable* are specified with the **-nv=** and **-dv=** arguments.
- pro=** the *probability_interval* is set to 0.90, which means that a 90% tolerance interval for the spectral ratio of the estimated multi-channel power spectra is computed
- mi=** the *missing_value* for the output NetCDF variables is set to 1.e+20
- logratio** the spectral ratios and their associated upper and lower critical spectral ratios are not Log transformed. However, if **-logratio** is specified, the spectral ratios, upper and lower critical spectral ratios are Log transformed
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

- bigfile** a NetCDF classical format file is created. However, if **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. However, if **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the frequency axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the frequency dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-nf=numerator_netcdf_file** argument specifies the NetCDF file containing the multi-channel power spectrum estimates and their associated degrees of freedom for the numerator of the spectral ratio.

It is assumed that this NetCDF file has been created by NCSTAT procedures performing spectral computations like *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* with the **-conflim** argument activated (e.g., this NetCDF file must contain two NetCDF variables named *edof* and *freq*).

- 2) The **-df=denominator_netcdf_file** argument specifies the NetCDF file containing the multi-channel power spectrum estimates and their associated degrees of freedom for the denominator of the spectral ratio. If this argument is not specified, it is assumed that the *denominator_netcdf_file* is the same as the *numerator_netcdf_file*.

It is assumed that this NetCDF file has been created by NCSTAT procedures performing spectral computations like *comp_cross_spectrum_3d* or *comp_cross_spectrum2_3d* with the **-conflim** argument activated (e.g., this NetCDF file must contain two NetCDF variables named *edof* and *freq*).

- 3) The **-nv=numerator_netcdf_variable** and **-dv=denominator_netcdf_variable** arguments specify the NetCDF variables containing the multi-channel PSD estimates for the numerator and denominator of the spectral ratio, respectively.

The geographical shapes of the *numerator_netcdf_variable* and *denominator_netcdf_variable* must agree.

- 4) The optional argument **-m=input_mesh_mask_netcdf_file** specifies the land-sea mask to apply to the *numerator_netcdf_variable* and *denominator_netcdf_variable* for transforming these tridimensional NetCDF variables as rectangular matrices before computing the spectral ratio of the multi-channel PSD estimates.

By default, it is assumed that each cell in the 2-D grid-mesh associated with the input tridimensional NetCDF variables is a valid power spectrum (e.g., missing values are not present).

The geographical shapes of the *numerator_netcdf_variable* (in the *numerator_netcdf_file*), the *denominator_netcdf_variable* (in the *denominator_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if an *input_mesh_mask_netcdf_file* is used.

Refer to *comp_clim_3d* or *comp_mask_3d* for creating a valid *input_mesh_mask_netcdf_file* NetCDF file for regular or gaussian grids before using *comp_spectrum_ratio_3d*.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the input NetCDF variables are from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). This argument is also used to determine the name of the *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used.
- 6) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing the whole geographical domain associated with the *numerator_netcdf_variable* and *denominator_netcdf_variable* is used.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *comp_spectrum_ratio_3d*.

- 7) If the **-np=***numerator_period* argument is specified, the multi-channel PSD estimates in the *numerator_netcdf_variable* have been computed for different periods separately (as determined by the value of the **-p=***period_length* argument in *comp_cross_spectrum_3d*). In that case, the **-np=** argument selects the period in the *numerator_netcdf_variable* which must be used for computing the spectral ratio of the two multi-channel PSD estimates.
- 8) If the **-dp=***denominator_period* argument is specified, the multi-channel PSD estimates in the *denominator_netcdf_variable* have been computed for different periods separately (as determined by the value of the **-p=***period_length* argument in *comp_cross_spectrum_3d*). In that case, the **-dp=** argument selects the period in the *denominator_netcdf_variable* which must be used for computing the spectral ratio of the two multi-channel PSD estimates.
- 9) If the **-t=freq1,freq2** argument is missing, the spectral ratios are computed for the whole set of frequencies stored in the *numerator_netcdf_variable* and *denominator_netcdf_variable* (e.g., the multi-channel power spectra must have been estimated for the same set of frequencies).

The selected frequency interval is a vector of two integers specifying the first and last frequencies for which the spectral ratios and the associated tolerance intervals must be computed. The indices are relative to 1.

- 10) The **-pro=***probability_interval* argument specifies that a (*probability_interval* * 100)% tolerance interval must be computed. This argument is used to determine the upper and lower critical spectral ratios of the computed tolerance interval. *probability_interval* must be greater than 0. and less than 1..

The default value is 0.90, e.g., a 90)% tolerance interval is estimated.

- 11) The **-mi=***missing_value* argument specifies the missing value indicator for the output NetCDF variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* in the output NetCDF variables is set to 1.e+20.

- 12) The **-logratio** argument specifies that the logarithms of the spectral ratios must be computed and stored in the output NetCDF file.
- 13) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built

with the `_USE_NETCDF44` CPP macro.

- 17) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 18) For more details on power spectrum analysis, theory for computing a pointwise tolerance interval for the ratio of two estimated power spectra and examples in the climate literature, see
 - “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
 - “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
 - “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
 - “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by S. Masson et al., 2012, Climate Dynamics, Vol. 39:681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
 - “The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model” by P. Terray et al., 2012, Climate Dynamics, Vol. 39:729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
 - “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_spectrum_ratio_3d` creates an output NetCDF file that contains the upper and lower critical spectral ratios associated with the specified pointwise tolerance interval and the multi-channel spectral ratios of the two input estimated multi-channel power spectra.

The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*, `nfreq` is the number of frequencies for which the multi-channel spectral ratios and, eventually, the upper and lower critical spectral ratios are computed as determined by the `-t=` and `-p=` arguments) :

- 1) `frequency(nfreq)` : the frequencies at which the spectral ratios are computed.
- 2) `periodicity(nfreq)` : the corresponding periods in number of time observations.
- 3) `lower_critical_ratio(nfreq)` : the lower critical spectral ratios associated with the specified pointwise tolerance interval.
 This variable is stored only if the **-logratio** argument has not been specified when calling `comp_spectrum_ratio_3d`.
- 4) `upper_critical_ratio(nfreq)` : the upper critical spectral ratios associated with the specified pointwise tolerance interval.
 This variable is stored only if the **-logratio** argument has not been specified when calling `comp_spectrum_ratio_3d`.
- 5) `numerator_netcdf_variable_denominator_netcdf_variable_ratio(nfreq,nlat,nlon)` : the multi-channel spectral ratios computed from the two multi-channel power spectra stored in the two input NetCDF variables *numerator_netcdf_variable* and *denominator_netcdf_variable*.
 This variable is stored only if the **-logratio** argument has not been specified when calling `comp_spectrum_ratio_3d`.

- 6) `lower_critical_logratio(nfreq)` : the lower critical spectral ratios associated with the specified pointwise tolerance interval.

This variable is stored only if the **-logratio** argument has been specified when calling `comp_spectrum_ratio_3d`.

- 7) `upper_critical_logratio(nfreq)` : the upper critical spectral ratios associated with the specified pointwise tolerance interval.

This variable is stored only if the **-logratio** argument has been specified when calling `comp_spectrum_ratio_3d`.

- 8) `numerator_netcdf_variable_denominator_netcdf_variable_logratio(nfreq)` : the multi-channel spectral ratios computed from the two multi-channel power spectra stored the two input NetCDF variables `numerator_netcdf_variable` and `denominator_netcdf_variable`.

This variable is stored only if the **-logratio** argument has been specified when calling `comp_spectrum_ratio_3d`.

All the multi-channel spectral ratios involving the two input NetCDF variables `numerator_netcdf_variable` and `denominator_netcdf_variable` are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with these input NetCDF variables even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For computing the multi-channel spectral ratios and the associated 90% confidence interval for two multi-channel power spectra stored in a NetCDF variable called `sst` extracted, respectively, from the files `spectrum_sst.monthly.1979_2005.nc` and `spectrum_sst.monthly.1950_1978.nc`, use the following command :

```
$ comp_spectrum_ratio_3d -fn=spectrum_sst.monthly.1979_2005.nc \  
                        -fd=spectrum_sst.monthly.1950_1978.nc \  
                        -nv=sst \  
                        -dv=sst \  
                        -o=spectrum_ratio_sst_1979_2005.1950_1978.nc
```

2.61 comp_spectvar_1d

2.61.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.61.2 Latest revision

30/04/2026

2.61.3 Purpose

Compute variance estimates in a selected frequency band for a real time series by windowed filtering in the frequency domain. These variance estimates correspond exactly to the variance of the filtered time series output by `comp_fftfiler_1d` when similar parameters are used in both procedures. Alternatively, `comp_spectvar_1d` can also

compute spectral variance estimates in a given frequency band from a spectrum analysis of a real time series as output by *comp_spectrum_1d* or *comp_cross_spectrum_1d*. The input time series is extracted from a uni- or bidimensional variable readed from a NetCDF dataset.

At the user option, the time series can be detrended before computing the power spectrum and the variance estimate in the selected frequency band (see the **-tr=** and **-tr2=** arguments). The variance estimate and power spectrum are computed with the help of the Fast Fourier Transform (FFT) of the real time series and windowed filtering in the frequency domain.

The user can select different data windows to apply to the time series for the computations of the Power Spectral Density (PSD) estimates (see the **-win2=** argument) and also different frequency windows to convolve with the selected filter response in the frequency domain (see the **-win=** argument) in order to get more accurate variance estimates.

Furthermore, in order to obtain stable PSD and variance estimates, the real time series can be segmented and PSD estimates computed on, eventually overlapping, segments can be averaged at the user option (see the **-sl=** and **-nooverlap=** arguments). The stability of the PSD and final variance estimates is increased by this averaging process. That is, the greater the number of segments, the more stable, the resulting PSD and final variance estimates. See the documentation of *comp_spectrum_1d* or *comp_cross_spectrum_1d* for more details on this method known as the Welch's method [Welch].

Optionally, these PSD estimates may also be smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument) as in a conventional spectrum analysis.

By default, the variance estimate is returned in units, which are the square of the data, and, at the user option, as percentage of the total variance in the selected frequency band if the **-normvar** argument is specified in the call to the procedure.

Finally, the variances estimates can be computed separately for different times intervals (see the **-p=** argument).

The variance estimates are stored in an output NetCDF dataset.

For more information on spectral analysis and filtering in the frequency domain, see the references below [Welch] [Bloomfield] [Diggle] [Iacubucci-Noullez].

2.61.4 Further Details

Usage

```
$ comp_spectvar_1d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -t=time1,time2 (optional) \
  -o=output_netcdf_file (optional) \
  -ni=index_for_2d_netcdf_variable (optional) \
  -p=period_length (optional) \
  -sl=segment_length (optional : 16,32,64,128,...) \
  -pl=minimum_period (optional) \
  -ph=maximum_period (optional) \
  -tr=trend_removal (optional : 0,1,2,3) \
  -tr2=trend_removal_on_segment (optional : 0,1,2,3) \
  -win=window_parameter (optional : 0.5 > 1.) \
  -win2=window_choice (optional : 1,2,3,4,5,6) \
  -tap=tapered_percentage (optional : 0.0 > 1.) \
  -nsm=Daniell_filter_length (optional) \
  -mi=missing_value (optional) \
  -nooverlap (optional) \
  -normvar (optional) \
```

(continues on next page)

(continued from previous page)

-double	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *spectvar_netcdf_variable.nc*
- ni=** if the *netcdf_variable* is bidimensional, the first time series is used
- p=** the *period_length* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time period
- sl=** the *segment_length* is set to 0, which means that the time series (or each time period if the **-p=** argument is used) is not segmented to stabilize the the PSD and final variance estimates
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=** the *trend_removal* is set to 1, which means that the mean of the whole time series is removed before the spectral computations
- tr2=** the *trend_removal_on_segment* is set to 0, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_parameter* is set to 1., which means that the rectangular window is convolved with the ideal filter response in the frequency domain in order to compute the final variance estimates
- win2=** the *window_choice* is set to 2, which means that the square window is applied to the time series and used in the spectral computations
- tap=** the *tapered_percentage* is set to 0.2, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to 6 with the **-win2=** argument), 20% of the data are tapered
- nsm=** the *Daniell_filter_length* is set to 0, which means that the PSD estimates are not smoothed in the frequency domain by a moving average before computing the variance estimates in the selected frequency band
- mi=** the *missing_value* for the output NetCDF variable is set to $1.e+20$
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- normvar** normally, the raw variance in the selected frequency band is output. However, if **-normvar** is specified, the percentage of total variance in the selected frequency band for the real multichannel time series is also computed and stored in the output NetCDF file.
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension (e.g., the segment axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a frequency variance analysis must be performed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.

- 2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to compute the spectral variance estimates in the selected frequency band.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

$time2 - time1 + 1$ must be an even integer if the **-sl=** and **-p=** arguments are not used and of any length if one of these arguments is used.

- 3) The **-ni=index_for_2d_netcdf_variable** argument specifies the index for selecting the time series if the *netcdf_variable* is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *index_for_2d_netcdf_variable* to 1.

- 4) If the **-p=** argument is specified, the frequency variance estimates are computed separately for each time period of length *period_length* (as determined by the value of the **-p=** argument).

The whole selected time period (e.g., $time2 - time1 + 1$) must be a multiple of the *period_length*. The *period_length* must also be of even length if the **-sl=** argument is not used.

- 5) The **-sl=segment_length** argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., $time2 - time1 + 1$) or the *period_length* if the **-p=** argument is used.

Spectral computations are at $(segment_length/2) + 1$ Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies $(i-1) / segment_length$ for $i=1, 2, \dots, (segment_length/2) + 1$).

By default, *segment_length* is set to 0, e.g., the time series is not segmented and the segment length is equal to $time2 - time1 + 1$ or *period_length* if the **-p=** argument is used.

- 6) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 1.

- 7) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 1 and less than the length of the time series or the *period_length* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 8) Setting **-pl=PL**, **-ph=PH** and $PH \ll PL$ is also allowed and performs band rejection of periods between PH and PL. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period *P* is applied to the time series to get the variance estimates at this particular frequency.
- 10) Setting both **-pl=0** and **-ph=0** is not allowed since in that case, no frequency filtering is done.
- 11) The **-tr=** argument specifies pre-filtering processing of the real time series before estimating the power spectrum of the time series. If:
- **-tr=+1**, the mean of the time series is removed before the frequency analysis
 - **-tr=+2**, the drift from the time series is removed before the frequency analysis. The drift for the time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
 - **-tr=+3**, the least-squares line from the time series is removed before the frequency analysis.

For other values of the **-tr=** argument, nothing is done before estimating the spectrum.

If the **-p=** argument is present, the pre-filtering processing is applied to each time period, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0, e.g., nothing is done before the spectrum computations.

- 12) The **-tr2=** argument specifies pre-filtering processing of the real time series before estimating the power spectrum on each time segment (as specified by the **-sl=** argument). If:
- **-tr2=+1**, the mean of the time segment is removed before computing the power spectrum on the segment
 - **-tr2=+2**, the drift from the time segment is removed before computing the power spectrum on the segment
 - **-tr2=+3**, the least-squares line from the time segment is removed before computing the power spectrum on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the spectrum on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 13) The **-win=window_parameter** argument specifies the form of the window which will be convolved with the ideal filter response as in the [comp_fftfiler_1d](#) procedure.

Set **-win=0.5** for using a Hanning window, **-win=0.54** for using a Hamming window or **-win=1.** for rectangular window.

The **-win=window_parameter** argument is a real number greater or equal to 0.5 and less or equal to 1..

By default, rectangular window filtering is used (i.e., **-win=1.**).

- 14) The **-win2=window_choice** argument specifies the form of the data window used in the computations of the power spectrum as the **-win=window_choice** argument in the [comp_spectrum_1d](#) procedure. If:

- **-win2=1** The Bartlett window is used
- **-win2=2** The square window is used
- **-win2=3** The Welch window is used
- **-win2=4** The Hann window is used
- **-win2=5** The Hamming window is used
- **-win2=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6.

The default value for the *window_choice* is 2, e.g., a square window window is used for the spectral computations.

- 15) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win2=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 16) The **-nsm=Daniell_filter_length** argument specifies that the PSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average) as in the *comp_spectrum_1d* procedure.

On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD estimates are not smoothed in the frequency domain.

- 17) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, *comp_spectvar_1d* segments the time series without any overlapping (in each time period if the **-p=** argument is used).

By default, *comp_spectvar_1d* overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting periodograms will be averaged together to obtain PSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 18) The **-normvar** argument specifies that the percentage of variance in the selected frequency band must be also computed and output in addition of the raw variance estimates.

- 19) The **-mi=missing_value** argument specifies the missing value indicator for the output variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 20) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 23) It is assumed that the data has no missing values.

- 24) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 25) For more details on power spectrum analysis and on windowed filtering in the frequency domain, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_spectvar_1d` creates an output NetCDF file that contains the variance estimates in the selected frequency band for the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nperiod` is 1 if the `-p=` argument is not used or is equal to `ntime / period_length` if the `-p=` argument is used, where `ntime = time2 - time1 + 1`):

- 1) `netcdf_variable_spectvar(nperiod)` : the corresponding variance estimates for each time period associated with the input NetCDF variable.
- 2) `netcdf_variable_spectvar_percent(nperiod)` : the corresponding percentages of total variance for each time period associated with the input NetCDF variable.

This variable is stored only if the `-normvar` argument has been specified when calling `comp_spectvar_1d`.

Examples

- 1) For estimating variance in the biennial time scale (e.g., between 18 and 30 months) for a real monthly time series readed from a unidimensional NetCDF variable called ```sst` extracted from the file `sst.monthly.nino34.nc`, which includes a monthly time series, and store the results in the NetCDF file `var_tbo_sst_nino34.nc`, use the following command :

```
$ comp_spectvar_1d \
-f=sst.monthly.nino34.nc \
-v=sst \
-pl=18 \
-ph=30 \
-o=var_tbo_sst_nino34.nc \
-normvar
```

2.62 comp_spectvar_3d

2.62.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.62.2 Latest revision

30/04/2026

2.62.3 Purpose

Compute variance estimates in a selected frequency band for a real multi-channel time series by windowed filtering in the frequency domain. These variance estimates correspond exactly to the variance of the filtered multi-channel time series output by `comp_fftfilter_3d` when similar parameters are used in both procedures. Alternatively, `comp_spectvar_3d` can also compute spectral variance estimates in a given frequency band from a spectrum analysis of a real multi-channel time series as output by `comp_spectrum_1d` or `comp_cross_spectrum_3d`. The input multi-channel time series is extracted from a tridimensional variable readed from a NetCDF dataset

At the user option, the multi-channel time series can be detrended before computing the power spectra and the variance estimates in the selected frequency band (see the **-tr=** and **-tr2=** arguments). The variance estimates and power spectra are computed with the help of the Fast Fourier Transform (FFT) of the real multi-channel time series and windowed filtering in the frequency domain.

The user can select different data windows to apply to the time series for the computations of the Power Spectral Density (PSD) estimates (see the **-win2=** argument) and also different frequency windows to convolve with the selected filter response in the frequency domain (see the **-win=** argument) in order to get more accurate variance estimates.

Furthermore, in order to obtain stable PSD and variance estimates, the real time series can be segmented and PSD estimates computed on, eventually overlapping, segments can be averaged at the user option (see the **-sl=** and **-nooverlap=** arguments). The stability of the PSD and final variance estimates is increased by this averaging process. That is, the greater the number of segments, the more stable, the resulting PSD and final variance estimates. See the documentation of `comp_spectrum_1d` or `comp_cross_spectrum_3d` for more details on this method known as the Welch's method [Welch].

Optionally, theses PSD estimates may also been smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument) as in a conventional spectrum analysis.

By default, the variance estimates are returned in units, which are the square of the data, and, at the user option, as percentage of the total variance in the selected frequency band if the **-normvar** argument is specified in the call to the procedure.

Finally, the variances estimates can be computed separately for different times intervals (see the **-p=** argument).

The variance estimates are stored in an output NetCDF dataset.

For more information on spectral analysis and filtering in the frequency domain, see the references below [Welch] [Bloomfield] [Diggle] [Iacobucci_Noullez].

2.62.4 Further Details

Usage

```
$ comp_spectvar_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file          (optional) \
  -g=grid_type                            (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                            (optional) \
  -y=lat1,lat2                            (optional) \
  -t=time1,time2                          (optional) \
  -o=output_netcdf_file                   (optional) \
```

(continues on next page)

(continued from previous page)

```

-p=period_length                (optional) \
-sl=segment_length              (optional : 16,32,64,128,...) \
-pl=minimum_period              (optional) \
-ph=maximum_period              (optional) \
-tr=trend_removal               (optional : 0,1,2,3) \
-tr2=trend_removal_on_segment  (optional : 0,1,2,3) \
-win=window_parameter           (optional : 0.5 > 1.) \
-win2=window_choice             (optional : 1,2,3,4,5,6) \
-tap=tapered_percentage         (optional : 0.0 > 1.) \
-nsm=Daniell_filter_length      (optional) \
-ngp=number_of_grid_points      (optional) \
-mi=missing_value               (optional) \
-nooverlap                      (optional) \
-normvar                        (optional) \
-double                         (optional) \
-bigfile                       (optional) \
-hdf5                          (optional) \
-cdf5                          (optional) \
-tlimited                       (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *spectvar_netcdf_variable.nc*
- p=** the *period_length* is set to *time2 - time1 + 1*, which means that the time series is considered as continuous with only one time period
- sl=** the *segment_length* is set to 0, which means that the time series (or each time period if the **-p=** argument is used) is not segmented to stabilize the the PSD and final variance estimates
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=** the *trend_removal* is set to 1, which means that the mean of the whole time series is removed before the spectral computations
- tr2=** the *trend_removal_on_segment* is set to 0, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_parameter* is set to 1., which means that the rectangular window is convolved with the ideal filter response in the frequency domain in order to compute the final variance estimates
- win2=** the *window_choice* is set to 2, which means that the square window is applied to the time series and used in the spectral computations
- tap=** the *tapered_percentage* is set to 0.2, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to 6 with the **-win2=** argument), 20% of the data are tapered

- nsm=** the *Daniell_filter_length* is set to 0, which means that the PSD estimates are not smoothed in the frequency domain by a moving average before computing the variance estimates in the selected frequency band
- ngp=** the *number_of_grid_points* is set to the number of points in the selected domain
- mi=** the *missing_value* for the output NetCDF variable is set to $1.e+20$
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- normvar** normally, the raw variance in the selected frequency band is output. However, if **-normvar** is specified, the percentage of total variance in the selected frequency band for the real multi-channel time series is also computed and stored in the output NetCDF file.
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the segment axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a frequency variance analysis must be performed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF *mesh_mask* variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_spectvar_3d*.

- 5) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to compute the spectral variance estimates in the selected frequency band.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

$time2 - time1 + 1$ must be an even integer if the **-sl=** and **-p=** arguments are not used and of any length if one of these arguments is used.

- 6) If the **-p=** argument is specified, the frequency variance estimates are computed separately for each time period of length *period_length* (as determined by the value of the **-p=** argument).

The whole selected time period (e.g., $time2 - time1 + 1$) must be a multiple of the *period_length*. The *period_length* must also be of even length if the **-sl=** argument is not used.

- 7) The **-sl=segment_length** argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., $time2 - time1 + 1$) or the *period_length* if the **-p=** argument is used.

Spectral computations are at $(segment_length/2) + 1$ Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies $(i-1) / segment_length$ for $i=1, 2, \dots, (segment_length/2) + 1$).

By default, *segment_length* is set to 0, e.g., the time series is not segmented and the segment length is equal to $time2 - time1 + 1$ or *period_length* if the **-p=** argument is used.

- 8) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 1.

- 9) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 1 and less than the length of the time series or the *period_length* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 10) Setting **-pl=PL**, **-ph=PH** and $PH \leq PL$ is also allowed and performs band rejection of periods between PH and PL. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.

- 11) Setting **-pl=** and **-ph=** to the same value P is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period P is applied to the time series to get the variance estimates at this particular frequency.

- 12) Setting both **-pl=0** and **-ph=0** is not allowed since in that case, no frequency filtering is done.

- 13) The **-tr=** argument specifies pre-filtering processing of the real multi-channel time series before estimating the power spectrum of the time series. If:

- **-tr=+1**, the mean of the time series is removed before the frequency analysis
- **-tr=+2**, the drift from the time series is removed before the frequency analysis. The drift for the time series is estimated using the formula: $drift = (tseries(ntime) - tseries(1)) / (ntime - 1)$
- **-tr=+3**, the least-squares line from the time series is removed before the frequency analysis.

For other values of the **-tr=** argument, nothing is done before estimating the spectrum.

If the **-p=** argument is present, the pre-filtering processing is applied to each time period, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0, e.g., nothing is done before the spectrum computations.

- 14) The **-tr2=** argument specifies pre-filtering processing of the real multi-channel time series before estimating the power spectrum on each time segment (as specified by the **-sl=** argument). If:

- **-tr2=+1**, the mean of the time segment is removed before computing the power spectrum on the segment
- **-tr2=+2**, the drift from the time segment is removed before computing the power spectrum on the segment
- **-tr2=+3**, the least-squares line from the time segment is removed before computing the power spectrum on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the spectrum on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 15) The **-win=window_parameter** argument specifies the form of the window which will be convolved with the ideal filter response as in the [comp_fftfilter_3d](#) procedure.

Set **-win=0.5** for using a Hanning window, **-win=0.54** for using a Hamming window or **-win=1.** for rectangular window.

The **-win=window_parameter** argument is a real number greater or equal to 0.5 and less or equal to 1..

By default, rectangular window filtering is used (i.e., **-win=1.**).

- 16) The **-win2=window_choice** argument specifies the form of the data window used in the computations of the power spectrum as the **-win=window_choice** argument in the [comp_spectrum_1d](#) procedure. If:

- **-win2=1** The Bartlett window is used
- **-win2=2** The square window is used
- **-win2=3** The Welch window is used
- **-win2=4** The Hann window is used
- **-win2=5** The Hamming window is used
- **-win2=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6.

The default value for the *window_choice* is 2, e.g., a square window window is used for the spectral computations.

- 17) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win2=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 18) The **-nsm=Daniell_filter_length** argument specifies that the PSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average) as in the [comp_spectrum_1d](#) procedure.

On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD estimates are not smoothed in the frequency domain.

- 19) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, [comp_spectvar_3d](#) segments the multi-channel time series without any overlapping (in each time period if the **-p=** argument is used).

By default, [comp_spectvar_3d](#) overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting periodograms will be averaged together to obtain PSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 20) The **-normvar** argument specifies that the percentage of variance in the selected frequency band must be also computed and output in addition of the raw variance estimates.
- 21) The **-ngp=** argument can be used if you have memory problems when running `comp_spectvar_3d` on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.
- 22) The **-mi=missing_value** argument specifies the missing value indicator for the output variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.
- 23) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 24) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 25) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 26) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 27) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 28) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 29) For more details on power spectrum analysis and on windowed filtering in the frequency domain, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>

- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_spectvar_3d` creates an output NetCDF file that contains the variance estimates in the selected frequency band for the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable and `nperiod` is 1 if the `-p=` argument is not used or is equal to `ntime / period_length` if the `-p=` argument is used, where `ntime = time2 - time1 + 1`):

- 1) `netcdf_variable_spectvar(nperiod, nlat, nlon)` : the corresponding variance estimates for each time period and each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_spectvar_percent(nperiod, nlat, nlon)` : the corresponding percentages of total variance for each time period and each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the `-normvar` argument has been specified when calling `comp_spectvar_3d`.

The variance estimates are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For estimating variance in the biennial time scale (e.g., between 18 and 30 months) for a real multi-channel monthly time series readed from a tridimensional NetCDF variable called `mslp` extracted from the file `mslp.monthly.mean_ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `var_tbo_mslp_ncep2.nc`, use the following command :

```
$ comp_spectvar_3d \
-f=mslp.monthly.mean_ncep2.nc \
-v=mslp \
-m=mesh_mask_mslp_ncep2.nc \
-pl=18 \
-ph=30 \
-o=var_tbo_mslp_ncep2.nc
```

2.63 comp_spectvar_4d

2.63.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.63.2 Latest revision

30/04/2026

2.63.3 Purpose

Compute variance estimates in a selected frequency band for a real multi-channel time series by windowed filtering in the frequency domain. These variance estimates correspond exactly to the variance of the filtered multi-channel time series output by `comp_fffilter_4d` when similar parameters are used in both procedures. Alternatively, `comp_spectvar_4d` can also compute spectral variance estimates in a given frequency band from a spectrum analysis of a real multi-channel time series as output by `comp_spectrum_1d` or `comp_cross_spectrum_3d`. The input multi-channel time series is extracted from a fourdimensional variable readed from a NetCDF dataset.

At the user option, the multi-channel time series can be detrended before computing the power spectra and the variance estimates in the selected frequency band (see the **-tr=** and **-tr2=** arguments). The variance estimates and power spectra are computed with the help of the Fast Fourier Transform (FFT) of the real multi-channel time series and windowed filtering in the frequency domain.

The user can select different data windows to apply to the time series for the computations of the Power Spectral Density (PSD) estimates (see the **-win2=** argument) and also different frequency windows to convolve with the selected filter response in the frequency domain (see the **-win=** argument) in order to get more accurate variance estimates.

Furthermore, in order to obtain stable PSD and variance estimates, the real time series can be segmented and PSD estimates computed on, eventually overlapping, segments can be averaged at the user option (see the **-sl=** and **-nooverlap=** arguments). The stability of the PSD and final variance estimates is increased by this averaging process. That is, the greater the number of segments, the more stable, the resulting PSD and final variance estimates. See the documentation of `comp_spectrum_1d` or `comp_cross_spectrum_3d` for more details on this method known as the Welch's method [Welch].

Optionally, theses PSD estimates may also been smoothed in the frequency domain by Daniell weights (e.g., a simple moving average; see the **-nsm=** argument) as in a conventional spectrum analysis.

By default, the variance estimates are returned in units, which are the square of the data, and, at the user option, as percentage of the total variance in the selected frequency band if the **-normvar** argument is specified in the call to the procedure.

Finally, the variances estimates can be computed separately for different times intervals (see the **-p=** argument).

The variance estimates are stored in an output NetCDF dataset.

For more information on spectral analysis and filtering in the frequency domain, see the references below [Welch] [Bloomfield] [Diggle] [Iacobucci_Noullez].

2.63.4 Further Details

Usage

```
$ comp_spectvar_4d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file           (optional) \
  -g=grid_type                             (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                             (optional) \
  -y=lat1,lat2                             (optional) \
  -z=level1,level2                         (optional) \
  -t=time1,time2                           (optional) \
```

(continues on next page)

(continued from previous page)

```

-o=output_netcdf_file           (optional) \
-p=period_length                (optional) \
-sl=segment_length              (optional : 16,32,64,128,...) \
-pl=minimum_period              (optional) \
-ph=maximum_period              (optional) \
-tr=trend_removal               (optional : 0,1,2,3) \
-tr2=trend_removal_on_segment  (optional : 0,1,2,3) \
-win=window_parameter           (optional : 0.5 > 1.) \
-win2=window_choice             (optional : 1,2,3,4,5,6) \
-tap=tapered_percentage        (optional : 0.0 > 1.) \
-nsm=Daniell_filter_length      (optional) \
-ngp=number_of_grid_points      (optional) \
-mi=missing_value               (optional) \
-nooverlap                      (optional) \
-normvar                        (optional) \
-double                         (optional) \
-bigfile                        (optional) \
-hdf5                           (optional) \
-cdf5                           (optional) \
-tlimited                        (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *spectvar_netcdf_variable.nc*
- p=** the *period_length* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time period
- sl=** the *segment_length* is set to 0, which means that the time series (or each time period if the **-p=** argument is used) is not segmented to stabilize the the PSD and final variance estimates
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=** the *trend_removal* is set to 1, which means that the mean of the whole time series is removed before the spectral computations
- tr2=** the *trend_removal_on_segment* is set to 0, which means that no trend processing is done on each time segment before the spectral computations
- win=** the *window_parameter* is set to 1., which means that the rectangular window is convolved with the ideal filter response in the frequency domain in order to compute the final variance estimates
- win2=** the *window_choice* is set to 2, which means that the square window is applied to the time series and used in the spectral computations

- tap=** the *tapered_percentage* is set to 0.2, which means that if a split-cosine-bell window is used for the spectral computations (e.g., if the *window_choice* is set to 6 with the **-win2=** argument), 20% of the data are tapered
- nsm=** the *Daniell_filter_length* is set to 0, which means that the PSD estimates are not smoothed in the frequency domain by a moving average before computing the variance estimates in the selected frequency band
- ngp=** the *number_of_grid_points* is set to the number of points in the selected domain
- mi=** the *missing_value* for the output NetCDF variable is set to 1.e+20
- nooverlap** normally, if the **-sl=** argument is used, the time segments are overlapped. However, if **-nooverlap** is specified, the time segments are not overlapped
- normvar** normally, the raw variance in the selected frequency band is output. However, if **-normvar** is specified, the percentage of total variance in the selected frequency band for the real multi-channel time series is also computed and stored in the output NetCDF file.
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension (e.g., the segment axis) is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a frequency variance analysis must be performed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF *mesh_mask* variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from *nlon+lon1+1* to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_spectvar_4d*.

