

IPSL Boot Camp Part 5: CDO and NCO

Sabine Radanovics, Jérôme Servonnat

March 24, 2016



Group exercise

Suppose...

We have

- 30 years climate model simulation
- 1 file per month, 6 hourly data
- netCDF format
- global data, regular lat/lon grid
- several variables, one of them is geopotential, on 16 standard pressure levels

We want

- 1 netCDF file with
- mean seasonal cycle (daily averages) of
- the thickness of the 850-500hPa layer ($hgt_{500} - hgt_{850}$)
- at each grid point in the region -15 to 25 deg East and 35 to 65 deg North

(850hPa and 500 hPa are standard pressure levels)

Tasks

1. Form groups of 3-4 people
2. Make a list of the necessary operations
3. Agree on a “speaker” who will report your results

Tools for Climate data and NetCDF files

CDO - Climate Data Operators

- Collection of operators for processing of climate and forecast model output
- simple statistical and arithmetic functions
- data selection and subsampling tools
- spatial interpolation
- Works for GRIB and netCDF datasets

Limitations

- GRIB: all time steps need to have the same variables and within a time step each variable may occur only once
- NetCDF: only 2-dimensional, 3-dimensional and 4-dimensional variables are supported
- Attribute conventions: GDT, COARDS or CF

How to use CDO?

- From the command line, in a shell script or in a system call inside a program in your favorite programming language
- select the operator(s) needed from the reference card
http://www.nco.ncep.noaa.gov/pmb/codes/nwprod/sorc/rtofs_cdo-1.4.0.1.fd/cdo-1.5.0/doc/cdo_refcard.pdf

start with a simple example: get some information about a dataset.

```
cdo info inputfile
```

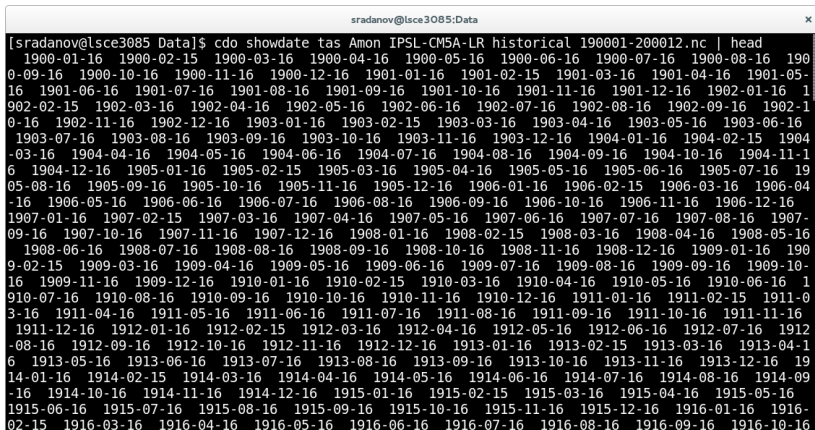
CDO info

```
sradanov@lsce3085:Data
[sradanov@lsce3085 Data]$ cdo info tas Amon IPSL-CM5A-LR historical 190001-200012.nc | head
-1 :      Date      Time      Level Gridsize      Miss :      Minimum      Mean      Maximum : Paramet
er ID
  1 : 1900-01-16 12:00:00      0      9216      0 :      219.78      273.99      304.16 : -1
  2 : 1900-02-15 00:00:00      0      9216      0 :      225.89      273.89      306.55 : -1
  3 : 1900-03-16 12:00:00      0      9216      0 :      215.67      274.11      303.21 : -1
  4 : 1900-04-16 00:00:00      0      9216      0 :      207.51      275.12      303.12 : -1
  5 : 1900-05-16 12:00:00      0      9216      0 :      206.28      276.85      305.00 : -1
  6 : 1900-06-16 00:00:00      0      9216      0 :      205.48      278.54      307.83 : -1
  7 : 1900-07-16 12:00:00      0      9216      0 :      205.72      278.75      309.69 : -1
  8 : 1900-08-16 12:00:00      0      9216      0 :      203.67      278.38      311.34 : -1
  9 : 1900-09-16 00:00:00      0      9216      0 :      203.84      276.87      306.25 : -1
[sradanov@lsce3085 Data]$
```

cdo showdate

How to display actual dates even with relative time axis?

