Tutorial 2: File format and nco commands

My File standard:

Most of the files stored on geog1 and geog2 servers are in netcdf format (binary format)

The extension is generally like: file.nc

This is my standard, generally:

VAR_1?_DATEINI_DATEFIN_EXPID.nc

VAR is the variable, using the IPCC standard:

http://www-pcmdi.llnl.gov/ipcc/standard_output.html

pr Rainfall

tas: 2m Temperature, tasmin, minimum Temp, tasmax, maximum temp

psl: mean sea level pressure

ts: surface skin temperature (SST generally)

ua, va: zonal, meridional wind

zg: geopotential (Z)

hus: specific humidity

etc etc

1? Is the time step

1m means monthly means

1d means daily means

1p means pentad means (5days average)

DATEINI, DATEFIN: initial date and final date

If monthly then this can be like 194801_200512 (which means monthly means for Jan 1948 to Dec 2005)

If daily this can be 19480101_20051231 (from the 1st of Jan 1948 to the 31th of Dec 2005)

EXPID

The dataset expid

Example: NCEP means NCEP reanalysis, CRUTS2.0, Climate research unit high resolution dataset, this can be the name of a model as well

Where to find the files:

Everything (almost) is archived under /data6/caminade (at the moment maybe this will move to /data7/caminade soon for space reasons)

The **observation** datasets are under:

/home/caminade/Obs

Each expid corresponds to a specific dataset (example NCEP means NCEP-NCAR reanalysis)

(Note that home/caminade/Obs is a symbolic link going straight forward to /data6/caminade/Obs)

The **simulation** datasets are under:

/home/caminade/Sim

Each expid corresponds to a specific project / type of simulations

Examples that can be usefull:

/home/caminade/Sim/ENSEMBLES/stream2/monthly

monthly means for ENSEMBLES stream2 seasonal forecasts

/home/caminade/Sim/ENSEMBLES/RCM/monthly/CTL

monthly means for ENSEMBLES RCM simulations driven by the ERA40 reanalysis (control experiment or CTL)

You can browse these directories and see what's available

NetCDF - Network Common Data Format

Online documentation for NetCDF is available at

http://www.unidata.ucar.edu/packages/netcdf/index.html.

Most of the files stored on geog1 and geog2 servers are in netcdf format (binary format)

The extension is generally like: file.nc

(Note that in Linux to know the format of files you can type **file file.nc** (the file command gives you the type of data you're looking at)

The command ncdump

NAME

ncdump − Convert netCDF file to text form (CDL)

SYNOPSIS

ncdump [-c] [-h] [-v *var1,...*] [-b *lang*] [-f *lang*] [-l *len*] [-n *name*] [-p *f_digits[,d_digits]*] [-k] [-x] [-s] [-t] *file*

DESCRIPTION

ncdump generates a text representation of a specified netCDF file on standard output. The text representation is in a form called CDL (‘‘network Common Data form Language’’) that can be viewed, edited, or serve as input to **ncgen**. **ncgen** is a companion program that can generate a binary netCDF file from a CDL file. Hence **ncgen** and **ncdump** can be used as inverses to transform the data representation between binary and text representations. See **ncgen** for a description of CDL and netCDF representations.

As of NetCDF version 4.1, and if DAP support was enabled when **ncdump** was built, the file name may specify a DAP URL. This allows **ncdump** to print out data sources from DAP servers. When used with the **-h** option, **ncdump** can be used to show the translation from the DAP DDS data model to the NetCDF data model.

ncdump defines a default display format used for each type of netCDF data, but this can be changed if a ‘C_format’ attribute is defined for a netCDF variable. In this case, **ncdump** will use the ‘C_format’ attribute to format each value. For example, if floating-point data for the netCDF variable ‘Z’ is known to be accurate to only three significant digits, it would be appropriate to use the variable attribute

```
Z:C_format = "%.3g"
```

ncdump may also be used as a simple browser for netCDF data files, to display the dimension names and sizes; variable names, types, and shapes; attribute names and values; and optionally, the values of data for all variables or selected variables in a netCDF file.

ncdump uses ‘_’ to represent data values that are equal to the ‘_FillValue’ attribute for a variable, intended to represent data that has not yet been written. If a variable has no ‘_FillValue’ attribute, the default fill value for the variable type is used if the variable is not of byte type.

ncdump may also be used to determine what kind of netCDF file is used (which variant of the netCDF file format) with the -k option.