- 5) If the **-t=***time1,time2* argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

time2 - time1 + 1 must be an even integer if the **-sl=** and **-p=** arguments are not used and of any length if one of these arguments is used.

- 6) If the **-p=** argument is specified, the frequency variance estimates are computed separately for each time period of length *period_length* (as determined by the value of the **-p=** argument).

The whole selected time period (e.g., *time2 - time1 + 1*) must be a multiple of the *period_length*. The *period_length* must also be of even length if the **-sl=** argument is not used.

- 7) The **-sl=***segment_length* argument specifies an integer used to segment the time series. The *segment_length* must be a positive even integer less or equal to the length of the time series (e.g., *time2 - time1 + 1*) or the *period_length* if the **-p=** argument is used.

Spectral computations are at (*segment_length/2*) + 1 Fourier frequencies and suggested values for *segment_length* are 16, 32, 64, 128, ... (e.g., the spectral estimates are taken at frequencies (i-1) / *segment_length* for i=1,2,...,(*segment_length/2*) + 1).

By default, *segment_length* is set to 0, e.g., the time series is not segmented and the segment length is equal to *time2 - time1 + 1* or *period_length* if the **-p=** argument is used.

- 8) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH** .

The **-pl=** argument is a positive integer equal to 0 or greater than 1.

- 9) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL** . For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 1 and less than the length of the time series or the *period_length* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 10) Setting **-pl=PL**, **-ph=PH** and $PH \ll PL$ is also allowed and performs band rejection of periods between PH and PL. In that case, the meaning of the **-pl=** and **-ph=** arguments reversed.

- 11) Setting **-pl=** and **-ph=** to the same value P is allowed. In this case, an -ideal- band-pass filter with peak response near one at the single period P is applied to the time series to get the variance estimates at this particular frequency.

- 12) Setting both **-pl=0** and **-ph=0** is not allowed since in that case, no frequency filtering is done.

- 13) The **-tr=** argument specifies pre-filtering processing of the real multi-channel time series before estimating the power spectrum of the time series. If:

- **-tr=+1**, the mean of the time series is removed before the frequency analysis
- **-tr=+2**, the drift from the time series is removed before the frequency analysis. The drift for the time series is estimated using the formula: $drift = (tseries(ntime) - tseries(1)) / (ntime - 1)$
- **-tr=+3**, the least-squares line from the time series is removed before the frequency analysis.

For other values of the **-tr=** argument, nothing is done before estimating the spectrum.

If the **-p=** argument is present, the pre-filtering processing is applied to each time period, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0, e.g., nothing is done before the spectrum computations.

- 14) The **-tr2=** argument specifies pre-filtering processing of the real multi-channel time series before estimating the power spectrum on each time segment (as specified by the **-sl=** argument). If:
- **-tr2=+1**, the mean of the time segment is removed before computing the power spectrum on the segment
 - **-tr2=+2**, the drift from the time segment is removed before computing the power spectrum on the segment
 - **-tr2=+3**, the least-squares line from the time segment is removed before computing the power spectrum on the segment.

For other values of the **-tr2=** argument, nothing is done before estimating the spectrum on each time segment.

The **-tr2=** argument must be an integer and the default value for the **-tr2=** argument is 0, e.g., nothing is done before the spectral computations on each time segment.

- 15) The **-win=window_parameter** argument specifies the form of the window which will be convolved with the ideal filter response as in the [comp_fftfiler_4d](#) procedure.

Set **-win=0.5** for using a Hanning window, **-win=0.54** for using a Hamming window or **-win=1.** for rectangular window.

The **-win=window_parameter** argument is a real number greater or equal to 0.5 and less or equal to 1..

By default, rectangular window filtering is used (i.e., **-win=1.**).

- 16) The **-win2=window_choice** argument specifies the form of the data window used in the computations of the power spectrum as the **-win=window_choice** argument in the [comp_spectrum_1d](#) procedure. If:
- **-win2=1** The Bartlett window is used
 - **-win2=2** The square window is used
 - **-win2=3** The Welch window is used
 - **-win2=4** The Hann window is used
 - **-win2=5** The Hamming window is used
 - **-win2=6** A split-cosine-bell window is used.

The *window_choice* must be an integer between 1 and 6.

The default value for the *window_choice* is 2, e.g., a square window window is used for the spectral computations.

- 17) The **-tap=tapered_percentage** argument specifies the total percentage of the data to be tapered if the *window_choice* is set to 6 with the **-win2=** argument. The *tapered_percentage* is a real number which must be greater than 0. and less or equal to 1..

The default value for the *tapered_percentage* is 0.2 (e.g., 20% of the data are tapered, 10% at the start of the series and 10% at the end).

- 18) The **-nsm=Daniell_filter_length** argument specifies that the PSD estimates must be computed by smoothing the periodogram with Daniell weights (e.g., a simple moving average) as in the [comp_spectrum_1d](#) procedure.

On entry, the **-nsm=** argument gives the length of the Daniell filter to be applied. The *Daniell_filter_length* must be an odd integer, greater than 2 and less or equal to $(segment_length/2) + 1$ if the **-sl=** argument is specified.

By default, the PSD estimates are not smoothed in the frequency domain.

- 19) If the **-nooverlap** argument is specified and the **-sl=** argument is also specified, `comp_spectvar_4d` segments the multi-channel time series without any overlapping (in each time period if the **-p=** argument is used).

By default, `comp_spectvar_4d` overlaps the segments by one half of their length (which is equal to the value of the **-sl=** argument).

In both cases, each segment will be FFT'd, and the resulting periodograms will be averaged together to obtain PSD estimates at the $(segment_length/2) + 1$ frequencies.

If the **-sl=** argument is not specified, the **-nooverlap** argument has no effect.

- 20) The **-normvar** argument specifies that the percentage of variance in the selected frequency band must be also computed and output in addition of the raw variance estimates.
- 21) The **-ngp=** argument can be used if you have memory problems when running `comp_spectvar_4d` on very large datasets. By default, the `number_of_grid_points` is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.
- 22) The **-mi=missing_value** argument specifies the missing value indicator for the output variables in the `output_netcdf_file`.

If the **-mi=** argument is not specified, the `missing_value` is set to `1.e+20`.

- 23) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 24) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the `output_netcdf_file` will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 25) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 26) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 27) It is assumed that the data has no missing values excepted those associated with a constant land-sea-depth mask.
- 28) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 29) For more details on power spectrum analysis and on windowed filtering in the frequency domain, see

- “The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”, by P.D. Welch, IEEE trans. on audio and electroacoustics, 15:2, 70-73, 1967. DOI: <https://doi.org/10.1109/TAU.1967.1161901>

- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Time series: a biostatistical introduction”, by P.J. Diggle, Clarendon Press, Oxford, Chap. 4, 1990. <http://www.oupcanada.com/catalog/9780198522263.html>
- “A Frequency Selective Filter for Short-Length Time Series”, by Iacobucci, A., and Noullez, A., Computational Economics, Vol. 25, 75-102, 2005. <https://link.springer.com/article/10.1007/s10614-005-6276-7>

Outputs

`comp_spectvar_4d` creates an output NetCDF file that contains the variance estimates in the selected frequency band for the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable and `nperiod` is 1 if the **-p=** argument is not used or is equal to `ntime / period_length` if the **-p=** argument is used, where `ntime = time2 - time1 + 1`) :

- 1) `netcdf_variable_spectvar(nperiod,nlev,nlat,nlon)` : the corresponding variance estimates for each time period and each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_spectvar_percent(nperiod,nlev,nlat,nlon)` : the corresponding percentages of total variance for each time period and each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-normvar** argument has been specified when calling `comp_spectvar_4d`.

The variance estimates are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For estimating variance in the biennial time scale (e.g., between 18 and 30``months) for a real multi-channel monthly time series readed from a fourdimensional NetCDF variable called ``uwnd extracted from the file `uwnd.monthly.mean_ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `var_qbo_uwnd_ncep2.nc`, use the following command :

```
$ comp_spectvar_4d \  
-f=uwnd.monthly.mean_ncep2.nc \  
-v=uwnd \  
-m=mesh_mask_uwnd_ncep2.nc \  
-pl=18 \  
-ph=30 \  
-o=var_qbo_uwnd_ncep2.nc
```


2.64 comp_stat_3d

2.64.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.64.2 Latest revision

29/04/2026

2.64.3 Purpose

Compute univariate statistics from a tridimensional variable extracted from a NetCDF dataset and, optionally, the associated mesh-mask and scale factors of the 2-D grid-mesh associated with the input NetCDF variable.

Mean, variance, standard-deviation, skewness, kurtosis, minimum and maximum are computed for each point in the multi-channel time series of the 2-D grid-mesh associated with the input NetCDF variable. These univariate statistics may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

The mean is a simple, but informative, measure, of the central tendency of a variable [vonStorch_Zwiers]. The standard-deviation and variance are conventional measures of variation of a variable [vonStorch_Zwiers]. If $X(:)$ is a vector of *ntime* observations for one grid-point in the multi-channel time series of the 2-D grid-mesh, the **MEAN**, **VAR** (e.g., variance) and **STD** (e.g., standard-deviation) statistics for this grid-point are estimated by:

- **MEAN** = $\text{sum}(X(:))/ntime$
- **VAR** = $\text{sum}([X(:)-\text{MEAN}]^2)/(ntime-1)$
- **STD** = $\text{sqrt}(\text{VAR})$

Note that the divisor used in calculating variance and standard-deviation is the number of degrees of freedom (e.g., the number of observations minus 1), this is in contrast with the formulae used in *comp_clim_3d*.

Skewness measures the deviation of the distribution of a variable from symmetry [vonStorch_Zwiers] [Burgers_Stephenson] [White] [Masson_etal]. For a symmetrical distribution, the skewness coefficient is always equal to 0, but the converse is not true. Skewness is 0 for a normal distribution. For unimodal distributions shifted to the right (left), the skewness coefficient is positive (negative).

If the argument **-nobias** is specified, the skewness coefficient is estimated as

$$\text{SKEWNESS} = (ntime.M3)/[(ntime-1).(ntime-2).STD3]$$

, otherwise the following (biased) classical formulae is used

$$\text{SKEWNESS} = M3/(ntime.STD3)$$

where

- **M3** is equal to the sum of the deviations of the observations from the mean raised to the third power (e.g., $\text{sum}([X(:)-\text{MEAN}]^3)$ where $X(:)$ is the vector of observations)
- **STD3** is the standard deviation raised to the third power

In order to interpret correctly the skewness of a variable, note that the Standard Error (SE) of the skewness coefficient calculated from a sample drawn from a Gaussian distribution is given by

$$SE[\text{SKEWNESS}] = \text{sqrt}([6.ntime.(ntime-1)]/[(ntime-2).(ntime+1).(ntime+3)])$$

$SE[\text{SKEWNESS}]$ is not very different from the quantity $\sqrt{6/ntime}$ when the number of observations is sufficiently high.

Moreover, the quantity $\text{SKEWNESS}/SE[\text{SKEWNESS}]$ follows asymptotically a normal (e.g., Gaussian) distribution with mean 0 and variance equal to 1 when the sample $X(\cdot)$ were independent Gaussian observations. With a sample of independent Gaussian observations, a value twice the standard error is thus associated with a 5% significance level. However, in climate analysis the observations are in general autocorrelated.

Kurtosis measures the flatness or peakedness of the distribution of a variable [vonStorch_Zwiers] [Burgers_Stephenson] [White]. As computed by `comp_stat_3d`, the kurtosis coefficient is always greater or equal to -2 and is equal to 0 for a normal distribution. In most cases, if the kurtosis is greater (lower) than 0 then the distribution is more peaked (flatter) than the normal distribution with the same mean and standard-deviation.

If the argument **-nobias** is specified, the kurtosis coefficient is estimated as

$$\text{KURTOSIS} = A - 3. \frac{(ntime-1).(ntime-1)}{(ntime-2).(ntime-3)}$$

, otherwise the following (biased) classical formulae is used

$$\text{KURTOSIS} = M4/(ntime.STD4) - 3$$

where

- **M4** is equal to the sum of the deviations of the observations from the mean raised to the fourth power (e.g., $\sum (X(\cdot) - \text{MEAN})^{**4}$) where $X(\cdot)$ is the vector of observations)
- **STD4** is the standard deviation raised to the fourth power
- **A** is equal to $(ntime.(ntime+1).M4)/[(ntime-1).(ntime-2).(ntime-3).STD4]$

In order to interpret correctly the kurtosis of a variable, note that the SE of the kurtosis coefficient calculated from a sample drawn from a Gaussian distribution is given by

$$SE[\text{KURTOSIS}] = \sqrt{B/C}$$

where

- **B** is equal to $24.ntime.(ntime-1).(ntime-1)$
- **C** is equal to $(ntime-3).(ntime-2).(ntime+3).(ntime+5)$

and the quantity $\text{KURTOSIS}/SE[\text{KURTOSIS}]$ follows also asymptotically a normal distribution with mean 0 and variance equal to 1 when the sample $X(\cdot)$ were independent Gaussian observations.

Extreme departures from the mean will cause very high values of kurtosis. Consequently, the kurtosis coefficient can be used to detect outliers. High values of kurtosis can also be a result of one or two extreme observations in a sample of observations.

The standard errors of the skewness and kurtosis coefficients are calculated and stored in the output NetCDF file if the argument **-stderror** is specified.

Moreover, the two-tailed significance levels of the statistics $\text{SKEWNESS}/SE[\text{SKEWNESS}]$ and $\text{KURTOSIS}/SE[\text{KURTOSIS}]$ (under the hypothesis that the sample $X(\cdot)$ were independent Gaussian observations) are calculated and stored in the output NetCDF file if the argument **-prob** is specified.

Thus, if you are interested in how well the distribution of the time series in the 2-D grid-mesh associated with the input NetCDF variable can be approximated by the normal distribution, the skewness and kurtosis coefficients, their standard errors and significance levels, can give you some useful information on this issue.

If your data contains missing values, use `comp_stat_miss_3d` instead of `comp_stat_3d` to estimate univariate statistics from your gappy dataset.

If the NetCDF variable is fourdimensional use `comp_stat_4d` instead of `comp_stat_3d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro. However, in this version of `comp_stat_3d`, parallelism is on the number of periods as determined by the `-p=periodicity` argument. This means that the number of processors or threads must not be greater than the *periodicity* parameter.

Moreover, this procedure computes all the univariate statistics with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence 2-D mask and scale factor variables, which may be used to compute the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures such as *comp_serie_3d*, *comp_eof_3d*, etc. When computing the scale factors, it is assumed that the 2-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

2.64.4 Further Details

Usage

```
$ comp_stat_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -p=periodicity           (optional) \
  -x=lon1,lon2             (optional) \
  -y=lat1,lat2             (optional) \
  -t=time1,time2           (optional) \
  -o=output_statistics_netcdf_file (optional) \
  -m=output_mesh_mask_netcdf_file (optional) \
  -yl=latl1,latl2          (optional) \
  -mi=missing_value        (optional) \
  -nobias                   (optional) \
  -stderror                 (optional) \
  -prob                     (optional) \
  -double                   (optional) \
  -bigfile                  (optional) \
  -hdf5                     (optional) \
  -cdf5                     (optional) \
  -tlimited                  (optional)
```

By default

- `-p=` the *periodicity* is equal to 1
- `-x=` the whole longitude domain associated with the *netcdf_variable*
- `-y=` the whole latitude domain associated with the *netcdf_variable*
- `-t=` the whole time period associated with the *netcdf_variable*
- `-o=` the *output_statistics_netcdf_file* is named `stat_netcdf_variable.nc`
- `-m=` the *output_mesh_mask_netcdf_file* is not created
- `-yl=` it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors

- mi=** the *missing_value* for the statistics variables in the output NetCDF file is set to $1.e+20$
- nobias** biased estimates of kurtosis and skewness coefficients are computed. However, if **-nobias** is activated, unbiased estimates of kurtosis and skewness are computed
- stderror** the standard errors of the kurtosis and skewness coefficients are not computed. However, if **-stderror** is activated, these standard errors are computed
- prob** the significance levels of the kurtosis and skewness coefficients are not computed. However, if **-prob** is activated, these significance levels are computed
- double** the statistic variables are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the statistic variables are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which univariate statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, statistics are computed for all the points of the 2-D grid-mesh associated with the *netcdf_variable*.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_stat_3d*.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the statistics.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.

- 5) It is assumed that the data has no missing values (excepted missing values associated with a constant land-sea mask as indicated by a *missing_value* or *_FillValue* attribute).

If it is the case, use [comp_stat_miss_3d](#) instead of *comp_stat_3d*.

- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 2-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 2-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 7) If the **-m=output_mesh_mask_netcdf_file** argument is present and if some scale factors can not be computed, these scale factors are set to 1.

- 8) The **-mi=missing_value** argument specifies the missing value indicator for the variance (VAR), standard-deviation (STD), skewness (SKEW) and kurtosis (KURT) variables in the *output_statistics_netcdf_file*.

If the **-mi=** argument is not specified and the NetCDF variable has a *missing_value* or *_FillValue* attribute, the *missing_value* is set to $1.e+20$. This argument is not used if the NetCDF variable specified in the **-v=** argument has no *missing_value* or *_FillValue* attribute (excepted for the skewness and kurtosis statistics).

- 9) If **-nobias** is specified, unbiased estimates of skewness and kurtosis are computed.

If the **-nobias** argument is absent, the biased standard estimates are computed.

- 10) If **-stderror** is specified, the standard errors of skewness and kurtosis are computed.

If the **-stderror** argument is absent, the standard errors are not computed.

- 11) If **-prob** is specified, the significance levels of skewness and kurtosis are computed. Moreover, the **-prob** argument implies also the **-stderror** argument, even if this argument is not activated.

If the **-prob** argument is absent, the significance levels are not computed.

- 12) At least 4 observations by period, as determined from the **-p=periodicity** argument, are required, otherwise the program will stop.

- 13) The **-double** argument specifies that the VAR, STD, SKEW and KURT variables must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_statistics_netcdf_file*.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 18) For more details and examples on the use of skewness and kurtosis coefficients in the climate literature see
- “Skewness, Kurtosis and Extreme Values of Northern Hemisphere Geopotential Heights”, by White, G., Monthly Weather Review, Vol. 108, 1446-1455, 1980. doi: [10.1175/1520-0493\(1980\)108<1446:SKAEVO>2.0.CO;2](https://doi.org/10.1175/1520-0493(1980)108<1446:SKAEVO>2.0.CO;2)
 - “The Normality of El Nino”, by Burgers, G., and Stephenson, D.B , Geophysical Research Letters, 26, 1027-1030, 1999. doi: [10.1029/1999GL900161](https://doi.org/10.1029/1999GL900161)
 - “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by Masson, S., et al., Climate Dynamics, Vol. 39, 681-707, 2012. doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](https://www.amazon.com/Statistical-Analysis-Climate-Research/dp/0521012300)

Outputs

`comp_stat_3d` creates an output NetCDF file that contains the univariate statistics and number of observations of the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_mean(periodicity,nlat,nlon)` : the means for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_var(periodicity,nlat,nlon)` : the variances for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_std(periodicity,nlat,nlon)` : the standard-deviations for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 4) `netcdf_variable_skew(periodicity,nlat,nlon)` : the skewness for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 5) `netcdf_variable_kurt(periodicity,nlat,nlon)` : the kurtosis for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 6) `netcdf_variable_min(periodicity,nlat,nlon)` : the minima for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 7) `netcdf_variable_max(periodicity,nlat,nlon)` : the maxima for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 8) `netcdf_variable_skew_prob(periodicity,nlat,nlon)` : the significance levels of the skewness coefficients.

This variable is stored only if the `-prob` argument has been specified when calling `comp_stat_3d`.

- 9) `netcdf_variable_kurt_prob(periodicity,nlat,nlon)` : the significance levels of the kurtosis coefficients.

This variable is stored only if the `-prob` argument has been specified when calling `comp_stat_3d`.

- 10) `netcdf_variable_nobs(periodicity)` : the number of observations used to compute the statistics.

- 11) *netcdf_variable_skew_se*(periodicity) : the standard-errors of the skewness coefficients.

This variable is stored only if the **-stderr** or **-prob** arguments have been specified when calling *comp_stat_3d*.

- 12) *netcdf_variable_kurt_se*(periodicity) : the standard-errors of the kurtosis coefficients.

This variable is stored only if the **-stderr** or **-prob** arguments have been specified when calling *comp_stat_3d*.

All the statistics and associated probabilities, excepted the number of observations and the standard-errors of the skewness and kurtosis coefficients are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values.

Optionally, *comp_stat_3d* can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) *netcdf_variable_nmask*(nlat, nlon) : a presence-absence or land-sea 2-D mask associated with the input NetCDF variable.
- 2) *netcdf_variable_e1n*(nlat, nlon) : the first scale factor associated with the 2-D grid-mesh of the input NetCDF variable.
- 3) *netcdf_variable_e2n*(nlat, nlon) : the second scale factor associated with the 2-D grid-mesh of the input NetCDF variable.

Multiplying the two scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly statistics from the NetCDF file *ST7_1m_00101_20012_grid_T_sosstsst.nc*, which includes a NetCDF variable *sosstsst*, and store the results in a NetCDF file named *stat_ST7_1m_00101_20012_grid_T_sosstsst.nc*, use the following command :

```
$ comp_stat_3d \
-f=ST7_1m_00101_20012_grid_T_sosstsst.nc \
-v=sosstsst \
-p=12 \
-o=stat_ST7_1m_00101_20012_grid_T_sosstsst.nc
```

- 2) For computing monthly unbiased univariate statistics from the NetCDF file *sst.mnmean.nc*, which includes a NetCDF variable *sst*, and store the results in a NetCDF file named *stat_sst.nc* and, in addition, generate an associated *mesh_mask_netcdf_file* named *mesh_mask_sst.nc*, use the following command :

```
$ comp_stat_3d \
-f=sst.mnmean.nc \
-v=sst \
-p=12 \
-nobias \
-m=mesh_mask_sst.nc
```

2.65 comp_stat_4d

2.65.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.65.2 Latest revision

29/04/2026

2.65.3 Purpose

Compute univariate statistics from a fourdimensional variable extracted from a NetCDF dataset and, optionally, the associated mesh-mask and scale factors of the 3-D grid-mesh associated with the input NetCDF variable.

Mean, variance, standard-deviation, skewness, kurtosis, minimum and maximum are computed for each point in the multi-channel time series of the 3-D grid-mesh associated with the input NetCDF variable. These univariate statistics may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

The mean is a simple, but informative, measure, of the central tendency of a variable. The standard-deviation and variance are conventional measures of variation of a variable. If $X(:)$ is a vector of $ntime$ observations for one grid-point in the multi-channel time series of the 3-D grid-mesh, the **MEAN**, **VAR** (e.g., variance) and **STD** (e.g., standard-deviation) statistics for this grid-point are estimated by:

- **MEAN** = $\text{sum}(X(:))/ntime$
- **VAR** = $\text{sum}([X(:)-\text{MEAN}]^2)/(ntime-1)$
- **STD** = $\text{sqrt}(\text{VAR})$

Note that the divisor used in calculating variance and standard-deviation is the number of degrees of freedom (e.g., the number of observations minus 1), this is in contrast with the formulae used in *comp_clim_4d*.

Skewness measures the deviation of the distribution of a variable from symmetry [vonStorch_Zwiers] [Burgers_Stephenson] [White] [Masson_etal]. For a symmetrical distribution, the skewness coefficient is always equal to 0, but the converse is not true. Skewness is 0 for a normal distribution. For unimodal distributions shifted to the right (left), the skewness coefficient is positive (negative).

If the argument **-nobias** is specified, the skewness coefficient is estimated as

SKEWNESS = $(ntime.M3)/[(ntime-1).(ntime-2).STD3]$

, otherwise the following (biased) formulae is used:

SKEWNESS = $M3/(ntime.STD3)$

where

- **M3** is equal to the sum of the deviations of the observations from the mean raised to the third power (e.g., $\text{sum}([X(:)-\text{MEAN}]^3)$ where $X(:)$ is the vector of observations)
- **STD3** is the standard deviation raised to the third power

In order to interpret correctly, the skewness of a variable, note that the Standard Error (SE) of the skewness coefficient is given by

$SE[\text{SKEWNESS}] = \text{sqrt}([6.ntime.(ntime-1)]/[(ntime-2).(ntime+1).(ntime+3)])$

$SE[\text{SKEWNESS}]$ is not very different from the quantity $\sqrt{6/ntime}$ when the number of observations is sufficiently high.

Moreover, the quantity $\text{SKEWNESS}/SE[\text{SKEWNESS}]$ follows asymptotically a normal (e.g., Gaussian) distribution with mean 0 and variance equal to 1 when the sample $X(\cdot)$ were independent Gaussian observations. With a sample of independent Gaussian observations, a value twice the standard error is thus associated with a 5% significance level. However, in climate analysis the observations are in general autocorrelated.

Kurtosis measures the flatness or peakedness of the distribution of a variable [vonStorch_Zwiers] [Burgers_Stephenson] [White]. As computed by `comp_stat_3d`, the kurtosis coefficient is always greater or equal to -2 and is equal to 0 for a normal distribution. In most cases, if the kurtosis is greater (lower) than 0 then the distribution is more peaked (flatter) than the normal distribution with the same mean and standard-deviation.

If the argument **-nobias** is specified, the kurtosis coefficient is estimated as

$$\text{KURTOSIS} = A - 3. \frac{(ntime-1).(ntime-1)}{(ntime-2).(ntime-3)}$$

, otherwise the following (biased) formulae is used:

$$\text{KURTOSIS} = M4/(ntime.STD4) - 3$$

where

- **M4** is equal to the sum of the deviations of the observations from the mean raised to the fourth power (e.g., $\sum (X(\cdot) - \text{MEAN})^{**4}$) where $X(\cdot)$ is the vector of observations)
- **STD4** is the standard deviation raised to the fourth power
- **A** is equal to $(ntime.(ntime+1).M4)/[(ntime-1).(ntime-2).(ntime-3).STD4]$

In order to interpret correctly the kurtosis of a variable, note that the SE of the kurtosis coefficient calculated from a sample drawn from a Gaussian distribution is given by

$$SE[\text{KURTOSIS}] = \sqrt{B/C}$$

where

- **B** is equal to $24.ntime.(ntime-1).(ntime-1)$
- **C** is equal to $(ntime-3).(ntime-2).(ntime+3).(ntime+5)$

and the quantity $\text{KURTOSIS}/SE[\text{KURTOSIS}]$ follows also asymptotically a normal distribution with mean 0 and variance equal to 1 when the sample $X(\cdot)$ were independent Gaussian observations.

Extreme departures from the mean will cause very high values of kurtosis. Consequently, the kurtosis coefficient can be used to detect outliers. High values of kurtosis can also be a result of one or two extreme observations in a sample of observations.

The standard errors of the skewness and kurtosis coefficients are calculated and stored in the output NetCDF file if the argument **-stderror** is specified.

Moreover, the two-tailed significance levels of the statistics $\text{SKEWNESS}/SE[\text{SKEWNESS}]$ and $\text{KURTOSIS}/SE[\text{KURTOSIS}]$ (under the hypothesis that the sample $X(\cdot)$ were independent Gaussian observations) are calculated and stored in the output NetCDF file if the argument **-prob** is specified.

Thus, if you are interested in how well the distribution of the time series in the 3-D grid-mesh associated with the input NetCDF variable can be approximated by the normal distribution, the skewness and kurtosis coefficients, their standard errors and significance levels, can give you some useful information on this issue.

If the NetCDF variable is tridimensional use `comp_stat_3d` instead of `comp_stat_4d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro (however, in this version of `comp_stat_4d`, parallelism is on the number of periods as determined by the **-p=periodicity** argument. This means that the number of processors or threads must not be greater than the *periodicity* parameter).

Moreover, this procedure computes all the univariate statistics with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence 3-D mask and scale factor variables which may be used to compute the surface or volume of each cell in the 3-D grid-mesh associated with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures such as *comp_serie_4d*, *comp_eof_4d*, etc. When computing the scale factors, it is assumed that the 3-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

2.65.4 Further Details

Usage

```
$ comp_stat_4d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-p=periodicity (optional) \  
-x=lon1,lon2 (optional) \  
-y=lat1,lat2 (optional) \  
-z=level1,level2 (optional) \  
-t=time1,time2 (optional) \  
-o=output_statistics_netcdf_file (optional) \  
-m=output_mesh_mask_netcdf_file (optional) \  
-vz=name_of_the_vertical_netcdf_variable (optional) \  
-z0=value_of_the_highest_level (optional) \  
-sf=method_for_computing_the_third_scale_factor (optional : method1, method2) \  
-yl=latl1,latl2 (optional) \  
-mi=missing_value (optional) \  
-nobias (optional) \  
-stderror (optional) \  
-prob (optional) \  
-double (optional) \  
-bigfile (optional) \  
-hdf5 (optional) \  
-cdf5 (optional) \  
-tlimited (optional)
```

By default

- p=** the *periodicity* is equal to 1
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_statistics_netcdf_file* is named `stat_netcdf_variable.nc`
- m=** the *output_mesh_mask_netcdf_file* is not created
- vz=** the variable with the same name as the third dimension, if any (e.g., the associated coordinate variable)

- z0=** a value of 0 is assumed for the highest level when computing the third scale factor
- sf=** *method2* (e.g., it is assumed that each level or depth is located in the middle of each layer and the third scale factor is computed accordingly)
- yl=** it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors
- mi=** the *missing_value* for the statistics variables in the output NetCDF file is set to $1.e+20$
- nobias** biased estimates of kurtosis and skewness coefficients are computed. However, if **-nobias** is activated, unbiased estimates of kurtosis and skewness are computed
- stderror** the standard errors of the kurtosis and skewness coefficients are not computed. However, if **-stderror** is activated, these standard errors are computed
- prob** the significance levels of the kurtosis and skewness coefficients are not computed. However, if **-prob** is activated, these significance levels are computed
- double** the statistic variables are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the statistics are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which univariate statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.
- 3) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, statistics are computed for all the points of the 3-D grid-mesh associated with the *netcdf_variable*.

The longitude, latitude or depth range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_stat_4d*.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the statistics.
The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) It is assumed that the data has no missing values (excepted missing values associated with a constant land-sea mask as indicated by a *missing_value* or *_FillValue* attribute).

- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 3-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 3-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 7) If the **-m=output_mesh_mask_netcdf_file** argument is present, the third scale factor is computed with the help of the vertical coordinate variable (or the **-vz=name_of_the_vertical_netcdf_variable**) if this vertical coordinate variable is strictly monotonic.
- 8) The **-z0=value_of_the_highest_level** argument specifies a value for the highest level or depth in order to compute the first/last element of the third scale factor. The default value is 0.
- 9) The **-sf=method_for_computing_the_third_scale_factor** argument allows to specify the method for computing the third scale factor, if a mesh-mask NetCDF file is created:
- **-sf=method1** : the third scale factor is computed as the differences between successive levels (or depths)
 - **-sf=method2** : the third scale factor is computed by assuming that each level or depth is located at the middle of the corresponding layer.
- 10) If the **-m=output_mesh_mask_netcdf_file** argument is present and if some scale factors can not be computed, these scale factors are set to 1.
- 11) The **-mi=missing_value** argument specifies the missing value indicator for the variance (VAR), standard-deviation (STD), skewness (SKEW) and kurtosis (KURT) variables in the *output_statistics_netcdf_file*.
- If the **-mi=** argument is not specified and the NetCDF variable has a *missing_value* or *_FillValue* attribute, the *missing_value* is set to 1 . e+20. This argument is not used if the NetCDF variable specified in the **-v=** argument has no *missing_value* or *_FillValue* attribute (excepted for the skewness and kurtosis statistics).
- 12) If **-nobias** is specified, unbiased estimates of skewness and kurtosis are computed.
- If the **-nobias** argument is absent, the biased standard estimates are computed.
- 13) If **-stderror** is specified, the standard errors of skewness and kurtosis are computed.
- If the **-stderror** argument is absent, the standard errors are not computed.
- 14) If **-prob** is specified, the significance levels of skewness and kurtosis are computed. Moreover, the **-prob** argument implies also the **-stderror** argument, even if this argument is not activated.
- If the **-prob** argument is absent, the significance levels are not computed.
- 15) The **-double** argument specifies that the VAR, STD, SKEW and KURT variables must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_climatology_netcdf_file*.
- 16) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) At least 4 observations by period, as determined from the **-p=periodicity** argument, are required, otherwise the program will stop.
- 20) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 21) For more details on the use of skewness and kurtosis coefficients in the climate literature see

- “Skewness, Kurtosis and Extreme Values of Northern Hemisphere Geopotential Heights”, by White, G., Monthly Weather Review, Vol. 108, 1446-1455, 1980. doi: [10.1175/1520-0493\(1980\)108<1446:SKAEVO>2.0.CO;2](https://doi.org/10.1175/1520-0493(1980)108<1446:SKAEVO>2.0.CO;2)
- “The Normality of El Nino”, by Burgers, G., and Stephenson, D.B , Geophysical Research Letters, 26, 1027-1030, 1999. doi: [10.1029/1999GL900161](https://doi.org/10.1029/1999GL900161)
- “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by Masson, S., et al., Climate Dynamics, Vol. 39, 681-707, 2012. doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](https://www.amazon.com/Statistical-Analysis-Climate-Research/dp/9780521012300)

Outputs

`comp_stat_4d` creates an output NetCDF file that contains the univariate statistics and number of observations of the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the **-p=periodicity** argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlon`, `nlat`, and `nlev` are the lengths of the geographical dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_mean(periodicity,nlev,nlat,nlon)` : the means for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_var(periodicity,nlev,nlat,nlon)` : the variances for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_std(periodicity,nlev,nlat,nlon)` : the standard-deviations for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 4) `netcdf_variable_skew(periodicity,nlev,nlat,nlon)` : the skewness for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 5) `netcdf_variable_kurt(periodicity,nlev,nlat,nlon)` : the kurtosis for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.

- 6) *netcdf_variable_min*(periodicity,nlev,nlat,nlon) : the minima for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 7) *netcdf_variable_max*(periodicity,nlev,nlat,nlon) : the maxima for each point in the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 8) *netcdf_variable_skew_prob*(periodicity,nlev,nlat,nlon) : the significance levels of the skewness coefficients.

This variable is stored only if the **-prob** argument has been specified when calling *comp_stat_4d*.

- 9) *netcdf_variable_kurt_prob*(periodicity,nlev,nlat,nlon) : the significance levels of the kurtosis coefficients.

This variable is stored only if the **-prob** argument has been specified when calling *comp_stat_4d*.

- 10) *netcdf_variable_nobs*(periodicity) : the number of observations used to compute the statistics.

- 11) *netcdf_variable_skew_se*(periodicity) : the standard-errors of the skewness coefficients.

This variable is stored only if the **-stderr** or **-prob** arguments have been specified when calling *comp_stat_4d*.

- 12) *netcdf_variable_kurt_se*(periodicity) : the standard-errors of the kurtosis coefficients.

This variable is stored only if the **-stderr** or **-prob** arguments have been specified when calling *comp_stat_4d*.

All the statistics and associated probabilities, excepted the number of observations and the standard-errors of the skewness and kurtosis coefficients are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values.

Optionally, *comp_stat_4d* can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) *netcdf_variable_nmask*(nlev,nlat,nlon) : a presence-absence or height-land-sea 3-D mask associated with the input NetCDF variable.
- 2) *netcdf_variable_e1n*(nlat,nlon) : the first scale factor associated with the 3-D grid-mesh of the input NetCDF variable.
- 3) *netcdf_variable_e2n*(nlat,nlon) : the second scale factor associated with the 3-D grid-mesh of the input NetCDF variable.
- 4) *netcdf_variable_e3n*(nlev,1,1) : the third scale factor associated with the 3-D grid-mesh of the input NetCDF variable.

Multiplying the two first scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable. Multiplying the three scale factors together gives the volume (or a quantity proportional to the weight if the unit of the vertical coordinate variable is in hPa) of each parcel in the 3-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly statistics from the NetCDF file *ST7_1m_00101_20012_grid_T_votemper.nc*, which includes a NetCDF variable *votemper*, and store the results in a NetCDF file named *stat_ST7_1m_00101_20012_grid_T_votemper.nc*, use the following command :

```
$ comp_stat_4d \
-f=ST7_1m_0101_20012_grid_T_votemper.nc \
-v=votemper \
-p=12 \
-o=stat_ST7_1m_00101_20012_grid_T_votemper.nc
```

- 2) For computing monthly unbiased univariate statistics from the NetCDF file `vwnd.mon.mean.nc`, which includes a NetCDF variable `vwnd`, and store the results in a NetCDF file named `stat_vwnd.nc` and, in addition, generate an associated mesh_mask_NetCDF_file named `mesh_mask_wind_ncep2.nc`, use the following command :

```
$ comp_stat_4d \
-f=vwnd.mon.mean.nc \
-v=vwnd \
-p=12 \
-nobias \
-m=mesh_mask_wind_ncep2.nc
```

2.66 comp_stat_miss_3d

2.66.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.66.2 Latest revision

29/04/2026

2.66.3 Purpose

Compute univariate statistics from a tridimensional variable with missing values extracted from a NetCDF dataset and, optionally, the associated mesh-mask and scale factors of the 2-D grid-mesh associated with the input NetCDF variable.

It is assumed that the input NetCDF variable has a scalar *missing_value* or *_FillValue* attribute, which allows a proper identification of the missing values in the dataset, otherwise the procedure will exit with an error message.

Mean, variance, standard-deviation, skewness, kurtosis, minimum and maximum are computed for each point in the multi-channel time series of the 2-D grid-mesh associated with the input NetCDF variable. These statistics are computed by using all available observations for each point time series. Since missing values are present, the number of observations used to compute the statistics may vary from one point to another in the 2-D grid-mesh associated with the NetCDF variable.

These univariate statistics may be computed by taking into account the periodicity of the data. These statistics are stored in an output NetCDF dataset.

Refer to [comp_stat_3d](#), for a precise definition of the statistics and how these univariate statistics are calculated in `comp_stat_miss_3d`.

This procedure is parallelized if OpenMP is used and the NCSTAT software has been built with the `_PARALLEL_READ` CPP macro (however, in this version of `comp_stat_miss_3d`, parallelism is on the number of periods as determined by the `-p=periodicity` argument. This means that the number of processors or threads must not be greater than the *periodicity* parameter).

Moreover, this procedure computes all the univariate statistics with only one pass through the data and an out-of-core strategy which is highly efficient on huge datasets.

Optionally, a mesh-mask NetCDF dataset may also be created. This dataset will contain a presence-absence 2-D mask and scale factor variables which may be used to compute the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable. This mesh-mask NetCDF dataset will be used by other NCSTAT procedures such as [comp_serie_miss_3d](#), [comp_eof_miss_3d](#), etc. When computing the scale factors, it is assumed that the 2-D grid-mesh associated with the input NetCDF variable is regular or that this grid is a regular gaussian grid (the grid of the input NetCDF variable is assumed to be gaussian if this NetCDF variable has a *grid_type* character attribute with the string “gaussian” as a value).

If your data does not contain missing values, use [comp_stat_3d](#) instead of `comp_stat_miss_3d` to estimate univariate statistics from your dataset.

2.66.4 Further Details

Usage

```
$ comp_stat_miss_3d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-p=periodicity                (optional) \  
-x=lon1,lon2                  (optional) \  
-y=lat1,lat2                  (optional) \  
-t=time1,time2                (optional) \  
-o=output_statistics_netcdf_file (optional) \  
-m=output_mesh_mask_netcdf_file (optional) \  
-np=nobs_limit_by_period      (optional) \  
-yl=latl1,latl2               (optional) \  
-mi=missing_value             (optional) \  
-nobias                       (optional) \  
-stderror                     (optional) \  
-prob                          (optional) \  
-double                        (optional) \  
-bigfile                       (optional) \  
-hdf5                          (optional) \  
-cdf5                          (optional) \  
-tlimited                       (optional)
```

By default

- p=** by default, the periodicity is equal to 1
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_statistics_netcdf_file* is named `stat_netcdf_variable.nc`
- m=** the *output_mesh_mask_netcdf_file* is not created
- np=** this argument is equal to 0
- yl=** it is assumed that the grid is gaussian and the domain is the whole globe or that the grid is regular when computing the scale factors

- mi=** by default, the *missing_value* for the statistic variables is equal to 1.e+20
- mi=** the *missing_value* for the statistics variables in the output NetCDF file is set to 1.e+20
- nobias** biased estimates of kurtosis and skewness coefficients are computed. However, if **-nobias** is activated, unbiased estimates of kurtosis and skewness are computed
- stderror** the standard errors of the kurtosis and skewness coefficients are not computed. However, if **-stderror** is activated, these standard errors are computed
- prob** the significance levels of the kurtosis and skewness coefficients are not computed. However, if **-prob** is activated, these significance levels are computed
- double** the statistic variables are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the statistic variables are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which statistics must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified, with yearly data **-p=1** may be used, etc. By default, the *periodicity* is set to 1. Note that the output NetCDF file will have *periodicity* time observations.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, statistics are computed for all the points of the 2-D grid-mesh associated with the *netcdf_variable*.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values for *lon1* are not allowed.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using `comp_stat_miss_3d`.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to compute the statistics.
The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1.
- 5) It is assumed that the specified *netcdf_variable* has a scalar *missing_value* or *_FillValue* attribute and that missing values in the data are identified by the value of this *missing_value* attribute.
- 6) If the **-m=output_mesh_mask_netcdf_file** argument is present and the **-yl=** argument is missing, it is assumed when computing the scale factors that the 2-D grid-mesh associated with the input NetCDF variable is gaussian and covers the whole globe or that this 2-D grid-mesh is regular.

If the grid is a regular gaussian grid, but the domain does not cover the whole globe, the **-yl=** argument must be used to specify the latitude boundaries of the domain, otherwise the first and last columns (elements) of the first two scale factors are wrong.

The **-yl=** argument specifies the latitude limits of the domain in degrees (*latl1* and *latl2* must be real numbers) and the ordering of these two values must be coherent with the ordering of the latitudinal coordinate variable in the *input_netcdf_file* file.

- 7) If the **-np=nobs_limit_by_period** and **-m=output_mesh_mask_netcdf_file** arguments are present, the mask in the *output_mesh_mask_netcdf_file* is constructed as follow:

- If the number of observations by period (as determined by the **-p=** argument) is less than *nobs_limit_by_period*, the corresponding mask value is set to 0 (e.g., missing), otherwise the mask value is set to 1.

If the **-np=nobs_limit_by_period** argument is not specified and the **-m=output_mesh_mask_netcdf_file** argument is present, the mask is constructed as follow:

- If the total number of non-missing observations is 0, the corresponding mask value is set to 0 (e.g., missing), otherwise the mask value is set to 1.

- 8) If the **-m=output_mesh_mask_netcdf_file** argument is present and if some scale factors can not be computed, these scale factors are set to 1.
- 9) The **-mi=missing_value** argument specifies the missing value indicator for the variance (VAR), standard-deviation (STD), skewness (SKEW) and kurtosis (KURT) variables in the *output_statistics_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* and *_FillValue* attributes are set to 1.e+20.

- 10) If **-nobias** is specified, unbiased estimates of skewness and kurtosis are computed.

If the **-nobias** argument is absent, the biased standard estimates are computed.

- 11) If **-stderror** is specified, the standard errors of skewness and kurtosis are computed.

If the **-stderror** argument is absent, the standard errors are not computed.

- 12) If **-prob** is specified, the significance levels of skewness and kurtosis are computed. Moreover, the **-prob** argument implies also the **-stderror** argument, even if this argument is not activated.

If the **-prob** argument is absent, the significance levels are not computed.

- 13) The **-double** argument specifies that the VAR, STD, SKEW and KURT variables must be stored as double-precision floating point numbers instead of single-precision floating point numbers in the *output_statistics_netcdf_file*.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) At least 4 observations by period, as determined from the `-p=periodicity` argument, are required, otherwise the program will stop.
- 18) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 19) For more details on the use of skewness and kurtosis coefficients in the climate literature see
 - “Skewness, Kurtosis and Extreme Values of Northern Hemisphere Geopotential Heights”, by White, G., Monthly Weather Review, Vol. 108, 1446-1455, 1980. doi: [10.1175/1520-0493\(1980\)108<1446:SKAEVO>2.0.CO;2](https://doi.org/10.1175/1520-0493(1980)108<1446:SKAEVO>2.0.CO;2)
 - “The Normality of El Nino”, by Burgers, G., and Stephenson, D.B , Geophysical Research Letters, 26, 1027-1030, 1999. doi: [10.1029/1999GL900161](https://doi.org/10.1029/1999GL900161)
 - “Impact of intra-daily SST variability on ENSO characteristics in a coupled model” by Masson, S., et al., Climate Dynamics, Vol. 39, 681-707, 2012. doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
 - “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 2, 484 pp., 2002. ISBN: [9780521012300](https://www.amazon.com/Statistical-Analysis-Climate-Research/dp/0521012300)

Outputs

`comp_stat_miss_3d` creates an output NetCDF file that contains the univariate statistics and number of observations for the input NetCDF variable, taking into account eventually the periodicity of the data as determined by the `-p=periodicity` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable) :

- 1) `netcdf_variable_mean(periodicity,nlat,nlon)` : the means for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_var(periodicity,nlat,nlon)` : the variances for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_std(periodicity,nlat,nlon)` : the standard-deviations for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 4) `netcdf_variable_skew(periodicity,nlat,nlon)` : the skewness for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 5) `netcdf_variable_kurt(periodicity,nlat,nlon)` : the kurtosis for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 6) `netcdf_variable_min(periodicity,nlat,nlon)` : the minima for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 7) `netcdf_variable_max(periodicity,nlat,nlon)` : the maxima for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 8) `netcdf_variable_nobs(periodicity,nlat,nlon)` : the number of observations used to compute the statistics for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

- 9) *netcdf_variable_skew_se*(periodicity,nlat,nlon) : the standard-errors of the skewness coefficients for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-stderror** or **-prob** arguments have been specified when calling *comp_stat_miss_3d*.

- 10) *netcdf_variable_kurt_se*(periodicity,nlat,nlon) : the standard-errors of the kurtosis coefficients for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-stderror** or **-prob** arguments have been specified when calling *comp_stat_miss_3d*.

- 11) *netcdf_variable_skew_prob*(periodicity,nlat,nlon) : the significance levels of the skewness coefficients for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-prob** argument has been specified when calling *comp_stat_miss_3d*.

- 12) *netcdf_variable_kurt_prob*(periodicity,nlat,nlon) : the significance levels of the kurtosis coefficients for each point in the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-prob** argument has been specified when calling *comp_stat_miss_3d*.

All the statistics and associated probabilities, excepted the number of observations and the standard-errors of the skewness and kurtosis coefficients are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, these output NetCDF variables are filled with missing values.

Optionally, *comp_stat_miss_3d* can also create an output mesh-mask NetCDF file that contains the following NetCDF variables, in addition of coordinate variables if they are present in the input NetCDF file :

- 1) *netcdf_variable_nmask*(nlat,nlon) : a presence-absence or land-sea 2-D mask associated with the input NetCDF variable.
- 2) *netcdf_variable_e1n*(nlat,nlon) : the first scale factor associated with the 2-D grid-mesh of the input NetCDF variable.
- 3) *netcdf_variable_e2n*(nlat,nlon) : the second scale factor associated with the 2-D grid-mesh of the input NetCDF variable.

Multiplying the two scale factors together gives the surface of each cell in the 2-D grid-mesh associated with the input NetCDF variable.

Examples

- 1) For computing monthly univariate statistics from the NetCDF file *precip.mon.mean.nc*, which includes a NetCDF variable *precip* with missing values, and store the results in a NetCDF file named *stat_cmap_1m_precip.nc*, use the following command :

```
$ comp_stat_miss_3d \
-f=precip.mon.mean.nc \
-v=precip \
```

(continues on next page)

(continued from previous page)

```
-p=12 \
-o=stat_cmap_1m_precip.nc
```

- 2) For computing monthly unbiased univariate statistics from the NetCDF file `precip.mon.mean.nc`, which includes a NetCDF variable `precip` with missing values, and store the results in a NetCDF file named `stat_cmap_1m_precip.nc` and, in addition, generate an associated *mesh_mask_netcdf_file* named `mask_cmap_precip.nc`, use the following command :

```
$ comp_stat_miss_3d \
-f=precip.mon.mean.nc \
-v=precip \
-p=12 \
-o=stat_cmap_1m_precip.nc \
-nobias \
-m=mask_cmap_precip.nc
```

2.67 comp_stl_1d

2.67.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.67.2 Latest revision

30/04/2026

2.67.3 Purpose

Decompose a time series extracted from a unidimensional variable in a NetCDF dataset into seasonal and trend components by the Seasonal-Trend decomposition procedure based on Loess (e.g., STL). The STL procedure is a powerful statistical technique for describing a discrete time series [Cleveland_etal] [Terrayc] . In the STL procedure, the analyzed time series is decomposed into three terms:

$$\mathbf{X}(t) = \mathbf{T}(t) + \mathbf{A}(t) + \mathbf{R}(t)$$

where t refers to a time index, the **T** term is used to quantify the trend and low-frequency variations in the time series, the **A** term describes the harmonic component (e.g., diurnal or seasonal cycle) and its modulation through time and, finally, the **R** term contains the residual component.

All the terms are estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., Loess) to data windows whose length is chosen by the user. The STL procedure is an iterative process, which may be interpreted as a frequency filter directly applicable to a non-stationary unidimensional time series including harmonic components [Cleveland_etal] . Other important features of STL are the specification of the amounts of seasonal and trend smoothing according to the choice of the user, the ability to produce robust estimates of the trend and harmonic components that are not distorted by aberrant behavior in the data and the stationarity of the **R** time series.

This procedure returns the seasonal and trend components as estimated by the STL procedure and, optionally, the residuals or the sum of the residual and trend components in a NetCDF dataset, for the time series associated with a NetCDF variable.

If the NetCDF variable is tridimensional or fourdimensional use *comp_stl_3d* or *comp_stl_4d*, respectively instead of *comp_stl_1d*. If the time series have no seasonal (or diurnal) cycle, use *comp_trend_1d* instead of *comp_stl_1d* in order to estimate the trend component by the Loess procedure.

The exact meaning and default values for most of the optional parameters of *comp_stl_1d* are exactly the same as in the original procedure described by [Cleveland_etal] and the user is referred to this publication for further details on the STL procedure.

2.67.4 Further Details

Usage

```
$ comp_stl_1d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -p=periodicity \
  -ns=seasonal_smoother_length      (ns) \
  -t=time1,time2                    (optional) \
  -a=type_of_analysis               (optional : stl, residual, \
                                     residual_trend) \
  -o=output_netcdf_file             (optional) \
  -nt=trend_smoother_length         (nt)      (optional) \
  -nl=low_pass_smoother_length      (nl)      (optional) \
  -isdeg=seasonal_smoother_degree   (isdeg)   (optional : 0, 1 ) \
  -itdeg=trend_smoother_degree      (itdeg)   (optional : 0, 1, 2 ) \
  -ildeg=low_pass_smoother_degree   (ildeg)   (optional : 0, 1, 2 ) \
  -nsjump=seasonal_skipping_value   (nsjump)  (optional) \
  -ntjump=trend_skipping_value      (ntjump)  (optional) \
  -nljump=low_pass_skipping_value   (nljump)  (optional) \
  -maxit=max_robustness_iterations  (maxit)   (optional) \
  -ni=number_of_loops               (ni)       (optional) \
  -sms=seasonal_smoothing_factor     (optional) \
  -smt=trend_smoothing_factor        (optional) \
  -robust                            (optional) \
  -double                            (optional) \
  -hdf5                              (optional) \
  -cdf5                              (optional) \
  -tlimited                           (optional)
```

By default

- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *stl*. This means that the residual times series are not computed and not stored in the *output_netcdf_file*
- o=** the *output_netcdf_file* is named *stl_netcdf_variable.nc*
- nt=** the *trend_smoother_length* is set to the smallest odd integer greater than or equal to $(1.5 * \text{periodicity}) / (1 - (1.5 / \text{ns}))$
- nl=** the *low_pass_smoother_length* is set to the smallest odd integer greater than or equal to *periodicity*
- isdeg=** the *seasonal_smoother_degree* is set to 1
- itdeg=** the *trend_smoother_degree* is set to 1
- ildeg=** the *low_pass_smoother_degree* is set to the *trend_smoother_degree*

-nsjump= the *seasonal_skipping_value* is set to `ns/10`

-ntjump= the *trend_skipping_value* is set to `nt/10`

-nljump= the *low_pass_skipping_value* is set to `n1/10`

-maxit= the *max_robustness_iterations* is set to 15

-ni= the *number_of_loops* is set to 1 if **-robust** is specified and 2 otherwise

-sms= no smoothing is applied to the seasonal component

-smt= no smoothing is applied to the trend component

-robust robustness iterations are not used, by default

-double the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

-hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

-cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the time series must be decomposed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.
- 3) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified. The *periodicity* must be greater than 2. If your time series do not have a periodic component, e.g., if *periodicity* is equal to 1, use *comp_trend_1d* instead of *comp_stl_1d*.
- 4) The **-ns=** argument specifies the length of the seasonal smoother, *ns*. The value of *ns* should be an odd integer greater than or equal to 3; a value greater than 6 is recommended. As the value of the **-ns=** argument increases the values of the seasonal component at a given point in the seasonal cycle (e.g., January values of a monthly series with a yearly cycle) become smoother. However, the value of *ns* has no direct effect on the smoothness of successive values of the seasonal component.
- 5) The **-nt=** argument specifies the length of the trend smoother, *nt*. The value of *nt* should be an odd integer greater than or equal to 3; a value of *nt* between $1.5 * \textit{periodicity}$ and $2 * \textit{periodicity}$ is recommended. As *nt* increases the values of the trend component become smoother.
- 6) The **-nl=** argument specifies the length of the low-pass smoother, *n1*. The value of *n1* should be an odd integer greater than or equal to 3; the smallest odd integer greater than or equal to *periodicity* is recommended.
- 7) The **-isdeg=** argument specifies the degree of locally-fitted polynomial in seasonal smoothing. The value is 0 or 1.
- 8) The **-itdeg=** argument specifies the degree of locally-fitted polynomial in trend smoothing. The value is 0, 1 or 2.

- 9) The **-ildeg=** argument specifies the degree of locally-fitted polynomial in low-pass smoothing. The value is 0, 1 or 2.
- 10) The **-nsjump=** argument specifies the skipping value for seasonal smoothing. The seasonal smoother skips ahead `nsjump` points and then linearly interpolates in between. The value of `nsjump` should be a positive integer; if `nsjump` is set to 1, a seasonal smooth is calculated at all points in the time series. To make the procedure run faster, a reasonable choice for `nsjump` is 10% or 20% of `ns`. The default value is `ns/10`.
- 11) The **-ntjump=** argument specifies the skipping value for trend smoothing. The default value is `nt/10`.
- 12) The **-nljump=** argument specifies the skipping value for the low-pass filter. The default value is `nl/10`.
- 13) The **-ni=** argument specifies the number of loops for updating the seasonal and trend components. The value of `ni` should be a positive integer. If the data are well behaved without outliers, then robustness iterations are not needed. In this case do not use the **-robust** argument, and set `ni` between 2 and 5 depending on how much security you want that the seasonal-trend looping converges. If outliers are present then use the **-robust** argument and set `ni` to 1 or 2.
- 14) The **-a=** argument specifies if the residuals from the trend and seasonal components are stored in the output NetCDF file. If:

- **-a=stl** means that the residuals are not computed
- **-a=residual** means that the residuals are computed and stored
- **-a=residual_trend** means that the residual and trend components are summed and the result is stored.

The default is **-a=stl**, e.g., the residuals are not stored. Note that in all cases, the seasonal and trend components are computed and stored in the output NetCDF file.

- 15) If **-robust** is specified, robustness iterations are carried out until convergence of both seasonal and trend components or with a maximum of `max_robustness_iterations` iterations as specified by the **-maxit=** argument. Convergence occurs if the maximum changes in individual seasonal and trend fits are less than 1% of the component's range after the previous iteration.
- 16) **-sms=seasonal_smoothing_factor** means that the seasonal component extracted from the `netcdf_variable` (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * \text{seasonal_smoothing_factor} + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual` or `residual_trend`). `seasonal_smoothing_factor` must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original STL procedure.
- 17) **-smt=trend_smoothing_factor** means that the trend component extracted from the `netcdf_variable` (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * \text{trend_smoothing_factor} + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual` or `residual_trend`). `trend_smoothing_factor` must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original STL procedure.
- 18) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 19) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the `output_netcdf_file` will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 20) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 21) It is assumed that the data has no missing values.
- 22) If the time series have no seasonal cycle, use *comp_trend_1d* instead of *comp_stl_1d*.
- 23) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 24) For more details on the STL procedure and examples of use in the climate literature, see
 - “A Seasonal-Trend Decomposition Procedure Based on Loess”, by R.B. Cleveland, W.S. Cleveland, J.E. McRae and I. Terpenning, 1990, Journal of Official Statistics, 6, 3-73. <http://www.jos.nu/Articles/abstract.asp?article=613>
 - “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by P. Terray, 2011, Climate Dynamics, Vol. 36:2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

comp_stl_1d creates an output NetCDF file that contains the trend and harmonic components extracted from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, *ntime* is the selected number of time observations) :

- 1) *netcdf_variable_trend(ntime)* : the trend component for the time series associated with the input NetCDF variable.
- 2) *netcdf_variable_seasonal(ntime)* : the harmonic component for the time series associated with the input NetCDF variable.
- 3) *netcdf_variable_residual(ntime)* : the residual component for the time series associated with the input NetCDF variable.

This variable is stored only if the `-a=residual` argument has been specified when calling *comp_stl_1d*.

- 4) *netcdf_variable_residual_trend(ntime)* : the sum of the residual and trend components for the time series associated with the input NetCDF variable.

This variable is stored only if the `-a=residual_trend` argument has been specified when calling *comp_stl_1d*.

Examples

- 1) For computing a STL decomposition from the unidimensional NetCDF variable called *ts* extracted from the file *ts_cnrm_cm3.picntrl_monthly_glob.nc*, which includes a global monthly time series, and store the results in the NetCDF file *ts_cnrm_cm3.picntrl_monthly_glob_stl.nc*, use the following command :

```
$ comp_stl_1d \
-f=ts_cnrm_cm3.picntrl_monthly_glob.nc \
-v=ts \
-p=12 \
-ns=35 \
```

(continues on next page)

(continued from previous page)

```
-nt=240 \  
-a=residual \  
-robust \  
-o=ts_cnrm_cm3.picntrl_monthly_glob_stl.nc
```

2.68 comp_stl_3d

2.68.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.68.2 Latest revision

30/04/2026

2.68.3 Purpose

Decompose a multi-channel time series extracted from a tridimensional variable in a NetCDF data set into seasonal and trend multi-channel components by the Seasonal-Trend decomposition procedure based on Loess (e.g., STL) [Cleveland_etal]. The STL procedure is a powerful statistical technique for describing a discrete time series [Cleveland_etal] [Terrayc]. In the STL procedure, the analyzed (multi-channel) time series is decomposed into three terms:

$$\mathbf{X}(t) = \mathbf{T}(t) + \mathbf{A}(t) + \mathbf{R}(t)$$

where t refers to a time index, the \mathbf{T} term is used to quantify the trend and low-frequency variations in the time series, the \mathbf{A} term describes the harmonic component (e.g., diurnal or seasonal cycle) and its modulation through time and, finally, the \mathbf{R} term contains the residual component.

All the terms are estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., Loess) to data windows whose length is chosen by the user. The STL procedure is an iterative process, which may be interpreted as a frequency filter directly applicable to non-stationary (unidimensional) time series including harmonic components [Cleveland_etal]. Other important features of STL are the specification of the amounts of seasonal and trend smoothing according to the choice of the user, the ability to produce robust estimates of the trend and harmonic components that are not distorted by aberrant behavior in the data and the stationarity of the \mathbf{R} time series.

This procedure returns the seasonal and trend multi-channel components as estimated by the STL procedure and, optionally, the residuals or the sum of the residual and trend components in a NetCDF dataset, for the multi-channel time series associated with a NetCDF variable.

If the NetCDF variable is unidimensional or fourdimensional use `comp_stl_1d` or `comp_stl_4d`, respectively instead of `comp_stl_3d`. If the multi-channel time series have no seasonal (or diurnal) cycle, use `comp_trend_3d` instead of `comp_stl_3d` in order to estimate the trend component by the Loess procedure.

The exact meaning and default values for most of the optional parameters of `comp_stl_3d` are exactly the same as in the original procedure described by [Cleveland_etal] and the user is referred to this publication for further details on the STL procedure.

This procedure is parallelized if OpenMP is used.

2.68.4 Further Details

Usage

```
$ comp_stl_3d \
-f=input_netcdf_file \
-v=input_netcdf_variable \
-p=periodicity \
-ns=seasonal_smoother_length      (ns) \
-m=input_mesh_mask_netcdf_file    (optional) \
-g=grid_type                      (optional : n, t, u, v, w, f) \
-x=lon1,lon2                     (optional) \
-y=lat1,lat2                     (optional) \
-t=time1,time2                   (optional) \
-a=type_of_analysis              (optional : stl, residual, \
                                residual_trend) \
-o=output_netcdf_file            (optional) \
-nt=trend_smoother_length        (nt)   (optional) \
-nl=low_pass_smoother_length     (nl)   (optional) \
-isdeg=seasonal_smoother_degree  (isdeg) (optional : 0, 1 ) \
-itdeg=trend_smoother_degree     (itdeg) (optional : 0, 1, 2 ) \
-ildeg=low_pass_smoother_degree  (ildeg) (optional : 0, 1, 2 ) \
-nsjump=seasonal_skipping_value  (nsjump) (optional) \
-ntjump=trend_skipping_value     (ntjump) (optional) \
-nljump=low_pass_skipping_value  (nljump) (optional) \
-maxit=max_robustness_iterations (maxit) (optional) \
-ni=number_of_loops              (ni)   (optional) \
-sms=seasonal_smoothing_factor   (optional) \
-smt=trend_smoothing_factor      (optional) \
-robust                          (optional) \
-mi=missing_value               (optional) \
-double                         (optional) \
-bigfile                        (optional) \
-hdf5                           (optional) \
-cdf5                           (optional) \
-tlimited                        (optional)
```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *stl*. This means that the residual times series are not computed and not stored in the *output_netcdf_file*
- o=** the *output_netcdf_file* is named *stl_netcdf_variable.nc*
- nt=** the *trend_smoother_length* is set to the smallest odd integer greater than or equal to $(1.5 * \text{periodicity}) / (1 - (1.5 / \text{ns}))$
- nl=** the *low_pass_smoother_length* is set to the smallest odd integer greater than or equal to *periodicity*

-isdeg= the *seasonal_smoother_degree* is set to 1

-itdeg= the *trend_smoother_degree* is set to 1

-ildeg= the *low_pass_smoother_degree* is set to the *trend_smoother_degree*

-nsjump= the *seasonal_skipping_value* is set to $ns/10$

-ntjump= the *trend_skipping_value* is set to $nt/10$

-nljump= the *low_pass_skipping_value* is set to $nl/10$

-maxit= the *max_robustness_iterations* is set to 15

-ni= the *number_of_loops* is set to 1 if **-robust** is specified and 2 otherwise

-sms= no smoothing is applied to the seasonal component

-smt= no smoothing is applied to the trend component

-robust robustness iterations are not used, by default

-mi= the *missing_value* for the output variables is equal to $1.e+20$

-double the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

-bigfile a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file

-hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file

-cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file

-tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the multi-channel time series must be decomposed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.

- 2) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_stl_3d*.