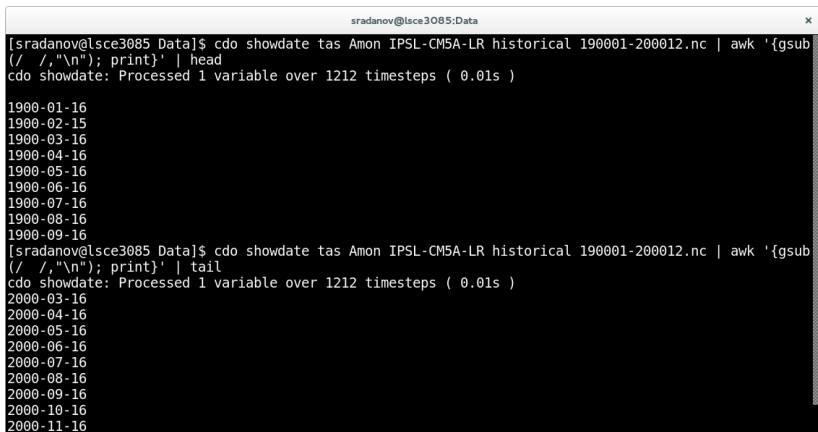
```
cdo showdate inputfile
```



```
sradanov@lsce3085:Data
[sradanov@lsce3085 Data]$ cdo showdate tas Amon IPSL-CMSA-LR historical 190001-200012.nc | head
1900-01-16 1900-02-15 1900-03-16 1900-04-16 1900-05-16 1900-06-16 1900-07-16 1900-08-16 190
0-09-16 1900-10-16 1900-11-16 1900-12-16 1901-01-16 1901-02-15 1901-03-16 1901-04-16 1901-05-
16 1901-06-16 1901-07-16 1901-08-16 1901-09-16 1901-10-16 1901-11-16 1901-12-16 1902-01-16 1
902-02-15 1902-03-16 1902-04-16 1902-05-16 1902-06-16 1902-07-16 1902-08-16 1902-09-16 1902-1
0-16 1902-11-16 1902-12-16 1903-01-16 1903-02-15 1903-03-16 1903-04-16 1903-05-16 1903-06-16
1903-07-16 1903-08-16 1903-09-16 1903-10-16 1903-11-16 1903-12-16 1904-01-16 1904-02-15 1904
-03-16 1904-04-16 1904-05-16 1904-06-16 1904-07-16 1904-08-16 1904-09-16 1904-10-16 1904-11-1
6 1904-12-16 1905-01-16 1905-02-15 1905-03-16 1905-04-16 1905-05-16 1905-06-16 1905-07-16 19
05-08-16 1905-09-16 1905-10-16 1905-11-16 1905-12-16 1906-01-16 1906-02-15 1906-03-16 1906-04
-16 1906-05-16 1906-06-16 1906-07-16 1906-08-16 1906-09-16 1906-10-16 1906-11-16 1906-12-16
1907-01-16 1907-02-15 1907-03-16 1907-04-16 1907-05-16 1907-06-16 1907-07-16 1907-08-16 1907-
09-16 1907-10-16 1907-11-16 1907-12-16 1908-01-16 1908-02-15 1908-03-16 1908-04-16 1908-05-16
1908-06-16 1908-07-16 1908-08-16 1908-09-16 1908-10-16 1908-11-16 1908-12-16 1909-01-16 190
9-02-15 1909-03-16 1909-04-16 1909-05-16 1909-06-16 1909-07-16 1909-08-16 1909-09-16 1909-10-
16 1909-11-16 1909-12-16 1910-01-16 1910-02-15 1910-03-16 1910-04-16 1910-05-16 1910-06-16 1
910-07-16 1910-08-16 1910-09-16 1910-10-16 1910-11-16 1910-12-16 1911-01-16 1911-02-15 1911-0
3-16 1911-04-16 1911-05-16 1911-06-16 1911-07-16 1911-08-16 1911-09-16 1911-10-16 1911-11-16
1911-12-16 1912-01-16 1912-02-15 1912-03-16 1912-04-16 1912-05-16 1912-06-16 1912-07-16 1912
-08-16 1912-09-16 1912-10-16 1912-11-16 1912-12-16 1913-01-16 1913-02-15 1913-03-16 1913-04-1
6 1913-05-16 1913-06-16 1913-07-16 1913-08-16 1913-09-16 1913-10-16 1913-11-16 1913-12-16 19
14-01-16 1914-02-15 1914-03-16 1914-04-16 1914-05-16 1914-06-16 1914-07-16 1914-08-16 1914-09
-16 1914-10-16 1914-11-16 1914-12-16 1915-01-16 1915-02-15 1915-03-16 1915-04-16 1915-05-16
1915-06-16 1915-07-16 1915-08-16 1915-09-16 1915-10-16 1915-11-16 1915-12-16 1916-01-16 1916-
02-15 1916-03-16 1916-04-16 1916-05-16 1916-06-16 1916-07-16 1916-08-16 1916-09-16 1916-10-16
```