More details: http://www.unidata.ucar.edu/software/netcdf/docs/ncdump-man-1.html

Examples:

Copy a netcdf file in your home directory for example type:

Under your home directory (home/Macleod) create a data dir:

mkdir data

cd data

cp /home/caminade/Obs/NCEP/psl_1m_194801_200801_NCEP.nc /home/macleod/data

the command:

ncdump –h psl_1m_194801_200801_NCEP.nc

displays the header of the file, this must look like that:

```
netcdf psl_1m_194801_200801_NCEP {
dimensions:
        lon = 144 ;
        lat = 73 ;
        time = UNLIMITED ; // (721 currently)
variables:
        float lat(lat) ;
                lat:units = "degrees_north" ;
                lat:actual_range = 90.f, -90.f ;
                lat:long_name = "Latitude" ;
        float lon(lon) ;
                lon:units = "degrees_east" ;
                lon:long_name = "Longitude" ;
                lon:actual_range = 0.f, 357.5f ;
        double time(time) ;
                time:units = "hours since 1-1-1 00:00:0.0" ;
                time:long_name = "Time" ;
                time:actual_range = 17067072., 17593032. ;
                time:delta_t = "0000-01-00 00:00:00" ;
                time:prev_avg_period = "0000-00-01 00:00:00" ;
```

```
	float psl(time, lat, lon) ;
		psl:long_name = "Sea Level Pressure" ;
		psl:valid_range = 870.f, 1150.f ;
		psl:actual_range = 962.4105f, 1082.558f ;
		psl:units = "millibars" ;
		psl:add_offset = 0.f ;
		psl:scale_factor = 1.f ;
		psl:missing_value = -9.96921e+36f ;
		psl:precision = 1s ;
		psl:least_significant_digit = 1s ;
		psl:var_desc = "Sea Level Pressure\n",
			"P" ;
		psl:dataset = "CDC Derived NCEP Reanalysis Products\n",
			"AC" ;
		psl:level_desc = "Sea Level\n",
			"I" ;
		psl:statistic = "Mean\n",
			"M" ;
		psl:parent_stat = "Other\n",
			"-" ;

// global attributes:
		:title = "monthly mean slp from the NCEP Reanalysis" ;
		:history = "Tue Mar 18 11:09:13 2008: ncrename -v slp,psl
psl_1m_194801_200801_NCEP.nc\n",
			"Thu May  4 18:12:35 2000: ncrcat -d time,0,622
/Datasets/ncep.reanalysis.derived/surface/slp.mon.mean.nc ./surface/slp.mon.mean.nc\n",
                    "Mon Jul  5 23:22:35 1999: ncrcat slp.mon.mean.nc
/Datasets/ncep.reanalysis.derived/surface/slp.mon.mean.nc
/dm/dmwork/nmc.rean.ingest/combinedMMs/slp.mon.mean.nc\n",

                    "/home/hoop/crdc/cpreanjuke2farm/cpreanjuke2farm Thu Oct 26 23:42:16
1995 from pre.sig995.85.nc\n",

                    "created 95/02/06 by Hoop (netCDF2.3)" ;

            :description = "Data is from NMC initialized reanalysis\n",

                    "(4x/day).  These are the 0.9950 sigma level values." ;

            :platform = "Model" ;

            :Conventions = "COARDS" ;

}
```

To summarize different informations are available:

1) **The dimensions: here time, lat, lon**

Typically gives you an indication about the number of time steps (here 721 values, monthly means from Jan 1948 to Jan 2008), the number of latitude (73) and longitude (144) points.

2) **The variables: lat, lon, time, psl**

lat(lat) means that latitude is a 1D vector of dimension lat (73 points)

lon(lon) means that longitude is a 1D vector of dimension lon (144 points)

psl(time,lat,lon) is a 3D matrix with dimensions (time, lat, lon) thus (73x144x721)

3) **The attributes:**

For a given variable different attributes are available. They provide extra information about the dataset (kind of metadatas).

Example: psl:long_name = "Sea Level Pressure" ; (this way you know what the variable is, here mean sea level pressure for NCEP reanalysis).