- 3) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 4) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).
If **-g=** is set to `n`, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.
This argument is also used to determine the name of the NetCDF `mesh_mask` variable if an `input_mesh_mask_netcdf_file` is used as specified with the **-m=** argument
- 6) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified. The *periodicity* must be greater than 2. If your time series do not have a periodic component, e.g., if *periodicity* is equal to 1, use `comp_trend_3d` instead of `comp_stl_3d`.
- 7) The **-ns=** argument specifies the length of the seasonal smoother, `ns`. The value of `ns` should be an odd integer greater than or equal to 3; a value greater than 6 is recommended. As the value of the **-ns=** argument increases the values of the seasonal component at a given point in the seasonal cycle (e.g., January values of a monthly series with a yearly cycle) become smoother. However, the value of `ns` has no direct effect on the smoothness of successive values of the seasonal component.
- 8) The **-nt=** argument specifies the length of the trend smoother, `nt`. The value of `nt` should be an odd integer greater than or equal to 3; a value of `nt` between $1.5 * \text{periodicity}$ and $2 * \text{periodicity}$ is recommended. As `nt` increases the values of the trend component become smoother.
- 9) The **-nl=** argument specifies the length of the low-pass smoother, `nl`. The value of `nl` should be an odd integer greater than or equal to 3; the smallest odd integer greater than or equal to *periodicity* is recommended.
- 10) The **-isdeg=** argument specifies the degree of locally-fitted polynomial in seasonal smoothing. The value is 0 or 1.
- 11) The **-itdeg=** argument specifies the degree of locally-fitted polynomial in trend smoothing. The value is 0, 1 or 2.
- 12) The **-ildeg=** argument specifies the degree of locally-fitted polynomial in low-pass smoothing. The value is 0, 1 or 2.
- 13) The **-nsjump=** argument specifies the skipping value for seasonal smoothing. The seasonal smoother skips ahead `nsjump` points and then linearly interpolates in between. The value of `nsjump` should be a positive integer; if `nsjump` is set to 1, a seasonal smooth is calculated at all points in the time series. To make the procedure run faster, a reasonable choice for `nsjump` is 10% or 20% of `ns`. The default value is `ns/10`.
- 14) The **-ntjump=** argument specifies the skipping value for trend smoothing. The default value is `nt/10`.
- 15) The **-nljump=** argument specifies the skipping value for the low-pass filter. The default value is `nl/10`.
- 16) The **-ni=** argument specifies the number of loops for updating the seasonal and trend components. The value of `ni` should be a positive integer. If the data are well behaved without outliers, then robustness iterations are not needed. In this case do not use the **-robust** argument, and set `ni` between 2 and 5 depending on how much security you want that the seasonal-trend looping converges. If outliers are present then use the **-robust** argument and set `ni` to 1 or 2.
- 17) The **-a=** argument specifies if the residuals from the trend and seasonal components are stored in the output NetCDF file. If:
 - **-a=stl** means that the residuals are not computed
 - **-a=residual** means that the residuals are computed and stored
 - **-a=residual_trend** means that the residual and trend components are summed and the result is stored.
 The default is **-a=stl**, e.g., the residuals are not stored. Note that in all cases, the seasonal and trend components are computed and stored in the output NetCDF file.
- 18) If **-robust** is specified, robustness iterations are carried out until convergence of both seasonal and trend components or with a maximum of `max_robustness_iterations` iterations as specified by the **-maxit=** argument.

Convergence occurs if the maximum changes in individual seasonal and trend fits are less than 1% of the component's range after the previous iteration.

- 19) **-sms=seasonal_smoothing_factor** means that the seasonal component extracted from the `netcdf_variable` (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * seasonal_smoothing_factor + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual` or `residual_trend`). *seasonal_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original STL procedure.
- 20) **-smt=trend_smoothing_factor** means that the trend component extracted from the `netcdf_variable` (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * trend_smoothing_factor + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual` or `residual_trend`). *trend_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original STL procedure.
- 21) The **-mi=missing_value** argument specifies the missing value indicator for the output variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to `1.e+20`.
- 22) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.
- 23) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 24) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.
- 25) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.
- 26) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 27) If the time series have no seasonal cycle, use `comp_trend_3d` instead of `comp_stl_3d`.
- 28) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 29) For more details on the STL procedure and examples of use in the climate literature, see
 - “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>

- “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by Terray, P., Climate Dynamics, Vol. 36, 2171-2199, 2011. doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

`comp_stl_3d` creates an output NetCDF file that contains the trend and harmonic components extracted from the time series associated with the input NetCDF variable. The output NetCDF data set contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable and `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_trend(ntime, nlat, nlon)` : the trend component for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_seasonal(ntime, nlat, nlon)` : the harmonic component for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 3) `netcdf_variable_residual(ntime, nlat, nlon)` : the residual component for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the `-a=residual` argument has been specified when calling `comp_stl_3d`.

- 4) `netcdf_variable_residual_trend(ntime, nlat, nlon)` : the sum of the residual and trend components for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the `-a=residual_trend` argument has been specified when calling `comp_stl_3d`.

The trend, harmonic and residual multi-channel components are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For computing a STL decomposition from the tridimensional NetCDF variable called `sosstsst` extracted from the file `ST7_1m_00101_20012_sosstsst_grid_T.nc`, which includes monthly time series, and store the results in the NetCDF file `stl_ST7_1m_00101_20012_sosstsst_grid_T.nc`, use the following command :

```
$ comp_stl_3d \
-f=ST7_1m_00101_20012_sosstsst_grid_T.nc \
-v=sosstsst \
-p=12 \
-ns=35 \
-nt=127 \
-a=residual \
-robust \
-o=stl_ST7_1m_00101_20012_sosstsst_grid_T.nc
```

2.69 comp_stl_4d

2.69.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.69.2 Latest revision

30/04/2026

2.69.3 Purpose

Decompose a multi-channel time series extracted from a fourdimensional variable in a NetCDF dataset into seasonal and trend multi-channel components by the Seasonal-Trend decomposition procedure based on Loess (e.g., STL). The STL procedure is a powerful statistical technique for describing a discrete time series [Cleveland_etal] [Terrayc]. In the STL procedure, the analyzed (multi-channel) time series is decomposed into three terms:

$$\mathbf{X}(t) = \mathbf{T}(t) + \mathbf{A}(t) + \mathbf{R}(t)$$

where t refers to a time index, the \mathbf{T} term is used to quantify the trend and low-frequency variations in the time series, the \mathbf{A} term describes the harmonic component (e.g., diurnal or seasonal cycle) and its modulation through time and, finally, the \mathbf{R} term contains the residual component.

All the terms are estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., Loess) to data windows whose length is chosen by the user. The STL procedure is an iterative process, which may be interpreted as a frequency filter directly applicable to non-stationary (unidimensional) time series including harmonic components [Cleveland_etal]. Other important features of STL are the specification of the amounts of seasonal and trend smoothing according to the choice of the user, the ability to produce robust estimates of the trend and harmonic components that are not distorted by aberrant behavior in the data and the stationarity of the \mathbf{R} time series.

This procedure returns the seasonal and trend components as estimated by the STL procedure and, optionally, the residuals or the sum of the residual and trend components in a NetCDF dataset, for the multi-channel time series associated with a NetCDF variable.

If the NetCDF variable is unidimensional or tridimensional use `comp_stl_1d` or `comp_stl_3d`, respectively instead of `comp_stl_4d`. If the multi-channel time series have no seasonal (or diurnal) cycle, use `comp_trend_4d` instead of `comp_stl_4d` in order to estimate the trend component by the Loess procedure.

The exact meaning and default values for most of the optional parameters of `comp_stl_4d` are exactly the same as in the original procedure described by [Cleveland_etal] and the user is referred to this publication for further details on the STL procedure.

This procedure is parallelized if OpenMP is used.

2.69.4 Further Details

Usage

```
$ comp_stl_4d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -p=periodicity \
  -ns=seasonal_smoother_length      (ns) \
  -m=input_mesh_mask_netcdf_file    (optional) \
  -g=grid_type                      (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                      (optional) \
  -y=lat1,lat2                      (optional) \
  -z=level1,level2                  (optional) \
  -t=time1,time2                    (optional) \
  -a=type_of_analysis               (optional : stl, residual, \
```

(continues on next page)

(continued from previous page)

```

                                residual_trend) \
-o=output_netcdf_file           (optional) \
-nt=trend_smoother_length      (nt)       (optional) \
-nl=low_pass_smoother_length   (nl)       (optional) \
-isdeg=seasonal_smoother_degree (isdeg)   (optional : 0, 1 ) \
-itdeg=trend_smoother_degree   (itdeg)   (optional : 0, 1, 2 ) \
-ildeg=low_pass_smoother_degree (ildeg) (optional : 0, 1, 2 ) \
-nsjump=seasonal_skipping_value (nsjump) (optional) \
-ntjump=trend_skipping_value    (ntjump)  (optional) \
-nljump=low_pass_skipping_value (nljump) (optional) \
-maxit=max_robustness_iterations (maxit) (optional) \
-ni=number_of_loops            (ni)       (optional) \
-sms=seasonal_smoothing_factor  (optional) \
-smt=trend_smoothing_factor     (optional) \
-robust                         (optional) \
-mi=missing_value              (optional) \
-double                        (optional) \
-bigfile                       (optional) \
-hdf5                          (optional) \
-cdf5                          (optional) \
-tlimited                       (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to `n` which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to `stl`. This means that the residual times series are not computed and not stored in the *output_netcdf_file*
- o=** the *output_netcdf_file* is named `stl_netcdf_variable.nc`
- nt=** the *trend_smoother_length* is set to the smallest odd integer greater than or equal to $(1.5 * \text{periodicity}) / (1 - (1.5 / \text{ns}))$
- nl=** the *low_pass_smoother_length* is set to the smallest odd integer greater than or equal to *periodicity*
- isdeg=** the *seasonal_smoother_degree* is set to 1
- itdeg=** the *trend_smoother_degree* is set to 1
- ildeg=** the *low_pass_smoother_degree* is set to the *trend_smoother_degree*
- nsjump=** the *seasonal_skipping_value* is set to `ns/10`
- ntjump=** the *trend_skipping_value* is set to `nt/10`
- nljump=** the *low_pass_skipping_value* is set to `nl/10`
- maxit=** the *max_robustness_iterations* is set to 15
- ni=** the *number_of_loops* is set to 1 if `-robust` is specified and 2 otherwise

- sms=** no smoothing is applied to the seasonal component
- smt=** no smoothing is applied to the trend component
- robust** robustness iterations are not used, by default.
- mi=** the *missing_value* for the output variables is equal to $1.e+20$
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the multi-channel time series must be decomposed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_stl_4d*.

- 3) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 4) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian and as such has no duplicate points.

This argument is also used to determine the name of the NetCDF *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 5) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 6) The **-p=periodicity** argument gives the periodicity of the input data. For example, with monthly data **-p=12** should be specified. The *periodicity* must be greater than 2. If your multi-channel time series do not have a periodic component, e.g., if *periodicity* is equal to 1, use [comp_trend_4d](#) instead of *comp_stl_4d*.
- 7) The **-ns=** argument specifies the length of the seasonal smoother, *ns*. The value of *ns* should be an odd integer greater than or equal to 3; a value greater than 6 is recommended. As the value of the **-ns=** argument increases the values of the seasonal component at a given point in the seasonal cycle (e.g., January values of a monthly

series with a yearly cycle) become smoother. However, the value of `ns` has no direct effect on the smoothness of successive values of the seasonal component.

- 8) The **-nt=** argument specifies the length of the trend smoother, `nt`. The value of `nt` should be an odd integer greater than or equal to 3; a value of `nt` between $1.5 * \textit{periodicity}$ and $2 * \textit{periodicity}$ is recommended. As `nt` increases the values of the trend component become smoother.
- 9) The **-nl=** argument specifies the length of the low-pass smoother, `nl`. The value of `nl` should be an odd integer greater than or equal to 3; the smallest odd integer greater than or equal to *periodicity* is recommended.
- 10) The **-isdeg=** argument specifies the degree of locally-fitted polynomial in seasonal smoothing. The value is 0 or 1.
- 11) The **-itdeg=** argument specifies the degree of locally-fitted polynomial in trend smoothing. The value is 0, 1 or 2.
- 12) The **-ildeg=** argument specifies the degree of locally-fitted polynomial in low-pass smoothing. The value is 0, 1 or 2.
- 13) The **-nsjump=** argument specifies the skipping value for seasonal smoothing. The seasonal smoother skips ahead `nsjump` points and then linearly interpolates in between. The value of `nsjump` should be a positive integer; if `nsjump` is set to 1, a seasonal smooth is calculated at all points in the time series. To make the procedure run faster, a reasonable choice for `nsjump` is 10% or 20% of `ns`. The default value is `ns/10`.
- 14) The **-ntjump=** argument specifies the skipping value for trend smoothing. The default value is `nt/10`.
- 15) The **-nljump=** argument specifies the skipping value for the low-pass filter. The default value is `nl/10`.
- 16) The **-ni=** argument specifies the number of loops for updating the seasonal and trend components. The value of `ni` should be a positive integer. If the data are well behaved without outliers, then robustness iterations are not needed. In this case do not use the **-robust** argument, and set `ni` between 2 and 5 depending on how much security you want that the seasonal-trend looping converges. If outliers are present then use the **-robust** argument and set `ni` to 1 or 2.
- 17) The **-a=** argument specifies if the residuals from the trend and seasonal components are stored in the output NetCDF file. If:
 - **-a=stl** means that the residuals are not computed
 - **-a=residual** means that the residuals are computed and stored
 - **-a=residual_trend** means that the residual and trend components are summed and the result is stored.

The default is **-a=stl**, e.g., the residuals are not stored. Note that in all cases, the seasonal and trend components are computed and stored in the output NetCDF file.
- 18) If **-robust** is specified, robustness iterations are carried out until convergence of both seasonal and trend components or with a maximum of *max_robustness_iterations* iterations as specified by the **-maxit=** argument. Convergence occurs if the maximum changes in individual seasonal and trend fits are less than 1% of the component's range after the previous iteration.
- 19) **-sms=seasonal_smoothing_factor** means that the seasonal component extracted from the `netcdf_variable` (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * \textit{seasonal_smoothing_factor} + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual` or `residual_trend`). *seasonal_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original STL procedure.
- 20) **-smt=trend_smoothing_factor** means that the trend component extracted from the `netcdf_variable` (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * \textit{trend_smoothing_factor} + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual` or `residual_trend`). *trend_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original STL procedure.

- 21) The **-mi=missing_value** argument specifies the missing value indicator for the output variables in the *output_netcdf_file*. If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.
- 22) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 23) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 24) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 25) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 26) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 27) If the time series have no seasonal cycle, use *comp_trend_4d* instead of *comp_stl_4d*.
- 28) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 29) For more details on the STL procedure and examples of use in the climate literature, see
 - “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>
 - “Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation” by Ter-ray, P., Climate Dynamics, Vol. 36, 2171-2199, 2011. doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)

Outputs

comp_stl_4d creates an output NetCDF file that contains the trend and harmonic components extracted from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, *nlev*, *nlat* and *nlon* are the lengths of the geographical dimensions of the input NetCDF variable and *ntime* is the selected number of time observations) :

- 1) *netcdf_variable_trend*(*ntime*, *nlev*, *nlat*, *nlon*) : the trend component for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 2) *netcdf_variable_seasonal*(*ntime*, *nlev*, *nlat*, *nlon*) : the harmonic component for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

- 3) *netcdf_variable_residual*(*ntime*, *nlev*, *nlat*, *nlon*) : the residual component for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-a=residual** argument has been specified when calling `comp_stl_4d`.

- 4) *netcdf_variable_residual_trend*(*ntime*, *nlev*, *nlat*, *nlon*) : the sum of the residual and trend components for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-a=residual_trend** argument has been specified when calling `comp_stl_4d`.

The trend, harmonic and residual multi-channel components are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For computing a STL decomposition from the fourdimensional NetCDF variable called `votemper` extracted from the file `ST7_1m_00101_20012_votemper_grid_T.nc`, which includes monthly time series, and store the results in the NetCDF file `stl_ST7_1m_00101_20012_votemper_grid_T.nc`, use the following commands :

```
$ comp_stl_4d \
-f=ST7_1m_00101_20012_votemper_grid_T.nc \
-v=votemper \
-p=12 \
-ns=35 \
-nt=30 \
-o=stl_ST7_1m_00101_20012_votemper_grid_T.nc
```

2.70 comp_svd_3d

2.70.1 Authors

Pascal Terray (LOCEAN/IPSL) and Eric Maisonnave (CERFACS)

2.70.2 Latest revision

05/05/2026

2.70.3 Purpose

Compute a Singular Values Decomposition (SVD) analysis, also known as Maximum Covariance Analysis (MCA), of a covariance (or correlation) matrix between (selected) time series associated with two input NetCDF variables (specified with the **-v=** and **-v2=** arguments) extracted from one or two NetCDF datasets (specified with the **-f=** and **-f2=** arguments).

SVD analysis or MCA can be considered as a generalization of Empirical Orthogonal Function (EOF) analysis [Bjornsson_Venegas] [Bretherton_etal] [vonStorch_Zwiers]. It aims at estimating the covariance matrix between two

fields and at computing the SVD of this covariance matrix for defining pairs of spatial patterns, which describe (maximize) a fraction of the total Square Covariance (SCF) between the two fields.

The procedure first repacks the first (or left) input tridimensional NetCDF variable (specified with the **-v=** argument) and the second (or right) input tri- or fourdimensional NetCDF variable (specified with the **-v2=** argument) as a *ntime* by *nv1* rectangular matrix, **X**, and a *ntime* by *nv2* rectangular matrix, **Y** respectively. The procedure then computes the covariance (or correlation or sum of squares and cross-products, at the user option) matrix between **X** and **Y**, e.g., the rectangular matrix product $(\text{transpose}(\mathbf{X}) \cdot \mathbf{Y}) / \text{ntime}$ after, eventually centering or standardizing the columns of the **X** and **Y** matrices.

In the following discussion, the **X** and **Y** matrices will be called the left and right fields, respectively.

Optionally, the covariance (or correlation or sum of squares and cross-products) matrix between the left and right fields can be estimated using (diagonal) metrics for the left and right fields such that the covariance matrix is weighted by the surface (or volume) associated with each cell in the grid associated with the input NetCDF variables. This implies that equal areas (or volumes) carry equal weights in the results of the MCA analysis (see the **-d=** and **-d2=** arguments description for more details).

The second step of the SVD analysis is to compute the leading *k* terms of the SVD decomposition of this covariance matrix between the left and right fields, which is given by

$$(\text{transpose}(\mathbf{X}) \cdot \mathbf{Y}) / \text{ntime} = \mathbf{USV}$$

where

- **U** is a *nv1* by *k* matrix with orthonormal columns (the left singular vectors stored columnwise)
- **S** is a square *k* by *k* matrix with nonnegative elements on its principal diagonal and zeros elsewhere (the diagonal elements of **S** are the singular values of the covariance matrix)
- **V** is a *k* by *nv2* matrix with orthonormal rows (the right singular vectors stored rowwise)

This partial SVD decomposition of the covariance matrix between the left and right fields can be computed efficiently by inverse iteration or deflation algorithms without computing the full SVD decomposition of the covariance matrix. At the user option, very fast randomized partial SVD algorithms [Halko_etal] [Li_etal] [Martinsson] can also be used, which will be highly efficient on huge datasets. See description of the **-alg=** argument for more details on the numerical algorithms available for performing the (partial) SVD analysis.

In a third step, the first *k* standardized left and right “Singular Variables” (SV) time series are computed by projecting the left and right fields onto the first *k* left and right singular vectors, respectively. These SV time series play a similar role as principal component time series in an Empirical Orthogonal Function (EOF) analysis. Refer to [comp_eof_3d](#) or [comp_eof_4d](#) for more details on EOF analysis.

Finally, from the *k* left and right SV time series, two types of regression maps are generated for each field: the *k*th homogeneous vector, which is the regression map between a given data field and its *k*th standardized SV time series, and the *k*th heterogeneous vector, which is the regression map between a given data field and the *k*th standardized SV time series of the other field. The *k*th heterogeneous vector indicates how well the time series of one field can be predicted from the *k*th SV time series of the other field.

Simple statistics associated with each singular triplet (e.g., a singular value and the associated left and right singular vectors) of the (partial) SVD of the covariance matrix are also computed. These include the Square Covariance Fraction (SCF) coefficient, which is a simple measure of the relative importance of each singular triplet in the linear relationship between the two fields, the correlation coefficients between the *k*th left and right SV time series of the two fields and the Normalized root-mean-square Covariance (NC) coefficient. See [Bretherton_etal] [Zhang_etal] for a discussion of the relative merits of these coefficients for determining how strongly related the coupled patterns described by the singular triplets are.

At the user option, a moving block bootstrap approach [Efron] [Politis] can also be used to obtain confidence intervals for the SCF, correlation and NC coefficients associated to the singular triplets of the SVD analysis and to test the stability of the computed leading singular triplets. More precisely, statistics (e.g., confidence intervals, means,

standard-deviations, skewness, ...) of SCF, correlation and NC coefficients, cosines of the angles between the pairwise observed and bootstrapped singular vectors, as well as spectral and (scaled) Frobenius distances between the pairwise observed and bootstrapped leading left and right singular subspaces are computed and can be used to select the number of singular triplets for further analysis. Furthermore, the **-nb=**, **-bl=**, **-bp=** and **-bm=** and **-nobias** arguments allow the user to determine the exact form of the blockwise bootstrap algorithm which is used. One of the **-nb=**, **-bl=**, **-bp=** and **-bm=** arguments must be present if you want to test the stability of the computed singular triplets of the SVD analysis by this bootstrap approach and store its results in the output NetCDF datasets.

The notions of distance between two singular subspaces of the same dimension used in the moving block bootstrap approach are based on the unique orthogonal projectors onto the ranges of these two singular subspaces and the distances between the two square matrices representing these two orthogonal projectors as measured in some matrix norms [Golub_VanLoan]. Here, we use both the spectral (e.g., the 2-norm) and Frobenius matrix norms to quantify the distances between two singular subspaces of the same dimension [Golub_VanLoan].

The spectral distance, **S_dist**, of two subspaces of the same dimension is simply equal to the sine of the largest principal angle between these two subspaces [Golub_VanLoan] and is always greater or equal to 0 and less or equal to 1. It is further possible to show that the spectral distance between two subspaces is equal to 0 if and only if the two subspaces are the same. On the other hand, if the spectral distance between two subspaces is equal to 1 then it exists at least one vector belonging to the first subspace, which is orthogonal to all vectors belonging to the second subspace and vice-versa.

On the other hand, it can be shown that the Frobenius distance, **F_dist**, between two subspaces of dimension n is equal to

$$\mathbf{F_dist} = \sqrt{2 \cdot \sum(\sin(PA(:n)))^2}$$

where the elements of the n -vector $PA(:)$ are the n principal angles between the two subspaces of dimension n . The Frobenius distance is equal to 0 if the two subspaces are the same and is always less than $\sqrt{2 \cdot n}$ and this maximum is attained only when the two subspaces are orthogonal to each other. Since this upper bound is a function of the dimension n of the two subspaces, it is further convenient to define the “scaled” Frobenius distance of two subspaces of dimension n as the ratio between the Frobenius distance and its upper bound. In this way, the “scaled” Frobenius distance is always greater or equal to 0 and less or equal to 1, as the spectral distance, and can also be used to compare “scaled” Frobenius distances between pairs of singular subspaces of different dimensions in order to determine the most stable left and right singular subspaces with the help of the moving block bootstrap method.

Note, finally, that `comp_svd2_3d` performs exactly the same task as `comp_svd_3d`, but differs in the form of the moving block bootstrap approach used to assess the significance of the singular triplets of the SVD analysis. In `comp_svd2_3d`, confidence levels for the SCF, NC and correlation coefficients are estimated by a moving block bootstrap algorithm in which these statistics are recomputed many times after replacing the right field by a permuted field constructed by resampling randomly blocks of observations from the original right field. This approach destroys completely the covarying signals between the two fields, excepted the one resulting from the rank deficiency of the covariance matrix (e.g., if the number of time observations is less than the number of grid cells in the left and right fields). This approach is useful if you want to assess that the analyzed covariance matrix is above the “noise”. However, if you already know that some covarying signals are present in the analyzed covariance matrix, the bootstrap approach used in `comp_svd_3d` is more appropriate and meaningful as this approach allows to assess the sampling distributions and stability of the SVD results without destroying the covarying signals in the bootstrapped left and right fields.

Two output NetCDF datasets containing the singular values, the left and right singular vectors, the corresponding left and right standardized SV time series and, the homogeneous and heterogeneous vectors for each field are created. The left and right singular vectors, and the homogeneous and heterogeneous vectors for each field, are repacked onto the original grids of the two input NetCDF variables in the output NetCDF datasets. In addition, if the bootstrap approach is used, the two output NetCDF datasets will also contain confidence intervals for the SCF, NC and correlation coefficients as well as statistics for SCF, NC and correlation coefficients, cosines of the angles between the pairwise observed and bootstrapped singular vectors and distances between the pairwise observed and bootstrapped leading singular subspaces.

This procedure is parallelized if OpenMP is used and will be also much faster if an optimized BLAS library is specified at compilation with the `_BLAS` CPP macro. Moreover, this procedure may use (randomized) partial SVD algorithms

[[Martinsson](#)], which are highly efficient on huge covariance matrices if you are interested only in the few leading terms of the SVD of the covariance matrix between the **X** and **Y** fields.

2.70.4 Further Details

Usage