cdo showdate

```
cdo showdate inputfile | awk '{gsub(/ /, "\n"); print}'
```



```
sradanov@lsce3085:Data
[sradanov@lsce3085 Data]$ cdo showdate tas Amon IPSL-CM5A-LR historical 190001-200012.nc | awk '{gsub(/ /, "\n"); print}' | head
cdo showdate: Processed 1 variable over 1212 timesteps ( 0.01s )
1900-01-16
1900-02-15
1900-03-16
1900-04-16
1900-05-16
1900-06-16
1900-07-16
1900-08-16
1900-09-16
[sradanov@lsce3085 Data]$ cdo showdate tas Amon IPSL-CM5A-LR historical 190001-200012.nc | awk '{gsub(/ /, "\n"); print}' | tail
cdo showdate: Processed 1 variable over 1212 timesteps ( 0.01s )
2000-03-16
2000-04-16
2000-05-16
2000-06-16
2000-07-16
2000-08-16
2000-09-16
2000-10-16
2000-11-16
```

cdo splitseas

How to split a file into four files (one for each season)?

```
cdo splitseas inputfile outputfileprefix
```

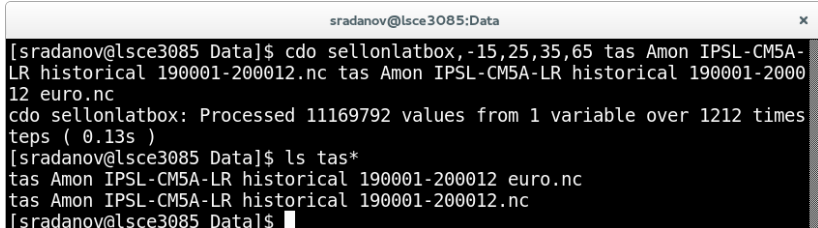
```
sradanov@lsce3085:Data
[sradanov@lsce3085 Data]$ ls tas*
tas Amon IPSL-CM5A-LR historical 190001-200012.nc
[sradanov@lsce3085 Data]$ cdo splitseas tas Amon IPSL-CM5A-LR historical 19
0001-200012.nc tas Amon IPSL-CM5A-LR historical 190001-200012
cdo splitseas: Processed 11169792 values from 1 variable over 1212 timestep
s ( 0.22s )
[sradanov@lsce3085 Data]$ ls tas*
tas Amon IPSL-CM5A-LR historical 190001-200012 DJF.nc
tas Amon IPSL-CM5A-LR historical 190001-200012 JJA.nc
tas Amon IPSL-CM5A-LR historical 190001-200012 MAM.nc
tas Amon IPSL-CM5A-LR historical 190001-200012.nc
tas Amon IPSL-CM5A-LR historical 190001-200012 SON.nc
[sradanov@lsce3085 Data]$
```


cdo sellonlatbox

How to select a region?

```
cdo sellonlatbox,lon1,lon2,lat1,lat2 inputfile outputfile
```