If you type:

ncdump psl_1m_194801_200801_NCEP.nc

This will print the whole data in the terminal (this can be long to display)

ncdump –v lat psl_1m_194801_200801_NCEP.nc

will print the values of the variable (-v) lat on the screen here you will see:

the header+

```
lat = 90, 87.5, 85, 82.5, 80, 77.5, 75, 72.5, 70, 67.5, 65, 62.5, 60, 57.5,
    55, 52.5, 50, 47.5, 45, 42.5, 40, 37.5, 35, 32.5, 30, 27.5, 25, 22.5, 20,
    17.5, 15, 12.5, 10, 7.5, 5, 2.5, 0, -2.5, -5, -7.5, -10, -12.5, -15,
    -17.5, -20, -22.5, -25, -27.5, -30, -32.5, -35, -37.5, -40, -42.5, -45,
    -47.5, -50, -52.5, -55, -57.5, -60, -62.5, -65, -67.5, -70, -72.5, -75,
    -77.5, -80, -82.5, -85, -87.5, -90 ;
```

So you can see that the grid ranges form 90N to 90S with a regular 2.5 degree step

Type man ncdump to see all available options (press space bar to scroll down, j to browse a line down, k to browse a line up, this works as well with the tool more)

The command ncgen

NAME

ncgen − From a CDL file generate a netCDF file, a C program, or a Fortran program

SYNOPSIS

ncgen [-b] [-c] [-f] [-k *kind_of_file*] [-x] [-n] [-o *netcdf_filename*] *input_file*

DESCRIPTION

ncgen generates either a netCDF file, or C or Fortran source code to create a netCDF file. The input to **ncgen** is a description of a netCDF file in a small language known as CDL (network Common Data form Language), described below. If no options are specified in invoking **ncgen**, it merely checks the syntax of the input CDL file, producing error messages for any violations of CDL syntax. Other options can be used to create the corresponding netCDF file, to generate a C program that uses the netCDF C interface to create the netCDF file, or to generate a Fortran program that uses the netCDF Fortran interface to create the same netCDF file.

ncgen may be used with the companion program **ncdump** to perform some simple operations on netCDF files. For example, to rename a dimension in a netCDF file, use **ncdump** to get a CDL version of the netCDF file, edit the CDL file to change the name of the dimensions, and use **ncgen** to generate the corresponding netCDF file from the edited CDL file.

OPTIONS

-b Create a (binary) netCDF file. If the **-o** option is absent, a default file name will be constructed from the netCDF name (specified after the **netcdf** keyword in the input) by appending the ‘.nc’ extension. If a file already exists with the specified name, it will be overwritten.

-c Generate **C** source code that will create a netCDF file matching the netCDF specification. The C source code is written to standard output.

-f Generate **Fortran** source code that will create a netCDF file matching the netCDF specification. The Fortran source code is written to standard output.

-o netcdf_file
Name for the binary netCDF file created. If this option is specified, it implies the "**-b**" option. (This option is necessary because netCDF files cannot be written directly to standard output, since standard output is not seekable.)

-k kind_of_file
Using -k2 or -k "64-bit offset" specifies that generated file (or program) should use version 2 of format that employs 64-bit file offsets. The default is to use version 1 ("classic") format with 32-bit file offsets, although this limits the size of the netCDF file, variables, and records to the sizes supported by the classic format. (NetCDF-4 will support additional kinds of netCDF files, "netCDF-4" and "netCDF-4 classic model".) Note: -v is also accepted to mean the same thing as -k for backward compatibility, but -k is preferred, to match the corresponding ncdump option.

-x Don’t initialize data with fill values. This can speed up creation of large netCDF files greatly, but later attempts to read unwritten data from the generated file will not be easily detectable.

This create a netdcf file from an ascii file. The header needs to be properly written as the necdf standard (see example below)

Example / exercice:

Convert netcdf file to ascii type:

ncdump psl_1m_194801_200801_NCEP.nc > psl_1m_194801_200801_NCEP.asc

redirect the output of ncdump to the ascii file *psl_1m_194801_200801_NCEP.asc (see Linux tutorial)*

Wait a bit, and look the difference of file sizes between ASCII and netcdf, comments?