```
$ comp_svd_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -a=type_of_analysis (optional : scp, cov, cor) \
  -n=number_of_sing_triplets (optional) \
  -g=grid_type (optional : n, t, u, v, w, f) \
  -r=resolution (optional : r2, r4) \
  -b=nlon_orca, nlat_orca (optional) \
  -x=lon1,lon2 (optional) \
  -y=lat1,lat2 (optional) \
  -t=time1,time2 (optional) \
  -c=input_climatology_netcdf_file (optional) \
  -d=distance (optional : dist2, ident) \
  -o=output_svd_netcdf_file_left_field (optional) \
  -f2=input_netcdf_file_right_field (optional) \
  -v2=netcdf_variable_right_field (optional) \
  -m2=input_mesh_mask_netcdf_file_right_field (optional) \
  -g2=grid_type_right_field (optional : n, t, u, v, w, f) \
  -r2=resolution_right_field (optional : r2, r4) \
  -b2=nlon_orca, nlat_orca (optional) \
  -x2=lon1_right_field,lon2_right_field (optional) \
  -y2=lat1_right_field,lat2_right_field (optional) \
  -z2=level1_right_field,level2_right_field (optional) \
  -t2=time1_right_field,time2_right_field (optional) \
  -c2=input_climatology_netcdf_file_right_field (optional) \
  -d2=distance_right_field (optional : dist2, dist3, ident) \
  -o2=output_svd_netcdf_file_right_field (optional) \
  -alg=algorithm (optional : svd, inviter, deflate, r
  →rsvd) \
  -bm=bootstrap_method (optional : normal, percentile) \
  -nb=number_of_shuffles (optional) \
  -bp=bootstrap_periodicity (optional) \
  -bl=bootstrap_block_length (optional) \
  -pro=critical_probability (optional : 0.0 > 1.) \
  -mi=missing_value (optional) \
  -ortho (optional) \
  -nobias (optional) \
  -double (optional) \
  -bigfile (optional) \
  -hdf5 (optional) \
  -cdf5 (optional) \
  -tlimited (optional)
```

By default

-a= the *type_of_analysis* is set to *scp*. This means that the singular vectors and singular values are computed from the sums of squares and cross-products matrix between the left and right fields

- n=** *number_of_sing_triplets* is set to 4. This means that the first 4 singular triplets of the covariance matrix between the left and right fields are computed and stored in the output NetCDF files *output_svd_netcdf_file_left_field* and *output_svd_netcdf_file_right_field*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the left field extracted from the input NetCDF variable *input_netcdf_file* is assumed to be regular or Gaussian
- r=** if the input NetCDF variable *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, of the left input NetCDF variable *netcdf_variable* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- c=** this argument is not used. This argument is required only if the *type_of_analysis* is set to *cov* or *cor* and is used to specify the climatology NetCDF file for computing anomalies or standardized anomalies for the left field
- d=** the *distance* is set to *dist2*. This means that distances and scalar products for the left field in the SVD analysis are computed with the diagonal metric associated with the 2-D grid-mesh associated with the input NetCDF variable *netcdf_variable*
- o=** the *output_svd_netcdf_file_left_field* is named *svd_left_netcdf_variable.nc*
- f2=** this argument is not used. It is required only if the right field is not stored in the same file as the left NetCDF variable
- v2=** this argument is not used. It is required only if the right field is not extracted from the same input NetCDF variable than the left field
- m2=** this argument will take the same value as the **-m=** argument. It is required only if the right field is not extracted from the same input NetCDF variable as the left field
- g2=** same as the **-g=** argument if the **-v2=** argument is omitted and **-g2=n** otherwise
- r2=** same as the **-r=** argument if **-v2=** is omitted and **-r2=r2** otherwise
- b2=** if **-g2=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, of the right field extracted from the input NetCDF variable *netcdf_variable_right_field* are determined from the **-r2=** argument. However, you may override this choice by default with the **-b2=** argument
- x2=** the whole longitude domain associated with the *netcdf_variable_right_field*
- y2=** the whole latitude domain associated with the *netcdf_variable_right_field*
- z2=** the whole level associated with the *netcdf_variable_right_field*
- t2=** the whole time period associated with the *netcdf_variable_right_field*
- c2=** this argument is not used. This argument is required only if the *type_of_analysis* is set to *cov* or *cor* and is used to specify the climatology NetCDF file for computing anomalies or standardized anomalies for the right field if this field is not extracted from the same input NetCDF variable as the left field
- d2=** same as **-d=** if **-v2=** is omitted, **-d2=dist2** if the *netcdf_variable_right_field* is a 3D variable and **-d2=dist3** if the *netcdf_variable_right_field* is a 4D variable
- o2=** the *output_svd_netcdf_file_right_field* is named *svd_right_netcdf_variable_right_field.nc*

- alg=** the *algorithm* option is set to *inviter*. This means that the SVD analysis is computed by a partial SVD of the covariance matrix using an inverse iteration algorithm
- bm=** *bootstrap_method* is set to *normal*. This means that bootstrap confidence intervals of SCF, NC and correlation coefficients are based on asymptotic normality
- nb=** *number_of_shuffles* is set to 99. This means that 99 bootstrap samples are generated in the moving block bootstrap algorithm for testing the significance and stability of the singular triplets
- bp=** the time series are assumed to be stationary and *bootstrap_periodicity* is set to 1 in the moving block bootstrap procedure for testing the significance and stability of the singular triplets. This means that the blocks in the bootstrap algorithm are not forced to begin at specific observations. Use this parameter if the time series are cyclostationary, see the remarks below for further details
- bl=** *bootstrap_block_length* is set to *bootstrap_periodicity*.2
- pro=** the *critical_probability* is set to 0.05, which means that the bootstrap confidence intervals for the SCF, NC and correlation coefficients are 95% confidence intervals
- mi=** the *missing_value* attribute in the output NetCDF files is set to 1.e+20
- ortho** the computed singular vectors and associated SV time series are not reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods, e.g., if **-alg=inviter** or **-alg=deflate**
- nobias** biased bootstrap confidence intervals for SCF, NC and correlation coefficients are computed. However, if **-nobias** is activated, unbiased bootstrap confidence intervals for SCF, NC and correlation coefficients are computed
- double** the results of the SVD analysis are stored as single-precision floating point numbers in the output NetCDF files. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** NetCDF classical format files are created. If **-bigfile** is activated, the output NetCDF files are 64-bit offset format files
- hdf5** NetCDF classical format files are created. If **-hdf5** is activated, the output NetCDF files are NetCDF-4/HDF5 format files
- cdf5** NetCDF classical format files are created. If **-cdf5** is activated, the output NetCDF files are CDF5 format files
- tlimited** the time dimension is defined as unlimited in the output NetCDF files. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF files

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the left field for the SVD analysis must be extracted and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-v2=netcdf_variable_right_field** argument specifies the NetCDF variable from which the right field of the SVD analysis must be extracted and the **-f2=input_netcdf_file_right_field** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file_right_field*.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the geographical domain used for defining the left field in the SVD analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_typemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*) which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing and the **-x=** and **-y=** arguments are also not specified, the whole geographical domain associated with the *netcdf_variable* is used for defining the left field in the SVD analysis.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

This remark applies also for the **-x2=**, **-y2=** and **-z2=** arguments used for defining the right field in the SVD analysis.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using `comp_svd_3d`.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to estimate the covariance matrix between the left and right fields.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the SV time series in the output NetCDF files will have $n_{time} = time2 - time1 + 1$ time observations.

This remark applies also for **-t2=** argument used to define the time dimension of the right field.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable *netcdf_variable* is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed when extracting the left field of the SVD analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe. On output, the duplicate points are restored when writing the SVD results (e.g., the singular, homogeneous and heterogeneous vectors), if the geographical domain of the input NetCDF variable *netcdf_variable* is the whole globe.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh associated with *netcdf_variable* is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typedmask*, *e1grid_type* and *e2grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by [comp_clim_3d](#) if the 2-D grid-mesh is regular or gaussian.

This remark applies also for **-g2=** argument used to define the grid type of the right field.

- 6) If **-g=** is set to *t*, *u*, *v*, *w* or *f* (e.g., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:
 - **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration
 - **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration.

This remark applies also for **-g2=** argument used to define the grid type of the right field.

- 7) If the NetCDF variable *netcdf_variable* is from an experiment with the NEMO ocean model, but the resolution is not *r2* or *r4*, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

This remark applies also for **-b2=** argument used to define the grid type of the right field.

- 8) The **-a=** argument specifies if the left and right fields are centered or standardized with an input climatology (specified with the **-c=** and **-c2=** arguments) before computing the covariance matrix between the two fields. If:
 - **-a=scp**, the SVD analysis is done on the raw data of the two fields
 - **-a=cov**, the SVD analysis is done on the anomalies of the two fields
 - **-a=cor**, the SVD analysis is done on the standardized anomalies of the two fields

By default, the **-a=** argument is set to *scp*.

- 9) The *input_climatology_netcdf_file* and *input_climatology_netcdf_file_right_field* specified, respectively, with the **-c=** and **-c2=** arguments are needed only if **-a=cov** or **-a=cor**.

- 10) If **-a=cov** or **-a=cor**, the selected time periods for the left and right fields specified, respectively, with the **-t=** and **-t2=** arguments, must agree with the periods used to estimate the climatologies.

This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument for the left field. This remark also applies for the right field and the **-t2=** and **-c2=** arguments.

- 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.

Similarly, for the right field, the geographical shapes of the *netcdf_variable_right_field* (in the *input_netcdf_file_right_field*), the mask (in the *input_mesh_mask_netcdf_file_right_field*), the scale factors (in the *input_mesh_mask_netcdf_file_right_field*), and the climatology (in the *input_climatology_netcdf_file_right_field*) must agree.

- 12) The **-n=number_of_sing_triplets** argument specifies the number of singular triplets of the SVD of the covariance matrix between the left and right fields, which must be stored (and also computed if **-alg=inviter**, **-alg=deflate** or **-alg=rsvd** is specified) in the output NetCDF files given in the **-o=** and **-o2=** arguments. The default value is 4.

- 13) The **-d=distance** argument specifies the metric and scalar product used for the left field in the SVD analysis. If:

- **-d=dist2**, the SVD analysis is done with the diagonal distance associated with the horizontal 2-D grid-mesh of the left field (e.g., each grid point is weighted accordingly to the surface associated with it)
- **-d=ident**, the SVD analysis is done with the identity metric : the Euclidean distance and the usual scalar product is used in the SVD analysis.

By default, the **-d=** argument is set to **dist2**.

This remark applies also for the **-d2=** argument used to define the distance and scalar product of the right field in the SVD analysis.

If the second NetCDF variable *netcdf_variable_right_field*, used to define the right field in the SVD analysis, has 4 dimensions, the following value is also allowed for the **-d2=** argument:

- **-d=dist3** meaning that the SVD analysis is done with the diagonal distance associated with the whole 3D grid of the right field (e.g., each grid point is weighted accordingly to the volume or weight associated with it).

By default, the **-d2=** argument is set to the same value as the **-d=** argument if **-v2=** is omitted, and, otherwise, **-d2=dist2** if the *netcdf_variable_right_field* is a 3D variable and **-d2=dist3** if the *netcdf_variable_right_field* is a 4D variable.

- 14) The **-alg=** argument determines how the singular values and vectors of the covariance matrix between the left and right fields are computed. If:

- **-alg=svd**, a full SVD of the rectangular covariance matrix is computed.
- **-alg=inviter**, a partial SVD of the rectangular covariance matrix is computed by inverse iteration.
- **-alg=deflate**, a partial SVD of the rectangular covariance matrix is computed by a deflation technique.
- **-alg=rsvd**, a partial SVD of the rectangular covariance matrix is computed by a very fast randomized algorithm.

All algorithms are parallelized if OpenMP is used. The default is **-alg=inviter** since computing a partial SVD is generally much faster than computing a full SVD. But, **-alg=deflate** is generally as fast as **-alg=inviter**. **-alg=rsvd** is generally much faster than all other options, but the computed singular vectors and SV variables may be less accurate depending on the shape of the distribution of the singular values of the covariance matrix.

- 15) If the **-ortho** argument is used, the computed singular vectors and associated SV time series are always reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods. By default, they are only partially orthogonalized if the computed singular values are not well separated. Note that this argument has no effect if **-alg=svd** or **-alg=rsvd**.
- 16) If any of the **-nb=**, **-bl=**, **-bp=** and **-bm=** arguments is specified, a moving block bootstrap algorithm is used to test the significance and stability of the computed singular triplets and help to select the number of significant singular triplets for further analysis.
- 17) The **-bm=bootstrap_method** argument specifies how bootstrap confidence intervals for SCF, NC and correlation coefficients are computed. If:
 - **-bm=normal**, bootstrap confidence intervals are based on asymptotic normality [Efron] [Politis] . These standard intervals will be bias-corrected [Politis] if the **-nobias** argument is also specified.
 - **-bm=percentile**, bootstrap confidence intervals are based on the percentile or bias-corrected percentile methods [Efron] . The bias-corrected percentile method will be used if the **-nobias** argument is also specified.

Note that the percentile method requires a larger number of shuffles (specified with the **-nb=** argument), especially if the *critical_probability* (specified with the **-pro=** argument) is small. More precisely, *number_of_shuffles.critical_probability.0.5* must be greater or equal to one in the case of the percentile method. If it is not the case, the procedure will exit with an error message. As an illustration, with the default *critical_probability* (e.g., 0.05), *number_of_shuffles* must be at least 40 for using the percentile method for estimating confidence intervals for SCFs, NCs and correlations between singular variables.

By default, the **-bm=** argument is set to `normal`.

- 18) The **-nb=number_of_shuffles** argument specifies the number of shuffles used in the bootstrap algorithm (by default 99).
- 19) The **-bp=bootstrap_periodicity** argument specifies that the index, i , of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where j is a random positive integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. This parameter is useful if the time series are cyclostationary instead of stationary. By default, *bootstrap_periodicity* is set to 1.
- 20) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater or equal to the *bootstrap_periodicity* and less than the length of the time series. If **-a=cov** or **-a=cor** is specified, it is highly recommended to set *bootstrap_block_length* as a multiple of the periodicity in the data as this will take properly into account the cyclostationary of the analyzed time series.

By default, *bootstrap_block_length* is set to *bootstrap_periodicity.2*. If you perform a MCA on huge datasets and you require bootstrap testing of the MCA results, the elapsed time will be largely reduced if you use a large *bootstrap_block_length*, especially if the NCSTAT software has been built with the `_BLAS` macro (e.g., if BLAS support has been activated). In such cases, *bootstrap_block_length* values between 24 and 122 (depending on the computer) speed significantly the bootstrap computations.
- 21) The **-pro=critical_probability** argument specifies the critical probability which is used to determine the lower and upper bounds of the confidence intervals for SCF, correlation and NC coefficients if the bootstrap method is used. More precisely, Lower and upper $(1 - \text{critical_probability}) \cdot 100\%$ bootstrap confidence bounds for SCF, correlation and NC coefficients are computed based on asymptotic normality or the percentile method depending on the value of the **-bm=bootstrap_method** argument. *critical_probability* must be greater than 0. and less than 1.. The default value is 0.05, meaning that 95% confidence intervals are computed by default.
- 22) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_netcdf_file* and *output_netcdf_file_right_field*.

If the **-mi=** argument is not specified *missing_value* is set to `1.e+20`.

- 23) If **-nobias** is specified, unbiased confidence intervals for SCF, correlation and NC coefficients are computed.

If the **-nobias** argument is absent, biased standard confidence intervals are estimated by default.

- 24) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF files.

By default, the results are stored as single-precision floating point numbers in the output NetCDF files.

- 25) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the output netcdf files will be 64-bit offset format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 26) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the output netcdf files will be NetCDF-4/HDF5 format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 27) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the output netcdf files will be CDF5 format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 28) It is assumed that the data has no missing values.

- 29) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 30) For more details on SVD (e.g., MCA) analysis in the climate literature, the bootstrap approach, distances between subspaces and randomized SVD algorithms, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “An intercomparison of methods for finding coupled patterns in climate data”, by Bretherton, Smith, C., and Wallace, J.M., Journal of Climate, Vol. 5, 541-560 pp, 1992. doi: [10.1175/1520-0442\(1992\)005<0541:AIOMFF>2.0.CO;2](https://doi.org/10.1175/1520-0442(1992)005<0541:AIOMFF>2.0.CO;2)
- “Seasonality of large scale atmosphere-ocean interaction over the North Pacific”, by Zhang, Y., Norris, J.R., and Wallace, J.M., Journal of Climate, Vol. 11, 2473-2481 pp, 1998. doi: [10.1175/1520-0442\(1998\)011<2473:SOLSAO>2.0.CO;2](https://doi.org/10.1175/1520-0442(1998)011<2473:SOLSAO>2.0.CO;2)
- “Matrix Computations”, by Golub, G.H., and Van Loan, C., The Johns Hopkins University Press, Baltimore, MD., 4rd Edition, 756 pp., 2013. ISBN: **‘978-1-4214-0794-4’**
- “Statistical Analysis in Climate Research”, by von Storch, H., and Zwiers, F.W., Cambridge University press, Cambridge, UK, Chapter 13, 484 pp., 2002. ISBN: [9780511526357](https://doi.org/10.1017/CBO9780511526357)
- “Nonparametric standard-errors and confidence intervals” By Efron, B., The Canadian Journal of Statistics, 9, 2, 139-172, (1981). <https://doi.org/10.2307/3314608>

- “Computer-Intensive Methods in Statistical Analysis” By Politis, D.N., IEEE Signal Processing Magazine, 15, 1, 39-55, (January 1998). <https://ieeexplore.ieee.org/document/647042>
- “Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions”, By Halko, N., Martinsson, P.G., and Tropp, J.A., SIAM Rev., 53, 217-288, 2011. <https://epubs.siam.org/doi/abs/10.1137/090771806>
- “Algorithm 971: An implementation of a randomized algorithm for principal component analysis”, by Li, H., Linderman, G.C., Szlam, A., Stanton, K.P., Kluger, Y., and Tygert, M., ACM Trans. Math. Softw. 43, 3, Article 28 (January 2017). <https://pubmed.ncbi.nlm.nih.gov/28983138>
- “Randomized methods for matrix computations”, by Martinsson, P.G., arXiv.1607.01649, 2019. <https://arxiv.org/abs/1607.01649>

Outputs

`comp_svd_3d` creates two output NetCDF files. The first output file (specified in the `-o=output_svd_netcdf_file_left_field` argument) contains all the SVD statistics associated with the left field (specified in the `-v=netcdf_variable` argument) and the second output file (specified in the `-o2=output_svd_netcdf_file_right_field` argument) the SVD statistics associated with the right field (specified in the `-v2=netcdf_variable_right_field` argument).

The `output_svd_netcdf_file_left_field` contains the singular values and left singular vectors of the covariance matrix, the left homogeneous and heterogeneous vectors, and the left SV time series of the SVD analysis. The number of SV time series, regression vectors, singular vectors and singular values stored in the output NetCDF dataset is determined by the `-n=number_of_sing_triplets` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the first input NetCDF variable `netcdf_variable`):

- 1) `netcdf_variable_svd` (`number_of_sing_triplets`, `nlat`, `nlon`) : the `number_of_sing_triplets` leading left singular vectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix between the left and right fields. The singular vectors are sorted by descending order of the associated singular values.

The left singular vectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values. If this is a problem, you can use `comp_norm_3d` for restricting the geographical domain in the input dataset before using `comp_svd_3d`.

- 2) `netcdf_variable_hom` (`number_of_sing_triplets`, `nlat`, `nlon`) : the selected left homogeneous vectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix. The left homogeneous vectors are sorted by descending order of the associated singular values. These vectors are scaled such that they give the scalar products (`-a=scp`), covariances (`-a=cov`) or correlations (`-a=cor`) between the original observed time series in the left field and the left SV time series.

The left homogeneous vectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

- 3) `netcdf_variable_het` (`number_of_sing_triplets`, `nlat`, `nlon`) : the selected left heterogeneous vectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix. The left heterogeneous vectors are sorted by descending order of the associated singular values. These vectors are scaled such that they give the scalar products (`-a=scp`),

covariances (**-a=cov**) or correlations (**-a=cor**) between the original observed time series in the left field and the right SV time series stored in the other output NetCDF file.

The left heterogeneous vectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=** and **-y=** arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

- 4) *netcdf_variable_sv*(*ntime*, *number_of_sing_triplets*) : the left SV time series sorted by descending order of the singular values.

The SV time series are always standardized to unit variance.

- 5) *netcdf_variable_SV_STDs*(*number_of_sing_triplets*) : the standard-deviations of the left SV time series sorted by descending order of the singular values.
- 6) *netcdf_variable_SV_EXPVARs*(*number_of_sing_triplets*) : the proportion of variance of the left field explained by the left SV time series sorted by descending order of the singular values.
- 7) *netcdf_variable_CORSV*(*number_of_sing_triplets*) : the symmetric correlation matrix between the left SV time series, only the upper triangle of this symmetric matrix is written in the output file.
- 8) *netcdf_variable_RAW_VAR*(1) : the raw variance (or inertia if **-a=scp**) of the left field averaged over the selected domain.
- 9) *Sing_vals*(*number_of_sing_triplets*) : the singular values of the sums of squares and cross-products (if **-a=scp**) or covariance (if **-a=cov**) or correlation (if **-a=cor**) matrix between the left and right fields sorted in decreasing order.
- 10) *SCFs*(*number_of_sing_triplets*) : the Squared Covariance Fractions (SCF) described by each of the leading singular triplets of the squares and cross-products (if **-a=scp**) or covariance (if **-a=cov**) or correlation (if **-a=cor**) matrix between the left and right fields.
- 11) *NCs*(*number_of_sing_triplets*) : the Normalized root-mean-square Covariance (NC) coefficients associated with each of the leading singular triplets of the squares and cross-products (if **-a=scp**) or covariance (if **-a=cov**) or correlation (if **-a=cor**) matrix between the left and right fields.
- 12) *Corrs*(*number_of_sing_triplets*) : the correlation coefficients between the corresponding left and right SV time series.

Furthermore, if one of the **-nb=**, **-bl=**, **-bp=** or **-bm=** arguments is specified when calling *comp_svd_3d*, the following NetCDF variables are computed and also stored in the *output_svd_netcdf_file_left_field* NetCDF file:

- 1) *SCFs_boot_lower_bound*(*number_of_sing_triplets*) : lower (1-critical_probability)*100% bootstrap confidence bounds for SCF statistics. By default, critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.
- 2) *SCFs_boot_upper_bound*(*number_of_sing_triplets*) : upper (1-critical_probability)*100% bootstrap confidence bounds for SCF statistics. By default, critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.

- 3) *SCFs_boot_mean*(number_of_sing_triplets) : the mean of bootstrapped SCF statistics.
- 4) *SCFs_boot_var*(number_of_sing_triplets) : the variance of bootstrapped SCF statistics.
- 5) *SCFs_boot_std*(number_of_sing_triplets) : the standard-deviation of bootstrapped SCF statistics.
- 6) *SCFs_boot_skew*(number_of_sing_triplets) : the skewness of bootstrapped SCF statistics.
- 7) *SCFs_boot_kurt*(number_of_sing_triplets) : the kurtosis of bootstrapped SCF statistics.
- 8) *SCFs_boot_min*(number_of_sing_triplets) : the minimum of bootstrapped SCF statistics.
- 9) *SCFs_boot_max*(number_of_sing_triplets) : the maximum of bootstrapped SCF statistics.
- 10) *NCs_boot_lower_bound*(number_of_sing_triplets) : lower (1-critical_probability)*100% bootstrap confidence bounds for NC statistics. By default, critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.
- 11) *NCs_boot_upper_bound*(number_of_sing_triplets) : upper (1-critical_probability)*100% bootstrap confidence bounds for NC statistics. By default, critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.
- 12) *NCs_boot_mean*(number_of_sing_triplets) : the mean of bootstrapped NC statistics.
- 13) *NCs_boot_var*(number_of_sing_triplets) : the variance of bootstrapped NC statistics.
- 14) *NCs_boot_std*(number_of_sing_triplets) : the standard-deviation of bootstrapped NC statistics.
- 15) *NCs_boot_skew*(number_of_sing_triplets) : the skewness of bootstrapped NC statistics.
- 16) *NCs_boot_kurt*(number_of_sing_triplets) : the kurtosis of bootstrapped NC statistics.
- 17) *NCs_boot_min*(number_of_sing_triplets) : the minimum of bootstrapped NC statistics.
- 18) *NCs_boot_max*(number_of_sing_triplets) : the maximum of bootstrapped NC statistics.
- 19) *Corrs_boot_lower_bound*(number_of_sing_triplets) : lower (1-critical_probability)*100% bootstrap confidence bounds for correlation coefficients between left and right SV variables. By default, critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.
- 20) *Corrs_boot_upper_bound*(number_of_sing_triplets) : upper (1-critical_probability)*100% bootstrap confidence bounds for correlation coefficients between left and right SV variables. By default, critical_probability is set to 0.05, such that 95% confidence intervals are computed, but the width of the confidence intervals can be changed at the user option with the **-pro=critical_probability** argument. Similarly, by default, biased confidence intervals are computed, but unbiased confidence intervals are estimated if the **-nobias=** option is used.

- 21) *Corrs_boot_mean*(number_of_sing_triplets) : the mean of bootstrapped correlation coefficients between SV variables.
- 22) *Corrs_boot_var*(number_of_sing_triplets) : the variance of bootstrapped correlation coefficients between SV variables.
- 23) *Corrs_boot_std*(number_of_sing_triplets) : the standard-deviation of bootstrapped correlation coefficients between SV variables.
- 24) *Corrs_boot_skew*(number_of_sing_triplets) : the skewness of bootstrapped correlation coefficients between SV variables.
- 25) *Corrs_boot_kurt*(number_of_sing_triplets) : the kurtosis of bootstrapped correlation coefficients between SV variables.
- 26) *Corrs_boot_min*(number_of_sing_triplets) : the minimum of bootstrapped correlation coefficients between SV variables.
- 27) *Corrs_boot_max*(number_of_sing_triplets) : the maximum of bootstrapped correlation coefficients between SV variables.
- 28) *netcdf_variable_cos_vec_boot_mean*(number_of_sing_triplets) : the mean of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 29) *netcdf_variable_cos_vec_boot_var*(number_of_sing_triplets) : the variance of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 30) *netcdf_variable_cos_vec_boot_std*(number_of_sing_triplets) : the standard-deviation of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 31) *netcdf_variable_cos_vec_boot_skew*(number_of_sing_triplets) : the skewness of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 32) *netcdf_variable_cos_vec_boot_kurt*(number_of_sing_triplets) : the kurtosis of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 33) *netcdf_variable_cos_vec_boot_min*(number_of_sing_triplets) : the minimum of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 34) *netcdf_variable_cos_vec_boot_max*(number_of_sing_triplets) : the maximum of the cosines of the angles between pairwise observed and bootstrapped left singular vectors.
- 35) *netcdf_variable_spect_dist_subspace_boot_mean*(number_of_sing_triplets) : the mean of the the spectral distances between pairwise observed and bootstrapped left singular subspaces.
- 36) *netcdf_variable_spect_dist_subspace_boot_var*(number_of_sing_triplets) : the variance of the spectral distances between pairwise observed and bootstrapped left singular subspaces.
- 37) *netcdf_variable_spect_dist_subspace_boot_std*(number_of_sing_triplets) : the standard-deviation of the spectral distances between pairwise observed and bootstrapped left singular subspaces.
- 38) *netcdf_variable_spect_dist_subspace_boot_skew*(number_of_sing_triplets) : the skewness of the spectral distances between pairwise observed and bootstrapped left singular subspaces.
- 39) *netcdf_variable_spect_dist_subspace_boot_kurt*(number_of_sing_triplets) : the kurtosis of the spectral distances between the pairwise observed and bootstrapped left singular subspaces.
- 40) *netcdf_variable_spect_dist_subspace_boot_min*(number_of_sing_triplets) : the minimum of the spectral distances between pairwise observed and bootstrapped left singular subspaces.
- 41) *netcdf_variable_spect_dist_subspace_boot_max*(number_of_sing_triplets) : the maximum of the spectral distances between pairwise observed and bootstrapped left singular subspaces.

- 42) `netcdf_variable_frob_dist_subspace_boot_mean(number_of_sing_triplets)` : the mean of the (scaled) Frobenius distances between pairwise observed and bootstrapped left singular subspaces.
- 43) `netcdf_variable_frob_dist_subspace_boot_var(number_of_sing_triplets)` : the variance of the (scaled) Frobenius distances between pairwise observed and bootstrapped left singular subspaces.
- 44) `netcdf_variable_frob_dist_subspace_boot_std(number_of_sing_triplets)` : the standard-deviation of the (scaled) Frobenius distances between pairwise observed and bootstrapped left singular subspaces.
- 45) `netcdf_variable_frob_dist_subspace_boot_skew(number_of_sing_triplets)` : the skewness of the (scaled) Frobenius distances between pairwise observed and bootstrapped left singular subspaces.
- 46) `netcdf_variable_frob_dist_subspace_boot_kurt(number_of_sing_triplets)` : the kurtosis of the (scaled) Frobenius distances between the pairwise observed and bootstrapped left singular subspaces.
- 47) `netcdf_variable_frob_dist_subspace_boot_min(number_of_sing_triplets)` : the minimum of the (scaled) Frobenius distances between pairwise observed and bootstrapped left singular subspaces.
- 48) `netcdf_variable_frob_dist_subspace_boot_max(number_of_sing_triplets)` : the maximum of the (scaled) Frobenius distances between pairwise observed and bootstrapped left singular subspaces.

The `output_svd_netcdf_file_right_field` contains the singular values and right singular vectors of the covariance matrix, the right homogeneous and heterogeneous vectors, and the right SV time series of the SVD analysis. The number of SV time series, regression vectors and singular vectors and singular values stored in this output NetCDF dataset is also determined by the `-n=number_of_sing_triplets` argument. This output NetCDF dataset contains exactly the same NetCDF variables than the first NetCDF output file, but for the statistics of the right field instead of the left field. Refer to the description above for the content and definition of the NetCDF variables in the file `output_svd_netcdf_file_left_field`.

Examples

- 1) For computing an SVD analysis from two NetCDF variables `sst` and `slp` stored, respectively, in the NetCDF files `HadISST1_1m_1979_2005_sst.nc` and `hads1p_1m_1979_2005_slp.nc` use the following command (note that the analysis is done on the anomalies after removing the annual cycle for each variable since `-a=cov` is specified) :

```
$ comp_svd_3d \
-a=cov \
-n=5 \
-f=HadISST1_1m_1979_2005_sst.nc \
-v=sst \
-c=clim_HadISST1_1m_1979_2005_sst.nc \
-x=111,330 \
-y=101,140 \
-m=mask_HadISST1_sst.nc \
-o=svd_HadISST1_1m_1979_2005_sst_oia1.nc \
-f2=hads1p_1m_1979_2005_slp.nc \
-v2=slp \
-c2=clim_hads1p_1m_1979_2005_slp.nc \
-x2=-14,31 \
```

(continues on next page)

(continued from previous page)

```
-y2=21,33 \
-m=mask_hadslp_slp.nc \
-o2=svd_hadslp_1m_1979_2005_slp_oiatl.nc
```

2.71 comp_svd2_3d

2.71.1 Authors

Pascal Terray (LOCEAN/IPSL) and Eric Maisonnave (CERFACS)

2.71.2 Latest revision

29/04/2026

2.71.3 Purpose

Compute a Singular Values Decomposition (SVD) analysis, also known as Maximum Covariance Analysis (MCA), of a covariance (or correlation) matrix between (selected) time series associated with two input NetCDF variables (specified with the **-v=** and **-v2=** arguments) extracted from one or two NetCDF datasets (specified with the **-f=** and **-f2=** arguments).

SVD analysis or MCA can be considered as a generalization of Empirical Orthogonal Function (EOF) analysis [Bjornsson_Venegas] [Bretherton_etal] [vonStorch_Zwiers]. It aims at estimating the covariance matrix between two fields and at computing the SVD of this covariance matrix for defining pairs of spatial patterns, which describe (maximize) a fraction of the total Square Covariance (SCF) between the two fields.

The procedure first repacks the first (or left) input tridimensional NetCDF variable (specified with the **-v=** argument) and the second (or right) input tri- or fourdimensional NetCDF variable (specified with the **-v2=** argument) as a *ntime* by *nv1* rectangular matrix, **X**, and a *ntime* by *nv2* rectangular matrix, **Y** respectively. The procedure then computes the covariance (or correlation or sum of squares and cross-products, at the user option) matrix between **X** and **Y**, e.g., the rectangular matrix product $(\text{transpose}(\mathbf{X}) \cdot \mathbf{Y}) / \text{ntime}$ after, eventually centering or standardizing the columns of the **X** and **Y** matrices.

In the following discussion, the **X** and **Y** matrices will be called the left and right fields, respectively.

Optionally, the covariance (or correlation or sum of squares and cross-products) matrix between the left and right fields can be estimated using (diagonal) metrics for the left and right fields such that the covariance matrix is weighted by the surface (or volume) associated with each cell in the grid associated with the input NetCDF variables. This implies that equal areas (or volumes) carry equal weights in the results of the MCA analysis (see the **-d=** and **-d2=** arguments description for more details).

The second step of the SVD analysis is to compute the leading *k* terms of the SVD decomposition of this covariance matrix between the left and right fields, which is given by

$$(\text{transpose}(\mathbf{X}) \cdot \mathbf{Y}) / \text{ntime} = \mathbf{USV}$$

where

- **U** is a *nv1* by *k* matrix with orthonormal columns (the left singular vectors stored columnwise)
- **S** is a square *k* by *k* matrix with nonnegative elements on its principal diagonal and zeros elsewhere (the diagonal elements of **S** are the singular values of the covariance matrix)
- **V** is a *k* by *nv2* matrix with orthonormal rows (the right singular vectors stored rowwise)

This partial SVD decomposition of the covariance matrix between the left and right fields can be computed efficiently by inverse iteration or deflation algorithms without computing the full SVD decomposition of the covariance matrix. At the user option, very fast randomized partial SVD algorithms [Halko_etal] [Li_etal] [Martinsson] can also be used, which will be highly efficient on huge datasets. See description of the **-alg=** argument for more details on the numerical algorithms available for performing the (partial) SVD analysis.

In a third step, the first k standardized left and right “Singular Variables” (SV) time series are computed by projecting the left and right fields onto the first k left and right singular vectors, respectively. These SV time series play a similar role as principal component time series in an Empirical Orthogonal Function (EOF) analysis. Refer to [comp_eof_3d](#) or [comp_eof_4d](#) for more details on EOF analysis.

Finally, from the k left and right SV time series, two types of regression maps are generated for each field: the k th homogeneous vector, which is the regression map between a given data field and its k th standardized SV time series, and the k th heterogeneous vector, which is the regression map between a given data field and the k th standardized SV time series of the other field. The k th heterogeneous vector indicates how well the time series of one field can be predicted from the k th SV time series of the other field.

Simple statistics associated with each singular triplet (e.g., a singular value and the associated left and right singular vectors) of the (partial) SVD of the covariance matrix are also computed. These include the Square Covariance Fraction (SCF) coefficient, which is a simple measure of the relative importance of each singular triplet in the linear relationship between the two fields, the correlation coefficients between the k th left and right SV time series of the two fields and the Normalized root-mean-square Covariance (NC) coefficient. See [Bretherton_etal] [Zhang_etal] for a discussion of the relative merits of these coefficients for determining how strongly related the coupled patterns described by the singular triplets are.

At the user option, confidence levels for the SCF, NC and correlation coefficients can be estimated by a moving block bootstrap algorithm in which these statistics are recomputed many times after replacing the right field by a permuted field constructed by resampling randomly blocks of observations from the original right field. These bootstrap confidence levels are estimated only if any of the optional bootstrap arguments (e.g., **-nb=**, **-bl=**, **-bp=**, **-ba=** and **-cb=**) are specified in the call to the procedure.

Furthermore, the **-nb=**, **-bl=**, **-bp=**, **-ba=** and **-cb=** arguments allow the user to determine and adapt the exact form of the blockwise bootstrap algorithm depending on the analyzed dataset. This moving block bootstrap algorithm is formally similar to the one described in [comp_cor_3d](#) or [comp_cor_4d](#) for testing the significance of a correlation coefficient when **-a=bootstrap**. Refer to [comp_cor_3d](#) for further details on this moving block bootstrap algorithm. This approach is useful if you want to assess that the analyzed covariance matrix is above the “noise”.

Note, that [comp_svd_3d](#) performs exactly the same task as [comp_svd2_3d](#), but differs in the form of the moving block bootstrap approach used to assess the significance of the singular triplets of the SVD analysis. In [comp_svd_3d](#), the used moving block bootstrap approach allows to assess the sampling distributions and stability of the SVD results without destroying the covarying signals in the bootstrapped left and right fields. If you already know that some significant covarying signals are present in the analyzed covariance matrix, this bootstrap approach is much more appropriate and meaningful.

Two output NetCDF datasets containing the singular values, the left and right singular vectors of the covariance matrix, the corresponding left and right standardized SV time series and, the homogeneous and heterogeneous vectors for each field are created. The left and right singular vectors, and the homogeneous and heterogeneous vectors for each field, are repacked onto the original grids of the two input NetCDF variables in the output NetCDF datasets. In addition, if confidence levels for the SCF, NC and correlation coefficients are estimated, these probabilities are also included in the output NetCDF datasets.

This procedure is parallelized if OpenMP is used and will be also much faster if an optimized BLAS library is specified at compilation with the `_BLAS` CPP macro. Moreover, this procedure may use (randomized) partial SVD algorithms [Martinsson], which are highly efficient on huge covariance matrices if you are interested only in the few leading terms of the SVD of the covariance matrix between the **X** and **Y** fields.

2.71.4 Further Details

Usage