This time the operator needs arguments. Operator arguments are separated by commas **without** spaces.



```
sradanov@lsce3085:Data
[sradanov@lsce3085 Data]$ cdo sellonlatbox,-15,25,35,65 tas Amon IPSL-CM5A-LR historical 190001-200012.nc tas Amon IPSL-CM5A-LR historical 190001-200012 euro.nc
cdo sellonlatbox: Processed 11169792 values from 1 variable over 1212 times
teps ( 0.13s )
[sradanov@lsce3085 Data]$ ls tas*
tas Amon IPSL-CM5A-LR historical 190001-200012 euro.nc
tas Amon IPSL-CM5A-LR historical 190001-200012.nc
[sradanov@lsce3085 Data]$
```

Options

cdo has a number of options such as

- a silent mode `-s`
- writing output with relative (`-r`) or absolute (`-a`) time axis
- writing output in a specific format (`-fnc4`, `-fgrb`, ...)

The options are placed before the operator

```
cdo -s ymonmean inputfile outputfile
```

Interpolation

Regridding

```
cdo remapbil,grid inputfile outputfile
```

grid

There are several possibilities to define *grid*.

1. Take the grid from another file.

```
cdo remapbil,obs.nc model.nc model_obsgrid.nc
```

2. Use a predefined grid

```
cdo remapbil,r360x180 model.nc model_1x1deggrid.nc
```

3. cdo grid description
4. SCRIP grid
5. PINGO grid

Combine operators

Why?

- less temporary output files
- faster because things can be done in parallel and with less I/O.

How?

```
cdo operator1 -operator2 inputfile(s) outputfile
```

Example

```
cdo cat -selvar, hgt hgt_NA_* hgt_NA_X.nc
```

Combine operators

Example operations

- concatenate monthly files (`cdo cat inputfiles outputfile`)
- select region (`cdo sellonlatbox,-15,25,35,65 inputfile outputfile`)
- select geopotential variable (`cdo selvar,hgt inputfile outputfile`)
- select two levels (`cdo sellevel,500 inputfile outputfile`)
- subtract the values for the two levels (`cdo sub inputfile1 inputfile2 outputfile`)
- calculate multi-year daily averages (`cdo ydaymean inputfile outputfile`)

NCO - netCDF Operators

- NCO is a suite of programs known as operators
- Each operator is a standalone, command line program executed at the shell-level.
- The operators take netCDF files as input, perform an operation and produce a netCDF file as output.
- Operators are primarily designed to aid manipulation and analysis of data
- Operators are as general as netCDF itself.
- NCO was written to consume the absolute minimum amount of system memory required to perform a given job.

NCO user guide: <http://nco.sourceforge.net/nco.pdf>

Examples for processing CMIP5 data with nco:

http://nco.sourceforge.net/xmp_cesm.html

Concatenators

nco has two operators to join files:

- nccat joins record variables along the record dimension (similar to cdo cat)
- nccat joins variables along a new dimension, that will be the record dimension. If there is already a record dimension it will be turned into a fixed length dimension. The corresponding variables (or hyperslabs) from each file need to have the same dimensions and size.

```
nccat -u ensemble ifiles_* outfile.nc
```

with -u you can give a name to the new record dimension, that will be called "record" otherwise.

How to concatenate along any dimension?

1. turn the desired dimension into the record dimension using `ncpdq`
2. concatenate with `ncrcat`
3. maybe reorder dimensions again with `ncpdq`

`ncpdq`

- can pack or unpack data
- can re-order dimension or
- can reverse dimensions (`cdo invertlat` and `cdo invertlev` are special cases)

```
ncpdq -a lon,-lat,time infile outfile
```

`-a` option is for arranging dimensions in the order listed in the argument. `-dim` reverses the dimension `dim`.