Open the ascii file with a text editor:

nedit psl_1m_194801_200801_NCEP.asc &

delete few attribute lines like:

psl:units = "millibars" ;

psl:add_offset = 0.f ;

psl:scale_factor = 1.f ;

psl:missing_value = -9.96921e+36f ;

psl:precision = 1s ;

psl:least_significant_digit = 1s ;

then save (CTRL+S)

now type

ncgen -o test.nc psl_1m_194801_200801_NCEP.asc

and then look at the new test.nc file by typing:

ncdump –h test.nc

What happened?

NCO (NetCdf operator):

What is NCO?

The netCDF Operators, or NCO, are a suite of programs known as **operators**. They are standalone, command-line programs like, e.g., `ls` or `mkdir`. Each operator takes netCDF files as input, then operates (e.g., derives new data, averages, hyperslabs, manipulates metadata) and produces a netCDF output file. NCO primarily aids manipulation and analysis of gridded scientific data. The single-command style of NCO allows users to manipulate and analyze files interactively, with simple scripts that avoid some overhead (and power) of higher level programming environments. The NCO User's Guide illustrates their use with examples from the field of climate modeling and analysis. Note that the "averagers" are misnamed because they perform many non-linear operations as well, e.g., total, minimum, maximum, RMS:

- `ncap2` netCDF Arithmetic Processor (examples)
- `ncatted` netCDF Attribute Editor (examples)
- `ncbo` netCDF Binary Operator (includes `ncadd`, `ncsubtract`, `ncmultiply`, `ncdivide`) (examples)
- `ncea` netCDF Ensemble Averager (examples)
- `ncecat` netCDF Ensemble Concatenator (examples)
- `ncflint` netCDF File Interpolator (examples)
- `ncks` netCDF Kitchen Sink (examples)
- `ncpdq` netCDF Permute Dimensions Quickly, Pack Data Quietly (examples)
- `ncra` netCDF Record Averager (examples)
- `ncrcat` netCDF Record Concatenator (examples)
- `ncrename` netCDF Renamer (examples)
- `ncwa` netCDF Weighted Averager (examples)

The operators are as general as netCDF itself: there are no restrictions on the contents of the netCDF file(s) used as input. NCO's internal routines are completely dynamic and impose no limit on the number or sizes of dimensions, variables, and files. NCO is designed to be used both interactively and with large batch jobs. The default operator behavior is often sufficient for everyday needs, and there are numerous command line (i.e., run-time) options, for special cases. NCO works well on most modern operating systems, including: Apple OS X, *BSD, Cray UNICOS, DEC Tru64, IBM AIX, HPUX, Linux, Microsoft Windows, NEC Super UX, SGI IRIX, and Sun Solaris.

The website:

http://nco.sourceforge.net/

Examples / exercices:

If you type man the_operator you will see all the available options type:

man ncea (q to quit, space to browse doan or j and k)

What does this operator perform?

Perform the JAS seasonal mean for *psl_1m_194801_200801_NCEP.nc* using ncks and ncea

ncks –F –d time,7,721,12 psl_1m_194801_200801_NCEP.nc jul.nc

ncks –F –d time,8,721,12 psl_1m_194801_200801_NCEP.nc aug.nc

ncks –F –d time,9,721,12 psl_1m_194801_200801_NCEP.nc sep.nc

ncea juil.nc aug.nc sep.nc JAS.nc

rm juil.nc aug.nc sep.nc (cleaning)

then *ncdump –h JAS.nc*

What happened in the files?

The option –F says you use fortran standard (starting at 1) so here you extract the July time step (7) until the end (721) per stride of 12 months (each year)

Then you do the same for August and September, and then average all files with ncea

Do the same for DJF (be carrefull there is a trap here...)

MAM and SON, then clean the monthly files.

Try *ncrename –v psl,sealevpress psl_1m_194801_200801_NCEP.nc toto.nc*

Then type *ncdump –h toto.nc*

What happened?

Do ncks –F –d time,1,224 *psl_1m_194801_200801_NCEP.nc block1.nc*

ncks –F –d time,225,721 *psl_1m_194801_200801_NCEP.nc block2.nc*

then ncrcat block1.nc block2.nc big_block.nc

What did you do?

The most usefull commands:

ncks, ncea, ncwa, ncrcat (cat files).

Browse the different examples on the website (see above) and play with *psl_1m_194801_200801_NCEP.nc in changing the attributes, renaming the dimensions, the variables and doing math operation with ncap2*