```
$ comp_svd2_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -a=type_of_analysis (optional : scp, cov, cor) \
  -n=number_of_sing_triplets (optional) \
  -g=grid_type (optional : n, t, u, v, w, f) \
  -r=resolution (optional : r2, r4) \
  -b=nlon_orca, nlat_orca (optional) \
  -x=lon1,lon2 (optional) \
  -y=lat1,lat2 (optional) \
  -t=time1,time2 (optional) \
  -c=input_climatology_netcdf_file (optional) \
  -d=distance (optional : dist2, ident) \
  -o=output_svd_netcdf_file_left_field (optional) \
  -f2=input_netcdf_file_right_field (optional) \
  -v2=netcdf_variable_right_field (optional) \
  -m2=input_mesh_mask_netcdf_file_right_field (optional) \
  -g2=grid_type_right_field (optional : n, t, u, v, w, f) \
  -r2=resolution_right_field (optional : r2, r4) \
  -b2=nlon_orca, nlat_orca (optional) \
  -x2=lon1_right_field,lon2_right_field (optional) \
  -y2=lat1_right_field,lat2_right_field (optional) \
  -z2=level1_right_field,level2_right_field (optional) \
  -t2=time1_right_field,time2_right_field (optional) \
  -c2=input_climatology_netcdf_file_right_field (optional) \
  -d2=distance_right_field (optional : dist2, dist3, ident) \
  -o2=output_svd_netcdf_file_right_field (optional) \
  -alg=algorithm (optional : svd, inviter, deflate, r
→rsvd) \
  -cb=bootstrap_statistic_significativity_type (optional : values, vector) \
  -nb=number_of_shuffles (optional) \
  -ba=bootstrap_algorithm (optional : svd, inviter, deflate, r
→rsvd) \
  -bp=bootstrap_periodicity (optional) \
  -bl=bootstrap_block_length (optional) \
  -mi=missing_value (optional) \
  -ortho (optional) \
  -double (optional) \
  -bigfile (optional) \
  -hdf5 (optional) \
  -cdf5 (optional) \
  -tlimited (optional)
```

By default

-a= the *type_of_analysis* is set to *scp*. This means that the singular vectors and singular values are computed from the sums of squares and cross-products matrix between the left and right fields

-n= *number_of_sing_triplets* is set to 4. This means that the first 4 singular triplets of the covariance matrix between the left and right fields are computed and stored in the output NetCDF files *output_svd_netcdf_file_left_field* and *output_svd_netcdf_file_right_field*

- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the left field extracted from the input NetCDF variable *input_netcdf_file* is assumed to be regular or Gaussian
- r=** if the input NetCDF variable *netcdf_variable* is from the NEMO model (e.g., if **-g=** argument is not set to *n*) the resolution is assumed to be *r2*
- b=** if **-g=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, of the left input NetCDF variable *netcdf_variable* are determined from the **-r=** argument. However, you may override this choice by default with the **-b=** argument
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- c=** this argument is not used. This argument is required only if the *type_of_analysis* is set to *cov* or *cor* and is used to specify the climatology NetCDF file for computing anomalies or standardized anomalies for the left field
- d=** the *distance* is set to *dist2*. This means that distances and scalar products for the left field in the SVD analysis are computed with the diagonal metric associated with the 2-D grid-mesh associated with the input NetCDF variable *netcdf_variable*
- o=** the *output_svd_netcdf_file_left_field* is named *svd_left_netcdf_variable.nc*
- f2=** this argument is not used. It is required only if the right field is not stored in the same file as the left NetCDF variable
- v2=** this argument is not used. It is required only if the right field is not extracted from the same input NetCDF variable than the left field
- m2=** this argument will take the same value as the **-m=** argument. It is required only if the right field is not extracted from the same input NetCDF variable as the left field
- g2=** same as the **-g=** argument if the **-v2=** argument is omitted and **-g2=n** otherwise
- r2=** same as the **-r=** argument if **-v2=** is omitted and **-r2=r2** otherwise
- b2=** if **-g2=** is not set to *n*, the dimensions of the 2-D grid-mesh, *nlon_orca* and *nlat_orca*, of the right field extracted from the input NetCDF variable *netcdf_variable_right_field* are determined from the **-r2=** argument. However, you may override this choice by default with the **-b2=** argument
- x2=** the whole longitude domain associated with the *netcdf_variable_right_field*
- y2=** the whole latitude domain associated with the *netcdf_variable_right_field*
- z2=** the whole level associated with the *netcdf_variable_right_field*
- t2=** the whole time period associated with the *netcdf_variable_right_field*
- c2=** this argument is not used. This argument is required only if the *type_of_analysis* is set to *cov* or *cor* and is used to specify the climatology NetCDF file for computing anomalies or standardized anomalies for the right field if this field is not extracted from the same input NetCDF variable as the left field
- d2=** same as **-d=** if **-v2=** is omitted, **-d2=dist2** if the *netcdf_variable_right_field* is a 3D variable and **-d2=dist3** if the *netcdf_variable_right_field* is a 4D variable
- o2=** the *output_svd_netcdf_file_right_field* is named *svd_right_netcdf_variable_right_field.nc*
- alg=** the *algorithm* option is set to *inviter*. This means that the SVD analysis is computed by a partial SVD of the covariance matrix using an inverse iteration algorithm

- cb=** *bootstrap_statistic_significativity_type* is set to *values*. This means that only the SCF and NC coefficients are tested by the moving block bootstrap algorithm. This saves computing time because this requires only the computation of singular values in the moving block bootstrap algorithm
- nb=** *number_of_shuffles* is set to 99. This means that 99 bootstrap samples are generated in the moving block bootstrap algorithm for testing the significance of the singular triplets
- ba=** the *bootstrap_algorithm* option is set to *rsvd*. This means that the (partial) SVDs of the bootstrap covariance matrices are computed by a randomized SVD algorithm during the bootstrap phase of the procedure for efficiency reasons
- bp=** the time series are assumed to be stationary and *bootstrap_periodicity* is set to 1 in the moving block bootstrap procedure for testing the significance of the singular triplets. This means that the blocks in the bootstrap algorithm are not forced to begin at specific observations. Use this parameter if the time series are cyclostationary, see the remarks below for further details
- bl=** *bootstrap_block_length* is set to *bootstrap_periodicity*.2
- mi=** the *missing_value* attribute in the output NetCDF files is set to 1.e+20
- ortho** the computed singular vectors and associated SV time series are not reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods, e.g., if **-alg=inviter** or **-alg=deflate**
- double** the results of the SVD analysis are stored as single-precision floating point numbers in the output NetCDF files. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** NetCDF classical format files are created. If **-bigfile** is activated, the output NetCDF files are 64-bit offset format files
- hdf5** NetCDF classical format files are created. If **-hdf5** is activated, the output NetCDF files are NetCDF-4/HDF5 format files
- cdf5** NetCDF classical format files are created. If **-cdf5** is activated, the output NetCDF files are CDF5 format files
- tlimited** the time dimension is defined as unlimited in the output NetCDF files. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF files

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the left field for the SVD analysis must be extracted and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file*.
- 2) The **-v2=netcdf_variable_right_field** argument specifies the NetCDF variable from which the right field of the SVD analysis must be extracted and the **-f2=input_netcdf_file_right_field** argument specifies that this NetCDF variable must be extracted from the NetCDF file, *input_netcdf_file_right_field*.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the geographical domain used for defining the left field in the SVD analysis is determined from the attributes of the input mesh mask NetCDF variable named *grid_ttypemask* (e.g., *lon1_Eastern_limit*, *lon2_Western_limit*, *lat1_Southern_limit* and *lat2_Northern_limit*) which is read from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing and the **-x=** and **-y=** arguments are also not specified, the whole geographical domain associated with the *netcdf_variable* is used for defining the left field in the SVD analysis.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the

longitude domain is from $nlon+lon1+1$ to $lon2$ where $nlon$ is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

This remark applies also for the **-x2=**, **-y2=** and **-z2=** arguments used for defining the right field in the SVD analysis.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using `comp_svd2_3d`.

- 4) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to estimate the covariance matrix between the left and right fields.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the SV time series in the output NetCDF files will have $n_{time} = time2 - time1 + 1$ time observations.

This remark applies also for **-t2=** argument used to define the time dimension of the right field.

- 5) If **-g=** is set to `t`, `u`, `v`, `w` or `f` it is assumed that the NetCDF variable *netcdf_variable* is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions). In this case, the duplicate points from the ORCA grid are removed when extracting the left field of the SVD analysis, as far as possible, and, in particular, if the 2-D grid-mesh of the input NetCDF variable covers the whole globe. On output, the duplicate points are restored when writing the SVD results (e.g., the singular, homogeneous and heterogeneous vectors), if the geographical domain of the input NetCDF variable *netcdf_variable* is the whole globe.

If **-g=** is set to `n`, it is assumed that the 2-D grid-mesh associated with *netcdf_variable* is regular or Gaussian and as such has no duplicate points.

The **-g=** argument is also used to determine the name of the NetCDF variables which contain the 2-D mesh-mask and the scale factors in the *input_mesh_mask_netcdf_file* (e.g., these variables are named *grid_typedmask*, *e1grid_type* and *e2grid_type*, respectively). This *input_mesh_mask_netcdf_file* may be created by [comp_clim_3d](#) if the 2-D grid-mesh is regular or gaussian.

This remark applies also for **-g2=** argument used to define the grid type of the right field.

- 6) If **-g=** is set to `t`, `u`, `v`, `w` or `f` (e.g., if the NetCDF variable is from an experiment with the NEMO model), the **-r=** argument gives the resolution used. If:

- **-r=r2** the NetCDF variable is from an experiment with the ORCA R2 configuration
- **-r=r4** the NetCDF variable is from an experiment with the ORCA R4 configuration.

This remark applies also for **-g2=** argument used to define the grid type of the right field.

- 7) If the NetCDF variable *netcdf_variable* is from an experiment with the NEMO ocean model, but the resolution is not `r2` or `r4`, the dimensions of the ORCA grid must be specified explicitly with the **-b=** argument.

This remark applies also for **-b2=** argument used to define the grid type of the right field.

- 8) The **-a=** argument specifies if the left and right fields are centered or standardized with an input climatology (specified with the **-c=** and **-c2=** arguments) before computing the covariance matrix between the two fields. If:

- **-a=scp**, the SVD analysis is done on the raw data of the two fields
- **-a=cov**, the SVD analysis is done on the anomalies of the two fields
- **-a=cor**, the SVD analysis is done on the standardized anomalies of the two fields

By default, the **-a=** argument is set to `scp`.

- 9) The *input_climatology_netcdf_file* and *input_climatology_netcdf_file_right_field* specified, respectively, with the **-c=** and **-c2=** arguments are needed only if **-a=cov** or **-a=cor**.
- 10) If **-a=cov** or **-a=cor**, the selected time periods for the left and right fields specified, respectively, with the **-t=** and **-t2=** arguments, must agree with the periods used to estimate the climatologies.

This means that the first selected time observation (*time1* if the **-t=** argument is present) must correspond to the first day, month, season of the climatology specified with the **-c=** argument for the left field. This remark also applies for the right field and the **-t2=** and **-c2=** arguments.

- 11) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*), the mask (in the *input_mesh_mask_netcdf_file*), the scale factors (in the *input_mesh_mask_netcdf_file*), and the climatology (in the *input_climatology_netcdf_file*) must agree.

Similarly, for the right field, the geographical shapes of the *netcdf_variable_right_field* (in the *input_netcdf_file_right_field*), the mask (in the *input_mesh_mask_netcdf_file_right_field*), the scale factors (in the *input_mesh_mask_netcdf_file_right_field*), and the climatology (in the *input_climatology_netcdf_file_right_field*) must agree.

- 12) The **-n=number_of_sing_triplets** argument specifies the number of singular triplets of the SVD of the covariance matrix between the left and right fields, which must be stored (and also computed if **-alg=inviter**, **-alg=deflate** or **-alg=rsvd** is specified) in the output NetCDF files given in the **-o=** and **-o2=** arguments. The default value is 4.
- 13) The **-d=distance** argument specifies the metric and scalar product used for the left field in the SVD analysis. If:
- **-d=dist2**, the SVD analysis is done with the diagonal distance associated with the horizontal 2-D grid-mesh of the left field (e.g., each grid point is weighted accordingly to the surface associated with it)
 - **-d=ident**, the SVD analysis is done with the identity metric : the Euclidean distance and the usual scalar product is used in the SVD analysis.

By default, the **-d=** argument is set to **dist2**.

This remark applies also for the **-d2=** argument used to define the distance and scalar product of the right field in the SVD analysis. If the second NetCDF variable *netcdf_variable_right_field*, used to define the right field in the SVD analysis, has 4 dimensions, the following value is also allowed for the **-d2=** argument:

- **-d=dist3** meaning that the SVD analysis is done with the diagonal distance associated with the whole 3D grid of the right field (e.g., each grid point is weighted accordingly to the volume or weight associated with it).

By default, the **-d2=** argument is set to the same value as the **-d=** argument if **-v2=** is omitted, and, otherwise, **-d2=dist2** if the *netcdf_variable_right_field* is a 3D variable and **-d2=dist3** if the *netcdf_variable_right_field* is a 4D variable.

- 14) The **-alg=** argument determines how the singular values and vectors of the covariance matrix between the left and right fields are computed. If:
- **-alg=svd**, a full SVD of the rectangular covariance matrix is computed.
 - **-alg=inviter**, a partial SVD of the rectangular covariance matrix is computed by inverse iteration.
 - **-alg=deflate**, a partial SVD of the rectangular covariance matrix is computed by a deflation technique.
 - **-alg=rsvd**, a partial SVD of the rectangular covariance matrix is computed by a very fast randomized algorithm.

All algorithms are parallelized if OpenMP is used. The default is **-alg=inviter** since computing a partial SVD is generally much faster than computing a full SVD. But, **-alg=deflate** is generally as fast as **-alg=inviter**. **-alg=rsvd** is generally much faster than all other options, but the computed singular vectors and SV variables may be less accurate depending on the shape of the distribution of the singular values of the covariance matrix.

- 15) If the **-ortho** argument is used, the computed singular vectors and associated SV time series are always reorthogonalized if a (partial) SVD is computed by the deflation or inverse iteration methods. By default, they are only partially orthogonalized if the computed singular values are not well separated. Note that this argument has no effect if **-alg=svd** or **-alg=rsvd**.

- 16) If any of the **-nb=**, **-bl=**, **-bp=** and **-cb=** arguments is specified, a moving block bootstrap algorithm is used to test the significance of the SCF, NC and, eventually, correlation coefficients associated with each singular triplet of the SVD analysis.
- 17) The **-cb=bootstrap_statistic_significativity_type** argument specifies which statistics are bootstrapped. If **-cb=values** (this is the default), the Square Covariance Fraction (SCF) and the Normalized root-mean-square Covariance (NC) coefficients are tested with the moving block bootstrap algorithm. If **-cb=vector**, the correlations between the Singular Variable time series of the left and right fields are also tested in addition of the SCF and NC coefficients. This may require much more computer time since singular vectors of the bootstrap versions of the covariance matrix are needed to estimate the Singular Variable time series. By default, bootstrap tests of these coefficients are not performed. These bootstrap tests are computed only if any of the optional bootstrap arguments (e.g., **-nb=**, **-bl=**, **-bp=**, **-ba=** and **-cb=**) are specified in the call to the procedure.
- 18) The **-nb=number_of_shuffles** argument specifies the number of shuffles for the bootstrap tests of the SCF, NC and correlation coefficients (by default 99).
- 19) The **-ba=** argument determines how the singular values and vectors of the bootstrap covariance matrices are computed in the bootstrap phase of the program. If:
 - **-ba=svd**, a full SVD of the bootstrap covariance matrix is computed.
 - **-ba=inviter**, a partial SVD of the bootstrap covariance matrix is computed by inverse iteration.
 - **-ba=deflate**, a partial SVD of the bootstrap covariance matrix is computed by a deflation technique.
 - **-ba=rsvd**, a partial SVD of the bootstrap covariance matrix is computed by a very fast randomized algorithm.

All algorithms are parallelized if OpenMP is used. The default is **-ba=rsvd** for efficiency reasons, as a randomized SVD algorithm is generally much faster than all other options, but the computed singular vectors and SV variables may be less accurate depending on the shape of the distribution of the singular values of the bootstrap covariance matrix.

- 20) The **-bp=bootstrap_periodicity** argument specifies that the index, i , of the first observation of each selected block in the moving block bootstrap algorithm verifies the condition $i = 1 + \text{bootstrap_periodicity} \cdot j$ where j is a random positive integer. *bootstrap_periodicity* must be greater than zero and less than the length of the time series. This parameter is useful if the time series are cyclostationary instead of stationary. By default, *bootstrap_periodicity* is set to 1.
- 21) The **-bl=bootstrap_block_length** argument specifies the size of the blocks in the moving block bootstrap algorithm. *bootstrap_block_length* must be greater or equal to the *bootstrap_periodicity* and less than the length of the time series. If **-a=cov** or **-a=cor** is specified, it is highly recommended to set *bootstrap_block_length* as a multiple of the periodicity in the data as this will take properly into account the cyclostationary of the analyzed time series.

By default, *bootstrap_block_length* is set to *bootstrap_periodicity*.2. If you perform a MCA on huge datasets and you require bootstrap testing of the MCA results, the elapsed time will be largely reduced if you use a large *bootstrap_block_length*, especially if the NCSTAT software has been built with the `_BLAS` CPP macro (e.g., if BLAS support has been activated). In such cases, *bootstrap_block_length* values between 24 and 122 (depending on the computer) may speed significantly the bootstrap computations.

- 22) The **-mi=missing_value** argument specifies the missing value indicator associated with the NetCDF variables in the *output_netcdf_file* and *output_netcdf_file_right_field*.

If the **-mi=** argument is not specified *missing_value* is set to $1.e+20$.

- 23) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF files.

By default, the results are stored as single-precision floating point numbers in the output NetCDF files.

- 24) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the output netcdf files will be 64-bit offset format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 25) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the output netcdf files will be NetCDF-4/HDF5 format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 26) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the output netcdf files will be CDF5 format files instead of NetCDF classic format files. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 27) It is assumed that the data has no missing values.

- 28) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 29) For more details on SVD (e.g., MCA) analysis in the climate literature and randomized SVD algorithms, see

- “A manual for EOF and SVD analyses of climate data”, by Bjornsson, H., and Venegas, S.A., McGill University, CCGCR Report No. 97-1, Montréal, Québec, 52pp, 1997. <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- “An intercomparison of methods for finding coupled patterns in climate data”, by Bretherton, Smith, C., and Wallace, J.M., Journal of Climate, Vol. 5, 541-560 pp, 1992. doi: [10.1175/1520-0442\(1992\)005<0541:AIOMFF>2.0.CO;2](https://doi.org/10.1175/1520-0442(1992)005<0541:AIOMFF>2.0.CO;2)
- “Seasonality of large scale atmosphere-ocean interaction over the North Pacific”, by Zhang, Y., Norris, J.R., and Wallace, J.M., Journal of Climate, Vol. 11, 2473-2481 pp, 1998. doi: [10.1175/1520-0442\(1998\)011<2473:SOLSAO>2.0.CO;2](https://doi.org/10.1175/1520-0442(1998)011<2473:SOLSAO>2.0.CO;2)
- “Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions”, By Halko, N., Martinsson, P.G., and Tropp, J.A., SIAM Rev., 53, 217-288, 2011. <https://epubs.siam.org/doi/abs/10.1137/090771806>
- “Algorithm 971: An implementation of a randomized algorithm for principal component analysis”, by Li, H., Linderman, G.C., Szlam, A., Stanton, K.P., Kluger, Y., and Tygert, M., ACM Trans. Math. Softw. 43, 3, Article 28 (January 2017). <https://pubmed.ncbi.nlm.nih.gov/28983138>
- “Randomized methods for matrix computations”, by Martinsson, P.G., arXiv.1607.01649, 2019. <https://arxiv.org/abs/1607.01649>

Outputs

`comp_svd2_3d` creates two output NetCDF files. The first output file (specified in the `-o=output_svd_netcdf_file_left_field` argument) contains all the SVD statistics associated with the left

field (specified in the `-v=netcdf_variable` argument) and the second output file (specified in the `-o2=output_svd_netcdf_file_right_field` argument) the SVD statistics associated with the right field (specified in the `-v2=netcdf_variable_right_field` argument).

The `output_svd_netcdf_file_left_field` contains the singular values and left singular vectors of the covariance matrix, the left homogeneous and heterogeneous vectors, and the left SV time series of the SVD analysis. The number of SV time series, regression vectors, singular vectors and singular values stored in the output NetCDF dataset is determined by the `-n=number_of_sing_triplets` argument. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the first input NetCDF variable `netcdf_variable`):

- 1) `netcdf_variable_svd(number_of_sing_triplets,nlat,nlon)` : the `number_of_sing_triplets` leading left singular vectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix between the left and right fields. The singular vectors are sorted by descending order of the associated singular values.

The left singular vectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values. If this is a problem, you can use `comp_norm_3d` for restricting the geographical domain in the input dataset before using `comp_svd2_3d`.

- 2) `netcdf_variable_hom(number_of_sing_triplets,nlat,nlon)` : the selected left homogeneous vectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix. The left homogeneous vectors are sorted by descending order of the associated singular values. These vectors are scaled such that they give the scalar products (`-a=scp`), covariances (`-a=cov`) or correlations (`-a=cor`) between the original observed time series in the left field and the left SV time series.

The left homogeneous vectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

- 3) `netcdf_variable_het(number_of_sing_triplets,nlat,nlon)` : the selected left heterogeneous vectors of the sums of squares and cross-products (`-a=scp`), covariance (`-a=cov`) or correlation (`-a=cor`) matrix. The left heterogeneous vectors are sorted by descending order of the associated singular values. These vectors are scaled such that they give the scalar products (`-a=scp`), covariances (`-a=cov`) or correlations (`-a=cor`) between the original observed time series in the left field and the right SV time series stored in the other output NetCDF file.

The left heterogeneous vectors are packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

- 4) `netcdf_variable_sv(ntime,number_of_sing_triplets)` : the left SV time series sorted by descending order of the singular values.

The SV time series are always standardized to unit variance.

- 5) `netcdf_variable_SV_STDs(number_of_sing_triplets)` : the standard-deviations of the left SV time series sorted by descending order of the singular values.
- 6) `netcdf_variable_SV_EXPVARs(number_of_sing_triplets)` : the proportion of variance of the left field explained by the left SV time series sorted by descending order of the singular values.

- 7) *netcdf_variable_CORSV*(*number_of_sing_triplets*) : the symmetric correlation matrix between the left SV time series, only the upper triangle of this symmetric matrix is written in the output file.
- 8) *netcdf_variable_RAW_VAR*(1) : the raw variance (or inertia if **-a=scp**) of the left field averaged over the selected domain.
- 9) *Sing_vals*(*number_of_sing_triplets*) : the singular values of the sums of squares and cross-products (if **-a=scp**) or covariance (if **-a=cov**) or correlation (if **-a=cor**) matrix between the left and right fields sorted in decreasing order.
- 10) *SCFs*(*number_of_sing_triplets*) : the Squared Covariance Fractions (SCF) described by each of the leading singular triplets of the squares and cross-products (if **-a=scp**) or covariance (if **-a=cov**) or correlation (if **-a=cor**) matrix between the left and right fields.
- 11) *NCs*(*number_of_sing_triplets*) : the Normalized root-mean-square Covariance (NC) coefficients associated with each of the leading singular triplets of the squares and cross-products (if **-a=scp**) or covariance (if **-a=cov**) or correlation (if **-a=cor**) matrix between the left and right fields.
- 12) *Corrs*(*number_of_sing_triplets*) : the correlation coefficients between the corresponding left and right SV time series.
- 13) *SCF_stat_sign*(*number_of_sing_triplets*) : the critical probabilities associated with the Squared Covariance Fractions (SCF) coefficients associated with each of the leading singular triplets estimated by a moving block bootstrap procedure. Large values indicate significant SCF coefficients. This NetCDF variable is computed and stored only if one of the **-nb=**, **-bl=**, **-bp=** or **-cb=** arguments is specified when calling the procedure.
- 14) *NC_stat_sign*(*number_of_sing_triplets*) : the critical probabilities associated with the Normalized root-mean-square Covariance (NC) coefficients associated with each of the leading singular triplets estimated by a moving block bootstrap procedure. Large values indicate significant NC coefficients. This NetCDF variable is computed and stored only if one of the **-nb=**, **-bl=**, **-bp=** or **-cb=** arguments is specified when calling the procedure.
- 15) *Corr_stat_sign*(*number_of_sing_triplets*) : the critical probabilities associated with the correlation coefficients between the corresponding left and right SV time series estimated by a moving block bootstrap procedure. Large values indicate significant correlation coefficients between the corresponding left and right SV time series. This NetCDF variable is computed and stored only if **-cb=vector** is specified when calling the procedure.

The *output_svd_netcdf_file_right_field* contains the singular values and right singular vectors of the covariance matrix, the right homogeneous and heterogeneous vectors, and the right SV time series of the SVD analysis. The number of SV time series, regression vectors and singular vectors and singular values stored in this output NetCDF dataset is also determined by the **-n=number_of_sing_triplets** argument. This output NetCDF dataset contains exactly the same NetCDF variables than the first NetCDF output file, but for the statistics of the right field instead of the left field. Refer to the description above for the content and definition of the NetCDF variables in the file *output_svd_netcdf_file_right_field*.

Examples

- 1) For computing an SVD analysis from two NetCDF variables *sst* and *slp* stored, respectively, in the NetCDF files *HadISST1_1m_1979_2005_sst.nc* and *hadslp_1m_1979_2005_slp.nc* use the following command (note that the analysis is done on the anomalies after removing the annual cycle for each variable since **-a=cov** is specified) :

```
$ comp_svd2_3d \
-a=cov \
-n=5 \
-f=HadISST1_1m_1979_2005_sst.nc \
-v=sst \
-c=clim_HadISST1_1m_1979_2005_sst.nc \
-x=111,330 \
-y=101,140 \
-m=mask_HadISST1_sst.nc \
-o=svd_HadISST1_1m_1979_2005_sst_oiat1.nc \
-f2=hads1p_1m_1979_2005_slp.nc \
-v2=slp \
-c2=clim_hads1p_1m_1979_2005_slp.nc \
-x2=-14,31 \
-y2=21,33 \
-m=mask_hads1p_slp.nc \
-o2=svd_hads1p_1m_1979_2005_slp_oiat1.nc
```

2.72 comp_symlin_filter_1d

2.72.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.72.2 Latest revision

30/04/2026

2.72.3 Purpose

Filter a real time series in a selected frequency band by linear symmetric filtering (e.g., Lanczos, Hamming, Hanning or “ideal” window filtering in this version of `comp_symlin_filter_1d`) [Bloomfield].

The time series is extracted from a uni or bidimensional variable readed from a NetCDF dataset and can be also detrended before linear symmetric filtering at the user option.

The number of coefficients used to build the linear symmetric filter can be selected and the filter can be applied to the time series in the time or frequency domain, also at the user option. This gives to the user some control on the desired end-effects of the filter (e.g., applying the filter in the frequency domain assumes implicitly that the time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series; on the other hand, applying the filter in the time domain implies some loss of data or some distortions of the desired response function of the filter at both ends of the filtered time series).

Additionally, the filtering can be done separately for different segments of equal length of the selected time series if this time series is not continuous in time.

The frequency response function (e.g., the transfer function) of the selected linear symmetric filter can be computed by `comp_freq_func_1d`. See the references cited below for more details on linear symmetric filtering.

This procedure returns the filtered real time series in a NetCDF dataset. If the NetCDF variable is tri- or fourdimensional use `comp_symlin_filter_3d` or `comp_symlin_filter_4d`, respectively, instead of `comp_symlin_filter_1d`. If the time series has a seasonal (or diurnal) cycle, use `comp_stl_1d` in order to estimate and remove the harmonic components of the time series before using `comp_symlin_filter_1d`.

2.72.4 Further Details

Usage

```
$ comp_symlin_filter_1d \
-f=input_netcdf_file \
-v=netcdf_variable \
-t=time1,time2 (optional) \
-o=output_netcdf_file (optional) \
-ni=index_for_2d_netcdf_variable (optional) \
-p=periodicity (optional) \
-pl=minimum_period (optional) \
-ph=maximum_period (optional) \
-tr=trend_removal (optional : 0, 1, 2, 3, -1, -2, -3) \
-nfc=number_of_filter_coefficients (optional) \
-mi=missing_value (optional) \
-hamming (optional) \
-win=window_choice (optional : 0.5 > 1.) \
-notestf (optional) \
-usefft (optional) \
-hdf5 (optional) \
-cdf5 (optional) \
-tlimited (optional)
```

By default

- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- ni=** if the *netcdf_variable* is bidimensional, the first time series is used
- p=** the *periodicity* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** the *trend_removal* is set to 0, which means that no detrending is done before filtering
- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter
- mi=** the *missing_value* for the output variable is set to $1.e+20$
- hamming** a Lanczos window filter is used. If the **-hamming** argument is specified a Hamming window filter is used instead
- win=** a Hamming window (e.g., **-win=0.54**) is convolved with the filter response by default if the **-hamming** argument is used, meaning that a Hamming window filter is computed. If the **-hamming** argument is not used, this argument has no effect
- notestf** normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where PH and PL are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the $[0., 0.5]$ frequency interval. By using the **-notestf** argument you can get ride of this limitation

- usefft** the linear symmetric filter is applied in the time domain. When you specify the **-usefft** argument the filter will be applied in the frequency domain, using an FFT algorithm and multiplication, instead of a convolution in the time domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a linear filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-t=time1,time2** argument is missing the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.
- 3) The **-ni=index_for_2d_netcdf_variable** argument specifies the index for selecting the time series if the *netcdf_variable* is a 2D NetCDF variable. By default, the first time series is used, which is equivalent to set *index_for_2d_netcdf_variable* to 1.
- 4) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *periodicity* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *periodicity*.
- 5) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.
- 6) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.
- 7) Setting **-pl=** and **-ph=** to the same value *P* is allowed only if the **-hamming** argument is not present (e.g., if Lanczos window filtering is used). In this case, an *-ideal-* band-pass filter with peak response near one at the single period *P* is computed and applied to the time series.
- 8) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.
- 9) The **-tr=** argument specifies pre- and post-filtering processing of the time series. If:

- **-tr=+/-1**, the mean of the time series is removed before time filtering
- **-tr=+/-2**, the drift from the time series is removed before time filtering. The drift for the time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
- **-tr=+/-3**, the least-squares line from the time series is removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the mean, drift or least-squares line are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 10) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients for the filtering of the time series. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to K , the first and last $(K-1)/2$ time observations in the output NetCDF file will be affected by end effects. Thus, the user must choose the number of filter terms, K , as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operation (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 11) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 12) If the **-hamming** argument is specified, a Hamming window filter is used instead of Lanczos window filter.

- 13) The **-win=** argument controls the form of the window which will be convolved with the filter if a Hamming window filter is requested with the **-hamming** argument. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., an “ideal” filter).

This argument has an effect only if the **-hamming** argument is also specified. The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1.

- 14) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $[0., 0.5]$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 15) The **-usefft** argument specifies that the linear symmetric filter must be applied in the frequency domain by using a Fast Fourier Transform and the convolution theorem. Moreover, if the **-usefft** argument is specified, the values at the ends of the output filtered series are computed implicitly by assuming that the input series is part of a periodic sequence.

- 16) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 17) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 18) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 19) If the time series has a seasonal or diurnal cycle, use *comp_stl_1d* to remove the pure harmonic components from the time series before filtering.

- 20) It is assumed that the data has no missing values.

- 21) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 22) For more details on linear symmetric filtering and examples of use in the climate literature, see

- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Lanczos filtering in one and two dimensions”, by Duchon, C., Journal of applied meteorology, vol. 18, 1016-1022, 1979. doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

`comp_symlin_filter_1d` creates an output NetCDF file that contains the filtered time series estimated from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, `ntime` is the selected number of time observations) :

- 1) *netcdf_variable_filt(ntime)* : the filtered time series for the raw time series associated with the input NetCDF variable.

Examples

- 1) For Hamming window filtering a real monthly time series between 18 and 30 months (e.g., biennial time scale) from a NetCDF variable called `sst` extracted from the file `sst.monthly.nino34.nc`, which includes a monthly time series, and store the results in the NetCDF file `qbo_sst_nino34.nc`, use the following command :

```
$ comp_symlin_filter_1d \
-f=sst.monthly.nino34.nc \
-v=sst \
-pl=18 \
-ph=30 \
```

(continues on next page)

(continued from previous page)

```
-hamming \
-o=qbo_sst_nino34.nc
```

2.73 comp_symlin_filter_3d

2.73.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.73.2 Latest revision

30/04/2026

2.73.3 Purpose

Filter a real multi-channel time series in a selected frequency band by linear symmetric filtering (e.g., Lanczos, Hamming, Hanning or “ideal” window filtering in this version of `comp_symlin_filter_3d`) [Bloomfield].

The multi-channel time series is extracted from a tridimensional variable readed from a NetCDF dataset and can also be detrended before linear symmetric filtering at the user option.

The number of coefficients used to build the linear symmetric filter can be selected and the filter can be applied to the multi-channel time series in the time or frequency domain, also at the user option. This gives to the user some control on the desired end-effects of the filter (e.g., applying the filter in the frequency domain assumes implicitly that the multi-channel time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series; on the other hand, applying the filter in the time domain implies some loss of data or some distortions of the desired response function of the filter at both ends of the filtered time series).

Additionally, the filtering can be done separately for different segments of equal length of the selected multi-channel time series if this time series is not continuous in time.

The frequency response function (e.g., the transfer function) of the selected linear symmetric filter can be computed by `comp_freq_func_1d`. See the references cited below for more details on linear symmetric filtering.

This procedure returns the filtered real multi-channel time series in a NetCDF dataset. If the NetCDF variable is uni- or fourdimensional use `comp_symlin_filter_1d` or `comp_symlin_filter_4d`, respectively, instead of `comp_symlin_filter_3d`. If the multi-channel time series has a seasonal (or diurnal) cycle, use `comp_clim_3d` and `comp_norm_3d` or `comp_stl_3d` in order to estimate and remove the harmonic components of the time series before using `comp_symlin_filter_3d`.

This procedure is parallelized if OpenMP is used.

2.73.4 Further Details

Usage

```
$ comp_symlin_filter_3d \
  -f=input_netcdf_file \
  -v=netcdf_variable \
  -m=input_mesh_mask_netcdf_file          (optional) \
```

(continues on next page)

(continued from previous page)

```

-g=grid_type                (optional : n, t, u, v, w, f) \
-x=lon1,lon2                (optional) \
-y=lat1,lat2                (optional) \
-t=time1,time2              (optional) \
-o=output_netcdf_file       (optional) \
-p=periodicity              (optional) \
-pl=minimum_period          (optional) \
-ph=maximum_period          (optional) \
-tr=trend_removal           (optional : 0, 1, 2, 3, -1, -2, -3) \
-nfc=number_of_filter_coefficients (optional) \
-mi=missing_value           (optional) \
-ngp=number_of_grid_points  (optional) \
-hamming                    (optional) \
-win=window_choice          (optional : 0.5 > 1.) \
-notestf                    (optional) \
-usefft                     (optional) \
-double                     (optional) \
-bigfile                    (optional) \
-hdf5                       (optional) \
-cdf5                       (optional) \
-tlimited                    (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- p=** the *periodicity* is set to *time2 - time1 + 1*, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** the *trend_removal* is set to 0, which means that no detrending is done before filtering
- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter
- mi=** the *missing_value* for the output variable is equal to $1.e+20$
- ngp=** the *number_of_grid_points* is set to the number of grid points in the selected domain
- hamming** a Lanczos window filter is used. If the **-hamming** argument is specified a Hamming window filter is used instead
- win=** a Hamming window (e.g., **-win=0.54**) is convolved with the filter response by default if the **-hamming** argument is used, meaning that a Hamming window filter is computed. If the **-hamming** argument is not used, this argument has no effect

- notestf** normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where PH and PL are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the $[0., 0.5]$ frequency interval. By using the **-notestf** argument you can get ride of this limitation
- usefft** the linear symmetric filter is applied in the time domain. When you specify the **-usefft** argument the filter will be applied in the frequency domain, using an FFT algorithm and multiplication, instead of a convolution in the time domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a linear filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF *mesh_mask* variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using [comp_symlin_filter_3d](#).