Specifying input files

Most nco operators accept several ways to specify the input files:

1. Type them explicitly in the command line
2. Specify multiple files using unix wild cards

```
ncrcat infileprefix*  
ncrcat infile????.nc  
ncrcat *.nc
```

3. Specify a (remote) directory using the `-p` option. All the input files are assumed to be in the specified directory. This has no influence on where the output file is written.

```
ncrcat -p /Myhugedisc/Model/Favoritmodel/  
infile1.nc infile2.nc
```

4. Specify multiple files using the `-n` option.

Specifying input files

-n option

The -n option can be used if the inputfiles are of the form
constant alphanumeric prefix fixed length numeric suffix .nc
for example: model_x_version_4_year_1850.nc

The -n option takes 3 arguments:

1. The number of files
2. The number of digits of the numeric suffix
3. The increment

Then one has to give the first file name and the others are created using the information from the arguments. Example:

```
ncrcat -n 6,4,10 model_x_version_4_year_1850.nc
```

will take the files model_x_version_4_year_1850.nc,
model_x_version_4_year_1860.nc, model_x_version_4_year_1870.nc,
model_x_version_4_year_1880.nc, model_x_version_4_year_1890.nc
and model_x_version_4_year_1900.nc as input.

Output options

```
ncks -O ...
```

will overwrite existing files without asking

```
ncks -A ...
```

will try to append the output to an existing output file.

Example

Union of two files (similar to `cdo merge`)

```
ncks -A file1 file2
```

Limitations:

- The record dimension (unlimited dimension) has to have the same name in both files
- Non-record dimensions with the same name in both files must have the same size

ncks - netCDF Kitchen Sink: Extract and merge

-v option, (-C, -c)

Select variables

Record variables to operate on can be selected using -v followed by a list of variables. The coordinate variables corresponding to the dimensions of the chosen variables are written to the output as well. This feature can be turned off using the -C option. With -c on the other hand ALL coordinate variables are written to the output.

Example:

```
ncks -v T,Rhum,U,V infile outfile
```

Whatever the variables in the infile were, the operator will process T, Rhum, U, V. The processed variables and the coordinate variables corresponding to the dimensions of these variables will be written to the outfile.

Selection, subsetting, hyperslabs

-x option

With the -x option all the variables are selected **except** those listed as arguments of -v.

```
ncks -x -v T,Rhum,U,V infile outfile
```

The operator will now process whatever variable was in the infile **except** T, Rhum, U and V.

Selection, subsetting, hyperslabs

-d option: subsetting variables along dimensions - hyperslabs

```
ncks -d dim,min[,max[,stride]] infile outfile
```

The -d option takes 2, 3 or 4 arguments:

1. The dimension name
2. The minimum index or value
3. The maximum index or value
4. A stride

Selection, subsetting, hyperslabs

-d option: subsetting variables along dimensions - hyperslabs

Specifying min and max, -F option

- If min and max are integers, they are interpreted as dimension indices.
- As a default nco uses c-style index conventions, that is starts counting with 0. This can be changed to fortran-style conventions (start counting with 1) adding the -F option.
- Leaving min (max) empty, is interpreted as the first (last) index or value. (The comma is not omitted.)

Example: Specify the first 5 indices of dimension time

```
ncks -d time,0,4 infile outfile
```

```
ncks -F -d time,1,5 infile outfile
```

```
ncks -F -d time,,5 infile outfile
```

Selection, subsetting, hyperslabs

-d option: subsetting variables along dimensions - hyperslabs

Specifying min and max

- Negative indices are counted from the last index (as in python), that is -1 corresponds to the last but first index.
- If min and max are floats (numbers with a decimal point), they are interpreted as dimension values.
- $\text{min} \leq \text{max}$, even if the values are stored in inversed order, unless you have wrapped coordinates (such as longitude) and your operator is ncks and you want to specify for example a hyperslab from 320. to 40. degree.

```
ncks -d lon,40.,320. infile outfile # will work
(selects everything inside the interval 40., 320.)
ncks -d lon,320.,40. infile outfile # will work
(selects everything outside the interval 40., 320.)
ncrcat -d lon,320.,40. infile outfile # won't work
(wrapped coordinates are only supported with ncks)
```


Selection, subsetting, hyperslabs

-d option: subsetting variables along dimensions - hyperslabs

Cross sections

A cross section can be selected by specifying only a dimension name and a minimum index or value without the trailing comma.