- 5) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 6) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *periodicity* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *periodicity*.

- 7) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 8) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the multi-channel time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed only if the **-hamming** argument is not present (e.g., if Lanczos window filtering is used). In this case, an *-ideal-* band-pass filter with peak response near one at the single period *P* is computed and applied to the multi-channel time series.
- 10) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.
- 11) The **-tr=** argument specifies pre- and post-filtering processing of the multi-channel time series. If:

- **-tr=+/-1**, the means of the time series are removed before time filtering
- **-tr=+/-2**, the drifts from the time series are removed before time filtering. The drift for each time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
- **-tr=+/-3**, the least-squares lines from the multi-channel time series are removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the means, drifts or least-squares lines are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 12) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients for the filtering of the multi-channel time series. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to *K*, the first and last $(K-1)/2$ time observations in the output NetCDF file will be affected by end effects. Thus, the user must choose the number of filter terms, *K*, as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operation (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 13) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 14) The **-ngp=** argument can be used if you have memory problems when running *comp_symlin_filter_3d* on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In

case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.

- 15) If the **-hamming** argument is specified, a Hamming window filter is used instead of Lanczos window filter.
- 16) The **-win=** argument controls the form of the window which will be convolved with the filter if a Hamming window filter is requested with the **-hamming** argument. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., an “ideal” filter).

This argument has an effect only if the **-hamming** argument is also specified. The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1.

- 17) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $[0., 0.5]$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 18) The **-usefft** argument specifies that the linear symmetric filter must be applied in the frequency domain by using a Fast Fourier Transform and the convolution theorem. Moreover, if the **-usefft** argument is specified, the values at the ends of the output filtered series are computed implicitly by assuming that the input series is part of a periodic sequence.

- 19) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 20) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 23) If the multi-channel time series has a seasonal or diurnal cycle, use *comp_stl_3d* or *comp_clim_3d* to remove the pure harmonic components from the time series before filtering.

- 24) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 25) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 26) For more details on linear symmetric filtering and examples of use in the climate literature, see
 - “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
 - “Lanczos filtering in one and two dimensions”, by Duchon, C., Journal of applied meteorology, vol. 18, 1016-1022, 1979. doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

`comp_symlin_filter_3d` creates an output NetCDF file that contains the filtered time series estimated from the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable and `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_filt(ntime, nlat, nlon)` : the filtered time series for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

The filtered multi-channel time series is packed in a tridimensional variable whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For Hamming window filtering a real multi-channel monthly time series between 18 and 30 months (e.g., biennial time scale) from a tridimensional NetCDF variable called `mslp` extracted from the file `mslp.monthly.mean_ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `tbo_mslp_ncep2.nc`, use the following command :

```
$ comp_symlin_filter_3d \
-f=mslp.monthly.mean_ncep2.nc \
-v=mslp \
-m=mesh_mask_mslp_ncep2.nc \
-pl=18 \
-ph=30 \
-hamming \
-o=tbo_mslp_ncep2.nc
```

2.74 comp_symlin_filter_4d

2.74.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.74.2 Latest revision

30/04/2026

2.74.3 Purpose

Filter a real multi-channel time series in a selected frequency band by linear symmetric filtering (e.g., Lanczos, Hamming, Hanning or “ideal” window filtering in this version of `comp_symlin_filter_4d`) [Bloomfield].

The multi-channel time series is extracted from a fourdimensional variable readed from a NetCDF dataset and can also be detrended before linear symmetric filtering at the user option.

The number of coefficients used to build the linear symmetric filter can be selected and the filter can be applied to the multi-channel time series in the time or frequency domain, also at the user option. This gives to the user some control on the desired end-effects of the filter (e.g., applying the filter in the frequency domain assumes implicitly that the multi-channel time series is part of a periodic infinite series whose period is exactly equal to the length of the analyzed time series; on the other hand, applying the filter in the time domain implies some loss of data or some distortions of the desired response function of the filter at both ends of the filtered time series).

Additionally, the filtering can be done separately for different segments of equal length of the selected multi-channel time series if this time series is not continuous in time.

The frequency response function (e.g., the transfer function) of the selected linear symmetric filter can be computed by `comp_freq_func_1d`. See the references cited below for more details on linear symmetric filtering.

This procedure returns the filtered real multi-channel time series in a NetCDF dataset. If the NetCDF variable is uni- or tridimensional use `comp_symlin_filter_1d` or `comp_symlin_filter_3d`, respectively, instead of `comp_symlin_filter_4d`. If the multi-channel time series has a seasonal (or diurnal) cycle, use `comp_clim_4d` and `comp_norm_4d` or `comp_stl_4d` in order to estimate and remove the harmonic components of the time series before using `comp_symlin_filter_4d`.

This procedure is parallelized if OpenMP is used.

2.74.4 Further Details

Usage

```
$ comp_symlin_filter_4d \
-f=input_netcdf_file \
-v=netcdf_variable \
-m=input_mesh_mask_netcdf_file           (optional) \
-g=grid_type                             (optional : n, t, u, v, w, f) \
-x=lon1,lon2                             (optional) \
-y=lat1,lat2                             (optional) \
-z=level1,level2                         (optional) \
-t=time1,time2                           (optional) \
-o=output_netcdf_file                    (optional) \
-p=periodicity                           (optional) \
-pl=minimum_period                       (optional) \
-ph=maximum_period                       (optional) \
-tr=trend_removal                        (optional : 0, 1, 2, 3, -1, -2, -3) \
-nfc=number_of_filter_coefficients       (optional) \
-mi=missing_value                        (optional) \
-ngp=number_of_grid_points                (optional) \
-hamming                                 (optional) \
```

(continues on next page)

(continued from previous page)

```

-win=window_choice          (optional : 0.5 > 1.) \
-notestf                    (optional) \
-usefft                     (optional) \
-double                     (optional) \
-bigfile                    (optional) \
-hdf5                      (optional) \
-cdf5                      (optional) \
-tlimited                    (optional)

```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 3-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- z=** the whole vertical resolution associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- o=** the *output_netcdf_file* is named *filt_netcdf_variable.nc*
- p=** the *periodicity* is set to $\text{time2} - \text{time1} + 1$, which means that the time series is considered as continuous with only one time segment
- pl=** the *minimum_period* is set to 0, which means that no filtering is done for the shorter periods
- ph=** the *maximum_period* is set to 0, which means that no filtering is done for the longer periods
- tr=0** the *trend_removal* is set to 0, which means that no detrending is done before filtering
- nfc=** the *number_of_filter_coefficients* is determined in order to optimize the frequency response function of the selected filter
- mi=** the *missing_value* for the output variable is equal to $1.e+20$
- ngp=** the *number_of_grid_points* is set to the number of grid points in the selected domain
- hamming** a Lanczos window filter is used. If the **-hamming** argument is specified a Hamming window filter is used instead
- win=** a Hamming window (e.g., **-win=0.54**) is convolved with the filter response by default if the **-hamming** argument is used, meaning that a Hamming window filter is computed. If the **-hamming** argument is not used, this argument has no effect
- notestf** normally, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies $1/PH$ and $1/PL$ (where *PH* and *PL* are the values of the **-ph=** and **-pl=** arguments, respectively) of the selected filter are inside the $[0., 0.5]$ frequency interval. By using the **-notestf** argument you can get ride of this limitation
- usefft** the linear symmetric filter is applied in the time domain. When you specify the **-usefft** argument the filter will be applied in the frequency domain, using an FFT algorithm and multiplication, instead of a convolution in the time domain
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers

- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable for which a linear filtering operation must be computed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The geographical and vertical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the NetCDF mesh_mask variable (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 3) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian.

This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument

- 4) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon + lon1 + 1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using *comp_symlin_filter_4d*.

- 5) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 6) If the **-p=** argument is specified, the filtering is applied separately for each time segment of length *periodicity* (as determined by the value of the **-p=** argument). The whole selected time period (e.g., $time2 - time1 + 1$) must also be a multiple of the *periodicity*.

- 7) The **-pl=** argument specifies the minimum period of oscillation of the filtered time series. The *minimum_period* is expressed in number of time observations.

Do not use the **-pl=** argument or use **-pl=0** for high-pass filtering frequencies corresponding to periods shorter than **-ph=PH**.

The **-pl=** argument is a positive integer equal to 0 or greater than 2.

- 8) The **-ph=** argument specifies the maximum period of oscillation of the filtered time series. The *maximum_period* is expressed in number of time observations. Do not use the **-ph=** argument or use **-ph=0** for low-pass filtering frequencies corresponding to periods longer than **-pl=PL**. For example, **-pl=6** (or 18) and **-ph=32** (or 96) select periods between 1.5 and 8 years for quarterly (monthly) time series.

The **-ph=** argument is a positive integer equal to 0 or greater than 2 and less than the length of the multi-channel time series or the *periodicity* if the **-p=** argument is used.

The **-ph=** argument must also be greater or equal to the **-pl=** argument if both are specified.

- 9) Setting **-pl=** and **-ph=** to the same value *P* is allowed only if the **-hamming** argument is not present (e.g., if Lanczos window filtering is used). In this case, an *-ideal-* band-pass filter with peak response near one at the single period *P* is computed and applied to the multi-channel time series.
- 10) Setting both **-pl=0** and **-ph=0** is also allowed. In that case, no frequencies filtering is done, but the data may be detrended if the **-tr=** argument is used with a value of 1, 2 or 3.
- 11) The **-tr=** argument specifies pre- and post-filtering processing of the multi-channel time series. If:
 - **-tr=+/-1**, the means of the time series are removed before time filtering
 - **-tr=+/-2**, the drifts from the time series are removed before time filtering. The drift for each time series is estimated using the formula: $\text{drift} = (\text{tseries}(\text{ntime}) - \text{tseries}(1)) / (\text{ntime} - 1)$
 - **-tr=+/-3**, the least-squares lines from the multi-channel time series are removed before time filtering.

If **-tr=-1**, **-2** or **-3**, the means, drifts or least-squares lines are reintroduced post-filtering, respectively.

For other values of the **-tr=** argument, nothing is done before or after filtering.

If the **-p=** argument is present, the pre-filtering and post-filtering processing is applied to each time segment, separately.

The **-tr=** argument must be an integer and the default value for the **-tr=** argument is 0.

- 12) The **-nfc=** argument specifies the desired number of symmetric linear filter coefficients for the filtering of the multi-channel time series. If **-nfc=** is not specified, an optimal value is chosen in order to obtain a good frequency response function for the selected filter.

However, if **-nfc=** is set to *K*, the first and last $(K-1)/2$ time observations in the output NetCDF file will be affected by end effects. Thus, the user must choose the number of filter terms, *K*, as a compromise between:

- 1) A sharp cutoff, that is, $1/K$ small; and
- 2) Minimizing the number of data points lost or affected by the filtering operation (since $(K-1)/2$ data points will be lost or affected from each end of the series).

Finally, the **-nfc=** argument must be greater or equal to 3 and odd.

- 13) The **-mi=missing_value** argument specifies the missing value indicator for the output variable in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 14) The **-ngp=** argument can be used if you have memory problems when running `comp_symlin_filter_3d` on very large datasets. By default, the *number_of_grid_points* is set to the number of cells in the selected domain. In case of memory problems, you can use the **-ngp=** argument with a lower value. This will reduce the memory used by the operator.

- 15) If the **-hamming** argument is specified, a Hamming window filter is used instead of Lanczos window filter.

- 16) The **-win=** argument controls the form of the window which will be convolved with the filter if a Hamming window filter is requested with the **-hamming** argument. By default, a Hamming window is used (e.g., **-win=0.54**).

Set **-win=0.5** for using a Hanning window or **-win=1.** for a rectangular window (e.g., an “ideal” filter).

This argument has an effect only if the **-hamming** argument is also specified. The **-win=** argument is a real number greater or equal to 0.5 and less or equal to 1.

- 17) The **-notestf** argument allows to bypass some of the restrictions on the number of filter coefficients as specified with the **-nfc=** argument.

By default, the value of the **-nfc=** argument must be chosen such that the full transition bands about the cutoff frequencies ($1/PH$ and $1/PL$) of the selected filter are inside the $[0., 0.5]$ frequency interval.

When the **-notestf** argument is specified, only the cutoff frequencies (e.g., $1/PH$ and $1/PL$) of the selected filter must lie in the $[0., 0.5]$ frequency interval and not the full transition bands around them.

This allows to diminish the number of filter coefficients and, thus, to minimize the number of data points lost by the filtering operation (if **-nfc=** is set to K , $(K-1)/2$ data points will be “lost” or affected by end effects from each end of the series).

- 18) The **-usefft** argument specifies that the linear symmetric filter must be applied in the frequency domain by using a Fast Fourier Transform and the convolution theorem. Moreover, if the **-usefft** argument is specified, the values at the ends of the output filtered series are computed implicitly by assuming that the input series is part of a periodic sequence.

- 19) The **-double** argument specifies that the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 20) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 21) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 22) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 23) If the multi-channel time series has a seasonal or diurnal cycle, use *comp_stl_4d* or *comp_clim_4d* to remove the pure harmonic components from the time series before filtering.

- 24) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.

- 25) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 26) For more details on linear symmetric filtering and examples of use in the climate literature, see

- “Fourier analysis of time series- An introduction”, by Bloomfield, P., John Wiley and Sons, New York, Chapter 6, 1976. <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- “Lanczos filtering in one and two dimensions”, by Duchon, C., Journal of applied meteorology, vol. 18, 1016-1022, 1979. doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)

Outputs

`comp_symlin_filter_4d` creates an output NetCDF file that contains the filtered time series estimated from the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variable (in the description below, `nlev`, `nlat` and `nlon` are the lengths of the geographical dimensions of the input NetCDF variable and `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_filt(ntime, nlev, nlat, nlon)` : the filtered time series for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

The filtered multi-channel time series is packed in a fourdimensional variable whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=`, `-y=` and `-z=` arguments. However, outside the selected domain, the output NetCDF variable is filled with missing values.

Examples

- 1) For Hamming window filtering a real multi-channel monthly time series between 18 and 30 months (e.g., biennial time scale) from a fourdimensional NetCDF variable called `uwnd` extracted from the file `uwnd.monthly.mean.ncep2.nc`, which includes monthly time series, and store the results in the NetCDF file `qbo_uwnd_ncep2.nc`, use the following command :

```
$ comp_symlin_filter_4d \
  -f=uwnd.monthly.mean.ncep2.nc \
  -v=uwnd \
  -m=mesh_mask_uwnd_ncep2.nc \
  -pl=18 \
  -ph=30 \
  -hamming \
  -o=qbo_uwnd_ncep2.nc
```

2.75 comp_trend_1d

2.75.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.75.2 Latest revision

30/04/2026

2.75.3 Purpose

Extract a trend component from an unidimensional variable in a NetCDF dataset by using a LOESS smoother [Cleveland] [Cleveland_Devlin]. The LOESS procedure is a powerful statistical technique for describing a discrete time series (see the references in the Remarks Section below). In the LOESS procedure, the analyzed time series is decomposed into two terms:

$$\mathbf{X}(t) = \mathbf{T}(t) + \mathbf{R}(t)$$

where t refers to a time index, the **T** term is used to quantify the trend and low-frequency variations in the time series, and, finally, the **R** term contains the residual component.

The trend is estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., Loess) to data windows whose length is chosen by the user. Other important features of the LOESS method are the specification of the amount of trend smoothing according to the choice of the user, the ability to produce a robust estimate of the trend component that is not distorted by aberrant behavior in the data and the stationarity of the **R** residual time series.

This procedure returns the trend component as estimated by the LOESS procedure and, optionally, the residuals in a NetCDF dataset, for the raw time series associated with a NetCDF variable.

If the NetCDF variable is trid- or fourdimensional use `comp_trend_3d` or `comp_trend_4d`, respectively instead of `comp_trend_1d`. If the time series has a seasonal (or diurnal) cycle, use `comp_stl_1d` instead of `comp_trend_1d` in order to estimate the trend component by the Loess procedure.

The exact meaning and default values for most of the optional parameters of `comp_trend_1d` are exactly the same as in the original (STL) procedure described by [Cleveland_etal] and the user is referred to this publication for further details on the LOESS procedure as implemented here.

2.75.4 Further Details

Usage

```
$ comp_trend_1d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -nt=trend_smoother_length      (nt) \
  -t=time1,time2                 (optional) \
  -a=type_of_analysis            (optional : trend, residual) \
  -o=output_netcdf_file          (optional) \
  -itdeg=trend_smoother_degree   (itdeg) (optional : 0, 1, 2 ) \
  -ntjump=trend_skipping_value   (ntjump) (optional) \
  -maxit=max_robustness_iterations (maxit) (optional) \
  -smt=trend_smoothing_factor    (optional) \
  -robust                        (optional) \
  -double                        (optional) \
  -hdf5                         (optional) \
  -cdf5                         (optional) \
  -tlimited                      (optional)
```

By default

- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to `trend`. This means that the residual times series is not computed and not stored in the *output_netcdf_file*
- o=** the *output_netcdf_file* is named `trend_netcdf_variable.nc`
- itdeg=** the *trend_smoother_degree* is set to 1
- ntjump=** the *trend_skipping_value* is set to `nt/10` where `nt` is the value of the *trend_smoother_length* argument
- maxit=** the *max_robustness_iterations* is set to 15
- smt=** no smoothing is applied to the trend component

- robust** robustness iterations are not used, by default
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the time series must be decomposed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $\text{ntime} = \text{time2} - \text{time1} + 1$ time observations.
- 3) The **-nt=** argument specifies the length of the trend smoother, *nt*. The value of *nt* should be an odd integer greater than or equal to 3. As *nt* increases the values of the trend component become smoother.
- 4) The **-itdeg=** argument specifies the degree of the locally-fitted polynomial in trend smoothing. The value is 0, 1 or 2.
- 5) The **-ntjump=** argument specifies the skipping value for trend smoothing. The trend smoother skips ahead *ntjump* points and then linearly interpolates in between. The value of *ntjump* should be a positive integer; if *ntjump* is set to 1, a trend smooth is calculated at all points in the time series. To make the procedure run faster, a reasonable choice for *ntjump* is 10% or 20% of *nt*.
- 6) The **-a=** argument specifies if the residuals from the trend component are stored in the output NetCDF file. If:
 - **-a=trend**, the residuals are not computed
 - **-a=residual**, the residuals are computed and stored.
 The default is **-a=trend**, e.g., the residuals are not stored. Note that in all cases, the trend component is computed and stored in the output NetCDF file.
- 7) If **-robust** is specified, robustness iterations are carried out until convergence of the trend component or with a maximum of *maxit* iterations as specified by the **-maxit=** argument. Convergence occurs if the maximum changes in individual trend fits are less than 1% of the component's range after the previous iteration.
- 8) **-smt=trend_smoothing_factor** means that the trend component extracted from the *netcdf_variable* (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * \text{trend_smoothing_factor} + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to *residual*). *trend_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original LOESS procedure.
- 9) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 10) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 11) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 12) It is assumed that the data has no missing values.

- 13) If the time series have a seasonal or diurnal cycle, use *comp_stl_1d* instead of *comp_trend_1d*.

- 14) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 15) For more details on the LOESS procedure, see

- “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: [10.1080/01621459.1979.10481038](https://doi.org/10.1080/01621459.1979.10481038)
- “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: [10.1080/01621459.1988.10478639](https://doi.org/10.1080/01621459.1988.10478639)
- “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>

Outputs

comp_trend_1d creates an output NetCDF file that contains the trend and residual components extracted from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, *ntime* is the selected number of time observations) :

- 1) *netcdf_variable_trend(ntime)* : the trend component for the time series associated with the input NetCDF variable.
- 2) *netcdf_variable_residual(ntime)* : the residual component for the time series associated with the input NetCDF variable.

This variable is stored only if the **-a=residual** argument has been specified when calling *comp_trend_1d*.

Examples

- 1) For estimating a trend from the unidimensional NetCDF variable called *ts* extracted from the file *ts_gfdl_cm2_0.picntrl_monthly_glob.nc*, which includes a monthly anomalies time series, and store the results in the NetCDF file *ts_gfdl_cm2_0.picntrl_monthly_glob_trend.nc*, use the following command :

```
$ comp_trend_1d \
-f=ts_gfdl_cm2_0.picntrl_monthly_glob.nc \
-v=ts \
-nt=111 \
-a=trend \
-robust \
-o=ts_gfdl_cm2_0.picntrl_monthly_glob_trend.nc
```

2.76 comp_trend_3d

2.76.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.76.2 Latest revision

30/04/2026

2.76.3 Purpose

Extract a trend component from a tridimensional variable in a NetCDF dataset by using a LOESS smoother [Cleveland] [Cleveland_Devlin]. The LOESS procedure is a powerful statistical technique for describing a discrete time series (see the references in the Remarks Section below). In the LOESS procedure, the analyzed multi-channel time series is decomposed into two terms:

$$\mathbf{X}(t) = \mathbf{T}(t) + \mathbf{R}(t)$$

where t refers to a time index, the \mathbf{T} term is used to quantify the trend and low-frequency variations in the time series, and, finally, the \mathbf{R} term contains the residual component.

The trend is estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., Loess) to data windows whose length is chosen by the user. Other important features of the LOESS method are the specification of the amount of trend smoothing according to the choice of the user, the ability to produce robust estimates of the trend component that are not distorted by aberrant behavior in the data and the stationarity of the \mathbf{R} time series.

This procedure returns the multi-channel trend component as estimated by the LOESS procedure and, optionally, the residuals in a NetCDF dataset, for the multi-channel time series associated with a NetCDF variable.

If the NetCDF variable is uni- or fourdimensional use `comp_trend_1d` or `comp_trend_4d`, respectively instead of `comp_trend_3d`. If the time series have a seasonal (or diurnal) cycle, use `comp_stl_3d` instead of `comp_trend_3d` in order to estimate the trend component by the Loess procedure.

The exact meaning and default values for most of the optional parameters of `comp_trend_3d` are exactly the same as in the original (STL) procedure described by [Cleveland_et al] and the user is referred to this publication for further details on the LOESS procedure as implemented here.

This procedure is parallelized if OpenMP is used.

2.76.4 Further Details

Usage

```
$ comp_trend_3d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -nt=trend_smoother_length      (nt) \
  -m=input_mesh_mask_netcdf_file (optional) \
  -g=grid_type                   (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                   (optional) \
  -y=lat1,lat2                   (optional) \
  -t=time1,time2                 (optional) \
  -a=type_of_analysis            (optional : trend, residual) \
  -o=output_netcdf_file          (optional) \
  -itdeg=trend_smoother_degree  (itdeg) (optional : 0, 1, 2 ) \
  -ntjump=trend_skipping_value  (ntjump) (optional) \
  -maxit=max_robustness_iterations (maxit) (optional) \
  -smt=trend_smoothing_factor   (optional) \
  -robust                        (optional) \
  -mi=missing_value             (optional) \
  -double                        (optional) \
  -bigfile                       (optional) \
  -hdf5                          (optional) \
  -cdf5                          (optional) \
  -tlimited                       (optional)
```

By default

- m=** an *input_mesh_mask_netcdf_file* is not used
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x=** the whole longitude domain associated with the *netcdf_variable*
- y=** the whole latitude domain associated with the *netcdf_variable*
- t=** the whole time period associated with the *netcdf_variable*
- a=** the *type_of_analysis* is set to *trend*. This means that the residual times series are not computed and not stored in the *output_netcdf_file*
- o=** the *output_netcdf_file* is named *trend_netcdf_variable.nc*
- itdeg=** the *trend_smoother_degree* is set to 1
- ntjump=** the *trend_skipping_value* is set to $nt/10$ where *nt* is the value of the *trend_smoother_length* argument
- maxit=** the *max_robustness_iterations* is set to 15
- smt=** no smoothing is applied to the trend component
- robust** robustness iterations are not used, by default
- mi=** the *missing_value* for the output variables is equal to $1.e+20$
- double** the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file

- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the multi-channel time series must be decomposed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the whole geographical domain associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from $nlon+lon1+1$ to *lon2* where *nlon* is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_3d](#) for transforming geographical coordinates as indices before using *comp_trend_3d*.

- 3) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $ntime = time2 - time1 + 1$ time observations.

- 4) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian and as such has no duplicate points.

This argument is also used to determine the name of the NetCDF *mesh_mask* variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument.

- 6) The **-nt=** argument specifies the length of the trend smoother, *nt*. The value of *nt* should be an odd integer greater than or equal to 3. As *nt* increases the values of the trend component become smoother.
- 7) The **-itdeg=** argument specifies the degree of the locally-fitted polynomial in trend smoothing. The value is 0, 1 or 2.
- 8) The **-ntjump=** argument specifies the skipping value for trend smoothing. The trend smoother skips ahead *ntjump* points and then linearly interpolates in between. The value of *ntjump* should be a positive integer; if *ntjump* is set to 1, a trend smooth is calculated at all points in the time series. To make the procedure run faster, a reasonable choice for *ntjump* is 10% or 20% of *nt*.
- 9) The **-a=** argument specifies if the residuals from the trend component are stored in the output NetCDF file. If:
 - **-a=trend**, the residuals are not computed
 - **-a=residual**, the residuals are computed and stored.

The default is **-a=trend**, e.g., the residuals are not stored. Note that in all cases, the trend component is computed and stored in the output NetCDF file.

- 10) If **-robust** is specified, robustness iterations are carried out until convergence of the trend component or with a maximum of `maxit` iterations as specified by the **-maxit=** argument. Convergence occurs if the maximum changes in individual trend fits are less than 1% of the component's range after the previous iteration.
- 11) **-smt=trend_smoothing_factor** means that the trend component extracted from the *netcdf_variable* (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * \text{trend_smoothing_factor} + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to `residual`). *trend_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original LOESS procedure.
- 12) The **-mi=missing_value** argument specifies the missing value attribute for the output variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to `1.e+20`.

- 13) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.

- 18) If the time series have a seasonal or diurnal cycle, use *comp_stl_3d* instead of *comp_trend_3d*.

- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

- 20) For more details on the LOESS procedure, see

- “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: [10.1080/01621459.1979.10481038](https://doi.org/10.1080/01621459.1979.10481038)
- “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: [10.1080/01621459.1988.10478639](https://doi.org/10.1080/01621459.1988.10478639)
- “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>

Outputs

`comp_trend_3d` creates an output NetCDF file that contains the trend and residual components extracted from the time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, `nlat` and `nlon` are the lengths of the spatial dimensions of the input NetCDF variable and `ntime` is the selected number of time observations) :

- 1) `netcdf_variable_trend(ntime, nlat, nlon)` : the trend component for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.
- 2) `netcdf_variable_residual(ntime, nlat, nlon)` : the residual component for each of the time series of the 2-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the `-a=residual` argument has been specified when calling `comp_trend_3d`.

The trend and residual multi-channel components are packed in tridimensional variables whose first and second dimensions are exactly the same as those associated with the input NetCDF variable `netcdf_variable` even if you restrict the geographical domain with the `-x=` and `-y=` arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For estimating a trend from the tridimensional NetCDF variable called `sosstsst` extracted from the file `ST7_1m_00101_20012_sosstsst_ano_grid_T.nc`, which includes monthly anomalies time series, and store the results in the NetCDF file `stl_ST7_1m_00101_20012_sosstsst_ano_grid_T.nc`, use the following command :

```
$ comp_trend_3d \
-f=ST7_1m_00101_20012_sosstsst_ano_grid_T.nc \
-v=sosstsst \
-nt=127 \
-a=residual \
-robust \
-o=stl_ST7_1m_00101_20012_sosstsst_ano_grid_T.nc
```

2.77 comp_trend_4d

2.77.1 Authors

Pascal Terray (LOCEAN/IPSL)

2.77.2 Latest revision

30/04/2026

2.77.3 Purpose

Extract a trend component from a fourdimensional variable in a NetCDF dataset by using a LOESS smoother [Cleveland] [Cleveland_Devlin]. The LOESS procedure is a powerful statistical technique for describing a discrete time series (see the references in the Remarks Section below). In the LOESS procedure, the analyzed multi-channel time series is decomposed into two terms:

$$\mathbf{X}(t) = \mathbf{T}(t) + \mathbf{R}(t)$$

where t refers to a time index, the \mathbf{T} term is used to quantify the trend and low-frequency variations in the time series, and, finally, the \mathbf{R} term contains the residual component.

The trend is estimated through a sequence of applications of locally weighted regression or low-order polynomial (e.g., Loess) to data windows whose length is chosen by the user. Other important features of the LOESS method are the specification of the amount of trend smoothing according to the choice of the user, the ability to produce robust estimates of the trend component that are not distorted by aberrant behavior in the data and the stationarity of the \mathbf{R} time series.

This procedure returns the multi-channel trend component as estimated by the LOESS procedure and, optionally, the residuals in a NetCDF dataset, for the multi-channel time series associated with a NetCDF variable.

If the NetCDF variable is uni- or tridimensional use `comp_trend_1d` or `comp_trend_3d`, respectively instead of `comp_trend_4d`. If the time series have a seasonal (or diurnal) cycle, use `comp_stl_4d` instead of `comp_trend_4d` in order to estimate the trend component by the Loess procedure.

The exact meaning and default values for most of the optional parameters of `comp_trend_4d` are exactly the same as in the original (STL) procedure described by [Cleveland_etal] and the user is referred to this publication for further details on the LOESS procedure as implemented here.

This procedure is parallelized if OpenMP is used.

2.77.4 Further Details

Usage

```
$ comp_trend_4d \
  -f=input_netcdf_file \
  -v=input_netcdf_variable \
  -nt=trend_smoother_length      (nt) \
  -m=input_mesh_mask_netcdf_file      (optional) \
  -g=grid_type                    (optional : n, t, u, v, w, f) \
  -x=lon1,lon2                    (optional) \
  -y=lat1,lat2                    (optional) \
  -z=level1,level2                (optional) \
  -t=time1,time2                  (optional) \
  -a=type_of_analysis             (optional : trend, residual) \
  -o=output_netcdf_file           (optional) \
  -itdeg=trend_smoother_degree    (itdeg) (optional : 0, 1, 2) \
  -ntjump=trend_skipping_value    (ntjump) (optional) \
  -maxit=max_robustness_iterations (maxit) (optional) \
  -smt=trend_smoothing_factor     (optional) \
  -robust                         (optional) \
  -mi=missing_value               (optional) \
  -double                         (optional) \
  -bigfile                        (optional) \
  -hdf5                           (optional) \
  -cdf5                           (optional) \
  -tlimited                        (optional)
```

By default

-m= an `input_mesh_mask_netcdf_file` is not used

- g= the *grid_type* is set to `n` which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- x= the whole longitude domain associated with the *netcdf_variable*
- y= the whole latitude domain associated with the *netcdf_variable*
- z= the whole vertical resolution associated with the *netcdf_variable*
- t= the whole time period associated with the *netcdf_variable*
- a= the *type_of_analysis* is set to `trend`. This means that the residual times series are not computed and not stored in the *output_netcdf_file*
- o= the *output_netcdf_file* is named `trend_netcdf_variable.nc`
- itdeg= the *trend_smoother_degree* is set to 1
- ntjump= the *trend_skipping_value* is set to `nt/10` where `nt` is the value of the *trend_smoother_length* argument
- maxit= the *max_robustness_iterations* is set to 15
- smt= no smoothing is applied to the trend component
- robust robustness iterations are not used, by default.
- mi= the *missing_value* for the output variables is equal to `1.e+20`
- double the results are stored as single-precision floating point numbers in the output NetCDF file. If **-double** is activated, the results are stored as double-precision floating point numbers
- bigfile a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- tlimited -cdf5
a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable from which the multi-channel time series must be decomposed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) If the **-x=lon1,lon2**, **-y=lat1,lat2** and **-z=level1,level2** arguments are missing, the whole geographical domain and vertical resolution associated with the *netcdf_variable* is used to select the multi-channel time series.

The longitude, latitude or level range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are allowed for *lon1*. In this case the longitude domain is from `nlon+lon1+1` to *lon2* where `nlon` is the number of longitude points in the grid associated with the NetCDF variable and it is assumed that the grid is periodic.

Refer to [comp_mask_4d](#) for transforming geographical coordinates as indices before using `comp_trend_4d`.

- 3) If the **-t=time1,time2** argument is missing, the whole time period associated with the *netcdf_variable* is used to decompose the time series.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 4) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file*) and the mask (in the *input_mesh_mask_netcdf_file*) must agree if the **-m=** argument is used.
- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the NEMO model (ORCA configuration and R2, R4 or R05 resolutions).

If **-g=** is set to *n*, it is assumed that the 3-D grid-mesh is regular or Gaussian and as such has no duplicate points.

This argument is also used to determine the name of the NetCDF mesh_mask variable if an *input_mesh_mask_netcdf_file* is used as specified with the **-m=** argument.

- 6) The **-nt=** argument specifies the length of the trend smoother, *nt*. The value of *nt* should be an odd integer greater than or equal to 3. As *nt* increases the values of the trend component become smoother.
- 7) The **-itdeg=** argument specifies the degree of the locally-fitted polynomial in trend smoothing. The value is 0, 1 or 2.
- 8) The **-ntjump=** argument specifies the skipping value for trend smoothing. The trend smoother skips ahead *ntjump* points and then linearly interpolates in between. The value of *ntjump* should be a positive integer; if *ntjump* is set to 1, a trend smooth is calculated at all points in the time series. To make the procedure run faster, a reasonable choice for *ntjump* is 10% or 20% of *nt*.
- 9) The **-a=** argument specifies if the residuals from the trend component are stored in the output NetCDF file. If:
 - **-a=trend**, the residuals are not computed
 - **-a=residual**, the residuals are computed and stored.

The default is **-a=trend**, e.g., the residuals are not stored. Note that in all cases, the trend component is computed and stored in the output NetCDF file.

- 10) If **-robust** is specified, robustness iterations are carried out until convergence of the trend component or with a maximum of *max_robustness_iterations* iterations as specified by the **-maxit=** argument. Convergence occurs if the maximum changes in individual trend fits are less than 1% of the component's range after the previous iteration.
- 11) **-smt=trend_smoothing_factor** means that the trend component extracted from the *netcdf_variable* (e.g., the **-v=** argument) must be smoothed with a moving average of approximately $2 * trend_smoothing_factor + 1$ terms. The smoothing is applied before estimating the residuals (e.g., if **-a=** argument is set to *residual*). *trend_smoothing_factor* must be a strictly positive integer (e.g., greater than 0). Note that this additional smoothing is not implemented in the original LOESS procedure.
- 12) The **-mi=missing_value** argument specifies the missing value attribute for the output variables in the *output_netcdf_file*.

If the **-mi=** argument is not specified, the *missing_value* is set to $1.e+20$.

- 13) The **-double** argument specifies that, the results are stored as double-precision floating point numbers in the output NetCDF file.

By default, the results are stored as single-precision floating point numbers in the output NetCDF file.

- 14) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 15) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 16) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 17) It is assumed that the data has no missing values excepted those associated with a constant land-sea mask.
- 18) If the time series have a seasonal or diurnal cycle, use *comp_stl_4d* instead of *comp_trend_4d*.
- 19) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.
- 20) For more details on the LOESS procedure, see

- “Robust Locally Weighted Regression and Smoothing Scatterplots”, by Cleveland, W.S., Journal of the American Statistical Association, Vol. 74, 829-836, 1979. doi: [10.1080/01621459.1979.10481038](https://doi.org/10.1080/01621459.1979.10481038)
- “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting”, by Cleveland, W.S., and Devlin, S.J., Journal of the American Statistical Association, Vol. 83, 596-610, 1988. doi: [10.1080/01621459.1988.10478639](https://doi.org/10.1080/01621459.1988.10478639)
- “A Seasonal-Trend Decomposition Procedure Based on Loess”, by Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., Journal of Official Statistics, 6, 3-73, 1990. <http://www.jos.nu/Articles/abstract.asp?article=613>

Outputs

comp_trend_4d creates an output NetCDF file that contains the trend and residual components extracted from the multi-channel time series associated with the input NetCDF variable. The output NetCDF dataset contains the following NetCDF variables (in the description below, *nlev*, *nlat* and *nlon* are the lengths of the geographical dimensions of the input NetCDF variable and *ntime* is the selected number of time observations) :

- 1) *netcdf_variable_trend*(*ntime*, *nlev*, *nlat*, *nlon*) : the trend component for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.
- 2) *netcdf_variable_residual*(*ntime*, *nlev*, *nlat*, *nlon*) : the residual component for each of the time series of the 3-D grid-mesh associated with the input NetCDF variable.

This variable is stored only if the **-a=residual** argument has been specified when calling *comp_trend_4d*.

The trend and residual multi-channel components are packed in fourdimensional variables whose first, second and third dimensions are exactly the same as those associated with the input NetCDF variable *netcdf_variable* even if you restrict the geographical domain with the **-x=**, **-y=** and **-z=** arguments. However, outside the selected domain, the output NetCDF variables are filled with missing values.

Examples

- 1) For estimating a trend from the fourdimensional NetCDF variable called `votemper` extracted from the file `ST7_1m_00101_20012_votemper_ano_grid_T.nc`, which includes monthly anomalies time series, and store the results in the NetCDF file `stl_ST7_1m_00101_20012_votemper_ano_grid_T.nc`, use the following command :

```
$ comp_trend_4d \  
-f=ST7_1m_00101_20012_votemper_ano_grid_T.nc \  
-v=votemper \  
-nt=127 \  
-a=residual \  
-robust \  
-o=stl_ST7_1m_00101_20012_votemper_ano_grid_T.nc
```

2.78 pack_masked_data_3d

2.78.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.78.2 Latest revision

30/04/2026

2.78.3 Purpose

Reduce the size of a tridimensional NetCDF variable (or parts of it) extracted from a NetCDF dataset by saving only unmasked data associated with a constant land-sea mask, as provided by the the **-m=input_mesh_mask_netcdf_file** argument, in an unidimensional NetCDF variable and store it in an output NetCDF dataset.

The original coordinate variables, if present in the input NetCDF dataset, are also copied in the output NetCDF dataset.

Optionally, the mask and scale factors of the 2-D grid-mesh associated with the input NetCDF variable, if present in the mesh-mask NetCDF dataset, specified with the **-m=input_mesh_mask_netcdf_file** argument, can also be stored in packed format in the output NetCDF dataset at the user option.

The output NetCDF variable stored in packed format in the output NetCDF file can later be unpacked with [*un-pack_masked_data_3d*](#).

2.78.4 Further Details

Usage

```
$ pack_masked_data_3d \  
-f=input_netcdf_file \  
-v=netcdf_variable \  
-m=input_mesh_mask_netcdf_file \  
-o=output_netcdf_file \  
-x=lon1,lon2 (optional) \  

```

(continues on next page)

(continued from previous page)

-y=lat1,lat2	(optional) \
-t=time1,time2	(optional) \
-g=grid_type	(optional : n, t, u, v, w, f) \
-3d	(optional) \
-scalfac	(optional) \
-bigfile	(optional) \
-hdf5	(optional) \
-cdf5	(optional) \
-tlimited	(optional)

By default

- x= the whole longitude domain associated with the *netcdf_variable*
- y= the whole latitude domain associated with the *netcdf_variable*
- t= the whole time period associated with the *netcdf_variable*
- g= the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the input NetCDF variable is assumed to be regular or Gaussian
- 3d the packed data are defined as a twodimensional NetCDF variable. However, if **-3d** is activated, the packed data is defined as an tridimensional NetCDF variable but with one dummy dimension (e.g., with a length equal to 1)
- scalfac The scale factors NetCDF variables in the *input_mesh_mask_netcdf_file* are not packed and copied to the output NetCDF file. However, if the **-scalfac** argument is specified, the associated packed scale factors NetCDF variables are copied and a dummy mask NetCDF variable (e.g., with all values set to 1 . is also written in the output NetCDF file
- bigfile a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5 a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5 a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=netcdf_variable** argument specifies the NetCDF variable which must be packed and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The argument **-m=input_mesh_mask_netcdf_file** specifies that the land-sea mask to apply to the *netcdf_variable* must be read from the *input_mesh_mask_netcdf_file*.
This *input_mesh_mask_netcdf_file* may be created by *comp_clim_3d* if the 2-D grid-mesh is regular or gaussian.
- 3) If the **-x=lon1,lon2** and **-y=lat1,lat2** arguments are missing, the geographical domain used in the packing operation is determined from the attributes of the input mesh-mask NetCDF variable named *grid_typedmask* (e.g., the lon1_Eastern_limit, lon2_Western_limit, lat1_Southern_limit and lat2_Northern_limit attributes), which is read

from the input NetCDF file *input_mesh_mask_netcdf_file*. If these attributes are missing, the whole geographical domain associated with the *netcdf_variable* is used to construct the packed NetCDF variable in the output NetCDF dataset.

The longitude or latitude range must be a vector of two integers specifying the first and last selected indices along each dimension. The indices are relative to 1. Negative values are not allowed for *lon1*.

Refer to *comp_mask_3d* for transforming geographical coordinates as indices or generating an appropriate mesh-mask before using *pack_masked_data_3d*.

- 4) If the **-t=***time1,time2* argument is missing the whole time period associated with the *netcdf_variable* is used to construct the packed NetCDF variable.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have `ntime = time2 - time1 + 1` time observations.

- 5) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the NetCDF variable is from an experiment with the ORCA or NEMO model.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian.

The **-g=** argument is also used to determine the name of the NetCDF variable which contains the 2-D mesh-mask in the *input_mesh_mask_netcdf_file* (e.g., this variable is named *grid_typedmask*).

- 6) The geographical shapes of the *netcdf_variable* (in the *input_netcdf_file* dataset) and the land-sea mask (in the *input_mesh_mask_netcdf_file* dataset) must agree.
- 7) The **-3d** argument specifies that the packed data must be stored as a tridimensional NetCDF variable with one dummy dimension in the output NetCDF file.

By default, the packed data are stored as a twodimensional NetCDF variable.

- 8) The **-scalfac** argument specifies that the scale factors from the *input_mesh_mask_netcdf_file* must be packed and stored in the output NetCDF file if they exist. If the **-scalfac** argument is specified, a dummy mask is also written in the output NetCDF file. This allows further processing of the packed data by other NCSTAT procedures. Note that the *grid_type* of the packed mask and scale factors in the output NetCDF file is set to *n* in all cases, even if the original 2-D grid-mesh is not regular or gaussian.

By default, the packed scale factors and a dummy mask are not stored in the output NetCDF file.

- 9) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 10) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 11) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 12) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Examples

- 1) For packing a NetCDF variable `sosstsst` extracted from the NetCDF dataset `ST7_1m_0101_20012_grid_T_sosstsst.nc` with the help of a land-sea mask extracted from the NetCDF file `mesh_ocean.nc` and store the results in a NetCDF file named `packed_ST7_1m_0101_20012_grid_T_sosstsst.nc` use the following command (note that the variable `sosstsst` is from a NEMO simulation since `-g=t` is specified) :

```
$ pack_masked_data_3d \
  -f=ST7_1m_0101_20012_grid_T_sosstsst.nc \
  -v=sosstsst \
  -m=mesh_ocean.nc \
  -o=packed_ST7_1m_0101_20012_grid_T_sosstsst.nc \
  -g=t
```

2.79 unpack_masked_data_3d

2.79.1 Authors

Eric Maisonnave (CERFACS) and Pascal Terray (LOCEAN/IPSL)

2.79.2 Latest revision

30/04/2026

2.79.3 Purpose

Unpack a packed tridimensional NetCDF variable produced by [pack_masked_data_3d](#) or derived from an NCSTAT analysis of such packed tridimensional NetCDF variable and store the results in an output NetCDF dataset.

2.79.4 Further Details

Usage

```
$ unpack_masked_data_3d \
  -f=input_netcdf_file \
  -v=packed_netcdf_variable \
  -m=input_mesh_mask_netcdf_file \
  -o=output_netcdf_file \
  -t=time1,time2 (optional) \
  -g=grid_type (optional : n, t, u, v, w, f) \
  -bigfile (optional) \
  -hdf5 (optional) \
  -cdf5 (optional) \
  -tlimited (optional)
```

By default

- t=** the whole time period associated with the *packed_netcdf_variable*
- g=** the *grid_type* is set to *n* which means that the 2-D grid-mesh associated with the original (unpacked) NetCDF variable is assumed to be regular or Gaussian
- bigfile** a NetCDF classical format file is created. If **-bigfile** is activated, the output NetCDF file is a 64-bit offset format file
- hdf5** a NetCDF classical format file is created. If **-hdf5** is activated, the output NetCDF file is a NetCDF-4/HDF5 format file
- cdf5** a NetCDF classical format file is created. If **-cdf5** is activated, the output NetCDF file is a CDF5 format file
- tlimited** the time dimension is defined as unlimited in the output NetCDF file. However, if **-tlimited** is activated, the time dimension is defined as limited in the output NetCDF file

Remarks

- 1) The **-v=packed_netcdf_variable** argument specifies the NetCDF variable which must be unpacked and the **-f=input_netcdf_file** argument specifies that this NetCDF variable must be extracted from the NetCDF file *input_netcdf_file*.
- 2) The argument **-m=input_mesh_mask_netcdf_file** specifies that the land-sea mask to use for unpacking the *packed_netcdf_variable* must be read from the *input_mesh_mask_netcdf_file*.

This *input_mesh_mask_netcdf_file* may be created by *comp_clim_3d* if the 2-D grid-mesh is regular or gaussian and it must be the one used originally to pack the data with *pack_masked_data_3d*.

- 3) If the **-t=time1,time2** argument is missing the whole time period associated with the *packed_netcdf_variable* is used to construct the output NetCDF variable.

The selected time period is a vector of two integers specifying the first and last time observations. The indices are relative to 1. Note that the output NetCDF file will have $n_{time} = time2 - time1 + 1$ time observations.

- 4) If **-g=** is set to *t*, *u*, *v*, *w* or *f* it is assumed that the original NetCDF variable is from an experiment with the ORCA or NEMO model.

If **-g=** is set to *n*, it is assumed that the 2-D grid-mesh is regular or Gaussian.

The **-g=** argument is also used to determine the name of the NetCDF variable which contains the 2-D mesh-mask in the *input_mesh_mask_netcdf_file* (e.g., it is assumed that this variable is named *grid_typedmask*).

- 5) The **-bigfile** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF36` or `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 3.6 library or higher.

If this argument is specified, the *output_netcdf_file* will be a 64-bit offset format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF36`, `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 6) The **-hdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros (e.g., `-D_USE_NETCDF4` or `-D_USE_NETCDF44`) and linked to the NetCDF 4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a NetCDF-4/HDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF4` or `_USE_NETCDF44` CPP macros.

- 7) The **-cdf5** argument is allowed only if the NCSTAT software has been compiled with the `_USE_NETCDF44` CPP macro (e.g., `-D_USE_NETCDF44`) and linked to the NetCDF 4.4 library or higher.

If this argument is specified, the *output_netcdf_file* will be a CDF5 format file instead of a NetCDF classic format file. However, this argument is recognized in the procedure only if the NCSTAT software has been built with the `_USE_NETCDF44` CPP macro.

- 8) Duplicate parameters are allowed, but this is always the last occurrence of a parameter which will be used for the computations. Moreover, the number of specified parameters must not be greater than the total number of allowed parameters.

Examples

- 1) For unpacking a NetCDF variable `sosstsst` extracted from the NetCDF dataset `packed_ST7_1m_0101_20012_grid_T_sosstsst.nc` with the help of a land-sea mask extracted from the NetCDF file `mesh_ocean.nc` and store the results in a NetCDF file named `ST7_1m_0101_20012_grid_T_sosstsst.nc` use the following command (note that the variable `sosstsst` is from a NEMO simulation since **-g=t** is specified) :

```
$ unpack_masked_data_3d \  
-f=ST7_1m_0101_20012_grid_T_sosstsst.nc \  
-v=sosstsst \  
-m=mesh_ocean.nc \  
-o=ST7_1m_0101_20012_grid_T_sosstsst.nc \  
-g=t
```


BIBLIOGRAPHY

- [accelerate] *Accelerate – Apple Accelerate framework* <https://developer.apple.com/documentation/accelerate/>
- [aocl] *AOCL – AMD Optimizing CPU Libraries* <https://www.amd.com/en/developer/aocl.html>
- [arm] *ARM – Arm Performance Libraries* <https://developer.arm.com/downloads/-/arm-performance-libraries>
- [atlas] *ATLAS – Automatically Tuned Linear Algebra Software* <https://math-atlas.sourceforge.net/>
- [blas] *BLAS – Basic Linear Algebra Subprograms* http://www.netlib.org/blas/#_presentation
- [blis] *BLIS – BLAS-like Library Instantiation Software Framework* <https://github.com/flame/blis>
- [cdo] *CDO – Climate Data Operators* <https://code.zmaw.de/projects/cdo>
- [fortran] Metcalf, M., Reid, J., and Cohen, M., (2013) *Modern FORTRAN Explained*, Oxford University Press, Oxford, UK, seven ed..
- [gotoblas] *GotoBLAS* <https://github.com/ianmartin/GotoBLAS-2/blob/master/01Readme.txt>
- [mkl] *MKL – Intel Math Kernel Library (Intel MKL)* <http://software.intel.com/en-us/intel-mkl/>
- [mpi] *MPI – Message Passing Interface* <http://www.mcs.anl.gov/research/projects/mpi/>
- [nco] *NCO – netCDF Operator* <http://nco.sourceforge.net/>
- [netcdf] *NetCDF – network Common Data Form* <https://www.unidata.ucar.edu/software/netcdf/>
- [netcdf-f90] *NetCDF Fortran 90 Interface* https://docs.unidata.ucar.edu/netcdf-fortran/current/f90_The-NetCDF-Fortran-90-Interface-Guide.html
- [openblas] *OpenBlas – An optimized BLAS library* <https://github.com/xianyi/OpenBLAS>
- [openmp] *OpenMP – OpenMP Application Program Interface* <http://www.openmp.org>
- [pnetcdf] *Parallel netCDF: A Parallel I/O Library for NetCDF File Access* <https://github.com/Parallel-NetCDF/PnetCDF>
- [statpack] *STATPACK – A Statistical package* <https://pagesperso.locean-ipsl.upmc.fr/terray/statpack2.4>
- [Bjornsson_Venegas] Bjornsson, H., and Venegas, S.A., (1997) *A manual for EOF and SVD analyses of climate data*. McGill University, CCGCR Report No. 97-1, Montreal, Quebec, 52pp, <https://www.jsg.utexas.edu/fu/files/EOFSVD.pdf>
- [Bloomfield] Bloomfield, P., (1976) *Fourier analysis of time series- An introduction*. ohn Wiley and Sons, New York, <http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471889482.html>
- [Braun_Kulperger] Braun, W.J., and Kulperger, R.J., (1997) *Properties of a fourier bootstrap method for time series*. Communications in Statistics - Theory and Methods, vol 26, 1329-1336, doi: 10.1080/03610929708831985

- [Bretherton_etal] Bretherton, C., Smith, c., and Wallace, J.M., (1992) *An intercomparison of methods for finding coupled patterns in climate data*. Journal of Climate, Vol. 5, 541-560, doi: [10.1175/1520-0442\(1992\)005<0541:AIOMFF>2.0.CO;2](https://doi.org/10.1175/1520-0442(1992)005<0541:AIOMFF>2.0.CO;2)
- [Brown_Hall] Brown, T.J., and Hall, B.L., (1999) *The Use of t values in Composite Analyses*. Journal of climate, vol. 12, 2941-294, doi: [10.1175/1520-0442\(1999\)012<2941:TUOTVI>2.0.CO;2](https://doi.org/10.1175/1520-0442(1999)012<2941:TUOTVI>2.0.CO;2)
- [Burgers_Stephenson] Burgers, G., and Stephenson, .D.B, (1999) *The Normality of El Nino*. Geophysical Research Letters, Vol. 26, 1027-1030, doi: [10.1029/1999GL900161](https://doi.org/10.1029/1999GL900161)
- [Clarkson_Jennrich] Clarkson, D.B., and Jennrich, R.I., (1988) *Quartic rotation criteria and algorithms*. Psychometrika, Vol. 53, 251-259, <https://link.springer.com/article/10.1007/BF02294136>
- [Cleveland] Cleveland, W.S., (1979) *Robust Locally Weighted Regression and Smoothing Scatterplots*. Journal of the American Statistical Association, Vol. 74, 829-836, doi: [10.1080/01621459.1979.10481038](https://doi.org/10.1080/01621459.1979.10481038)
- [Cleveland_Devlin] Cleveland, W.S. and Devlin, S.J., (1988) *Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting*. Journal of the American Statistical Association, Vol. 83, 596-610, doi: [10.1080/01621459.1988.10478639](https://doi.org/10.1080/01621459.1988.10478639)
- [Cleveland_etal] Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I., (1990) *A Seasonal-Trend Decomposition Procedure Based on Loess*. Journal of Official Statistics, Vol. 6, 3-73, <http://www.jos.nu/Articles/abstract.asp?article=613>
- [Coates_Diggle] Coates, D.S., and Diggle, P.J., (1986) *Tests for comparing two estimated spectral densities*. Journal of Time series Analysis, Vol. 7:1, 7-20, doi: <https://doi.org/10.1111/j.1467-9892.1986.tb00482.x>
- [Davison_Hinkley] Davison, A.C., and Hinkley, D.V., (1997) *Bootstrap methods and their application*. Cambridge University press, Cambridge, UK., doi: [10.1017/CBO9780511802843](https://doi.org/10.1017/CBO9780511802843)
- [Diggle] Diggle, P.J., (1990) *Time series: a biostatistical introduction*. Clarendon Press, Oxford, 268 pp., ISBN-10: 0198522266, <http://www.oupcanada.com/catalog/9780198522263.html>
- [Diggle_Fisher] Diggle, P.J., and Fisher, N.I., (1991) *Nonparametric comparison of cumulative periodograms*. Applied Statistics, Vol. 40:3, 423-434, doi: <https://doi.org/10.2307/2347522>
- [Duchon] Duchon, C., (1979) *Lanczos filtering in one and two dimensions*. Journal of applied meteorology, vol. 18, 1016-1022, doi: [10.1175/1520-0450\(1979\)018<1016:LFIOAT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2)
- [Ebisuzaki] Ebisuzaki, W., (1997) *A method to estimate the statistical significance of a correlation when the data are serially correlated*. Journal of climate, vol. 10, 2147-2153, doi: [10.1175/1520-0442%281997%29010%3C2147%3AAMTETS%3E2.0.CO%3B2](https://doi.org/10.1175/1520-0442%281997%29010%3C2147%3AAMTETS%3E2.0.CO%3B2)
- [Efron] Efron, B., (1981) *Nonparametric standard-errors and confidence intervals*. The Canadian Journal of Statistics, vol. 9, 2, 139-172, doi: <https://doi.org/10.2307/3314608>
- [Golub_VanLoan] Golub, G.H., and Van Loan, C., (2013) *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD., 4rd Edition, 756 pp., ISBN: 978-1-4214-0794-4
- [Halko_etal] Halko, N., Martinsson, P.G., and Tropp, J.A., (2011) *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*. SIAM Rev., Vol. 53, 217-288, doi: <https://epubs.siam.org/doi/abs/10.1137/090771806>
- [Hannachi] Hannachi, A., (2004) *A primer for EOF analysis of climate data*. Reading, UK, 33pp, 2004., <http://www.met.reading.ac.uk/~han/Monitor/eofprimer.pdf>
- [Iacobucci_Noullez] Iacobucci, A., and Noullez, A., (2005) *A Frequency Selective Filter for Short-Length Time Series*. Computational Economics, Vol. 25, 75-102, doi: [10.1007/s10614-005-6276-7](https://doi.org/10.1007/s10614-005-6276-7)
- [Jackson] Jackson, J.E., (2003) *A user's guide to principal components*. John Wiley and Sons, New York, USA, 592 pp., ISBN: 978-0-471-47134-9

- [Jenkins_Watts] Jenkins, G.M., and Watts, D.G., (1968) *Spectral Analysis and its Applications*. San Francisco: Holden-Day. ISBN-10: 0816244642
- [Jolliffe] Jolliffe, I.T., (2002) *Principal Component Analysis*. Springer-Verlag, New York, USA, 487 pp., 2nd Ed, ISBN: 978-0-387-22440-4, <https://www.springer.com/gp/book/9780387954424>
- [Li_etal] Li, H., Linderman, G.C., Szlam, A., Stanton, K.P., Kluger, Y., and Tygert, M., (2017) *Algorithm 971: An implementation of a randomized algorithm for principal component analysis*. ACM Trans. Math. Softw. Vol. 43, 3, Article 28 (January 2017), <https://pubmed.ncbi.nlm.nih.gov/28983138>
- [Martinsson] Martinsson, P.G., (2019) *Randomized methods for matrix computations*. arXiv.1607.01649, <https://arxiv.org/abs/1607.01649>
- [Masson_etal] Masson, S., et al. (2012) *Impact of intra-daily SST variability on ENSO characteristics in a coupled model*. Climate Dynamics, Vol. 39, 681-707, doi: [10.1007/s00382-011-1247-2](https://doi.org/10.1007/s00382-011-1247-2)
- [Noreen] Noreen, E.W., (1989) *Computer-intensive methods for testing hypotheses: an introduction*. Wiley and Sons, New York, USA, ISBN: 978-0-471-61136-3
- [Potscher_Reschenhofer] Potscher, B.,M., and Reschenhofer, E., (1988) *Discriminating between two spectral densities in case of replicated observations*. Journal of Time series Analysis, 9:3, 221-224, doi: <https://doi.org/10.1111/j.1467-9892.1988.tb00466.x>
- [Potscher_Reschenhoferb] Potscher, B.,M., and Reschenhofer, E., (1989) *Distribution of the Coates-Diggle test statistic in case of replicated observations*. Statistics, 20:3, 417-421. DOI: <https://doi.org/10.1080/02331888908802190>
- [Politis] Politis, D.,N., (1998) *Computer-Intensive Methods in Statistical Analysis..* IEEE Signal Processing Magazine, 15, 1, 39-55, (January 1998), doi: <https://ieeexplore.ieee.org/document/647042>
- [Priestley] Priestley, M.B., (1981) *Spectral Analysis and Time Series*. London: Academic Press. ISBN-10: 0125649223
- [Richman] Richman, M.B., (1986) *Rotation of principal components*. International Journal of Climatology, Vol. 6, 293-335, doi: <https://doi.org/10.1002/joc.3370060305>
- [Rusta] Rust, B.W., (2001) *Fitting nature's basic functions Part I: polynomials and linear least squares* Computing in Science and Engineering, Vol. 3, no 5, 84-89, doi: [10.1109/MCISE.2001.947111](https://doi.org/10.1109/MCISE.2001.947111)
- [Rustb] Rust, B.W., (2001) *Fitting nature's basic functions Part II: estimating uncertainties and testing hypotheses*. Computing in Science and Engineering, Vol. 3, no 6, 60-64, doi: [10.1109/5992.963429](https://doi.org/10.1109/5992.963429)
- [Schneider] Schneider, T., (2001) *Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values*. Journal of Climate, Vol. 14, 853-871, doi: [10.1175/1520-0442\(2001\)014<0853:AOICDE>2.0.CO;2](https://doi.org/10.1175/1520-0442(2001)014<0853:AOICDE>2.0.CO;2)
- [Terraya] Terray, P., (1995) *Space/Time structure of monsoons interannual variability*. Journal of Climate, Vol. 8, 2595-2619, doi: [10.1175/1520-0442\(1995\)008<2595:STSOMI>2.0.CO;2](https://doi.org/10.1175/1520-0442(1995)008<2595:STSOMI>2.0.CO;2)
- [Terrayb] Terray, P., (2002) *Application of Weighted Empirical Orthogonal Function Analysis to ship's datasets*. Compte-Rendu de la IVeme journee Statistique IPSL (Classification et Analyse spatiale). NAI no23. pp. 11-28, ISSN 1626-8334., https://www.lmd.polytechnique.fr/nai/nai_23.pdf
- [Terrayc] Terray, P., (2011) *Southern Hemisphere extra-tropical forcing: A new paradigm for El Nino-Southern Oscillation*. Climate Dynamics, Vol. 36, 2171-2199, doi: [10.1007/s00382-010-0825-z](https://doi.org/10.1007/s00382-010-0825-z)
- [Terrayd] Terray, P., (2025) *Variable projection framework for the reduced-rank matrix approximation problem by weighted least-squares*. arXiv:2505.03347, 220 pp., doi: [10.48550/arXiv.2505.03347](https://doi.org/10.48550/arXiv.2505.03347)
- [Terray_etal] Terray, P., Delecluse P., Labattu S., and Terray L., (2003) *Sea Surface Temperature associations with the Late Indian Summer Monsoon*. Climate Dynamics, vol. 21, 593-618, doi: [10.1007/s00382-003-0354-0](https://doi.org/10.1007/s00382-003-0354-0)

- [Terray_etalb] Terray, P. et al., (2012) *The role of the intra-daily SST variability in the Indian monsoon variability in a global coupled model* Climate Dynamics, vol. 39, 729-754, doi: [10.1007/s00382-011-1240-9](https://doi.org/10.1007/s00382-011-1240-9)
- [Theiler_etal] Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., and Farmer, J.D. (1992) *Testing for nonlinearity in time series: the method of surrogate data*. Physica D, vol. 58, 77-94, doi: [10.1016/0167-2789\(92\)90102-s](https://doi.org/10.1016/0167-2789(92)90102-s)
- [vonStorch_Zwiers] von Storch, H., and Zwiers, F.W., (2002) *Statistical Analysis in Climate Research*. Cambridge Univesity press, Cambridge, UK, 484 pp., ISBN: [9780521012300](https://www.isbn-international.org/product/9780521012300)
- [Welch] Welch, P.D., (1967) *The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms*. IEEE trans. on audio and electroacoustics, vol. 15:2, 70-73, DOI: <https://doi.org/10.1109/TAU.1967.1161901>
- [White] White, G., (1980) *Skewness, Kurtosis and Extreme Values of Northern Hemisphere Geopotential Heights*. Monthly Weather Review, Vol. 108, 1446-1455, doi: [10.1175/1520-0493\(1980\)108<1446:SKAEVO>2.0.CO;2](https://doi.org/10.1175/1520-0493(1980)108<1446:SKAEVO>2.0.CO;2)
- [Wills_etal] Wills, R.C., Schneider, T., Wallace, J.M., Battisti, D.S., and Hartmann, D.L. (2018) *Disentangling Global Warming, Multidecadal Variability, and El Nino in Pacific Temperatures*. Geophysical Research Letters, Vol. 45, 2487-2496, doi: <https://doi.org/10.1002/2017GL076327>
- [Zhang_etal] Zhang, Y., Norris, J.R., and Wallace, J.M., (1998) *Seasonality of large scale atmosphere-ocean interaction over the North Pacific*. Journal of Climate, Vol. 11, 2473-2481, doi: [10.1175/1520-0442\(1998\)011<2473:SOLSAO>2.0.CO;2](https://doi.org/10.1175/1520-0442(1998)011<2473:SOLSAO>2.0.CO;2)

INDEX

E

environment variable
 NCSTATDIR, 4
environment variable
 EXECDIR, 5, 6, 9
 FC, 4
 FLAGS, 4
 LBLAS, 4, 5
 LDLFLAGS, 4, 5
 MKL_NUM_THREADS, 14
 NCSTATDIR, 3
 NETCDF, 4, 5
 OMP_DYNAMIC, 13
 OMP_MAX_ACTIVE_LEVELS, 13
 OMP_NESTED, 13
 OMP_NUM_THREADS, 12, 14
 OMP_STACKSIZE, 13
 STATPACK, 4, 5
 TOPDIR, 4
EXECDIR, 5, 6, 9

F

FC, 4
FLAGS, 4

L

LBLAS, 4, 5
LDLFLAGS, 4, 5

M

MKL_NUM_THREADS, 14

N

NCSTATDIR, 4
NCSTATDIR, 3
NETCDF, 4, 5

O

OMP_DYNAMIC, 13
OMP_MAX_ACTIVE_LEVELS, 13
OMP_NESTED, 13
OMP_NUM_THREADS, 12, 14

OMP_STACKSIZE, 13

S

STATPACK, 4, 5

T

TOPDIR, 4