Example selecting the 6th ensemble member:

```
ncks -F -d ensemble,6 infile outfile
```

Stride

- Stride must be an integer following the third comma in the -d argument list.
- If stride is not specified the third comma has to be omitted as well.

Example: Select every 12th time step starting from the 3rd one.

```
ncks -F -d time,3,,12 infile outfile
```

Selection, subsetting, hyperslabs

-d option: subsetting variables along dimensions - hyperslabs

Multislabs

It is possible to specify multiple hyperslabs in one call for the same dimension and for different dimensions.

```
ncra -F -d time,3,,12 -d lat,-60.,-40. -d lat,40.,60.  
-v Tas infile outfile
```

Averagers

ncra - record averager

Averages record variables along the record dimension.

```
ncra -d lat,40.,60. -d lon,-15.,35. infile(s) outfile
```

Options can be used to select variables (-v, -x) and to define hyperslabs (-d).

ncea - ensemble averager

Averages (pointwise) variables with the same name from different files. Coordinate variables are not averaged.

```
ncea -n 50,2,1 -v T,Rhum ifile01.nc outfile.nc
```

Averagers

ncwa - weighted averager

- Average variables in a single file over arbitrary dimensions, specified with the -a option. Without the -a option, variables are averaged over all their dimensions resulting in a scalar value for each of them. Coordinate variables are averaged as well.
- Weights can be specified with the -w option
- A mask can be specified with the -B option
- ncwa can computed (weighted) sums as well if the -N option is used.

Averagers

ncwa - examples

Zonal mean

```
ncwa -a lon infile outfile
```

Weighted meridional mean

```
ncwa -a lat -w weightvar infile outfile
```

weightvar is a variable in the file containing the weights.

area average where the variable *ORO* > 0.5

```
ncwa -B 'ORO < 0.5' -a lat,lon infile outfile
```

ncap2 - Arithmetics

- Specify command line operations with `-s`
- Specify the operations in a separate file with `-S`

```
ncap2 -s 'latw=cos(lat)' infile outfile
```

The cosine of the variable *lat* in the infile is calculated and stored in a new variable named *latw*. All variables in infile plus the new variable *latw* are written to outfile.

- Several `-s` strings are accepted
- Several operations in a single string are possible if separated by semicolons.

```
ncap2 -v -s 'const=7.;latw=cos(lat);npq=latw*const'  
infile outfile
```

- `-v` means that only the new variables are written to the output file.

ncatted

Sometimes you need to edit attributes, or you might just want to add some.

```
ncatted -a att_name,var_name,mode,att_type,att_value
```

- att_name: name of the attribute
- var_name: the name of the variable the attribute belongs to or “global” for a global attribute
- mode: one of a (append), c (create), d (delete), m (modify), o (overwrite)
- att_type: one of f (float), d (double), l (long), s (short), c (character), b (byte)
- att_value: the value of the attribute

```
cdo subc,273.15 tas_obs.nc tas_obs_degC.nc  
ncatted -a units,tas,o,c,degC tos_obs_degC.nc
```

ncrename

rename attributes

```
ncrename -a oldname,newname infile outfile
```

rename variables

```
ncrename -v oldname,newname infile outfile
```

rename dimensions

```
ncrename -d oldname,newname infile outfile
```

The -a, -v and -d options can be specified more than once.

Packing vs. compression

Packing

Packing is the NetCDF3 way of reducing the file size. Record variables are stored as short integers along with the `scale_factor` and `add_offset` attributes. It is a lossy algorithm.

Get rid of 'short' variables

```
ncpdq -P unpack infile outfile
```

Compression

NetCDF4 allows compression using a loss-less algorithm. With any `nco` operator you can specify the output format, for example `-4` for NetCDF4 and a compression level with the `-L` option followed by a number between 0 (no compression) and 9 (maximum compression) - the compression level.

```
ncks -4 -L 5 infile.nc outfile.nc
```