

Day 3: Diagnostics and Output

Adam Phillips

asphilli@ucar.edu

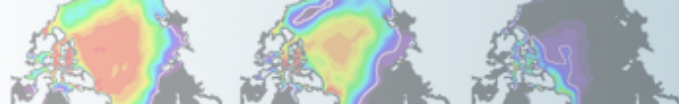
Climate Variability and Change Working Group Liaison
CGD/NCAR

Thanks to Dennis Shea, Christine Shields, Dave Bailey,
Susan Bates, and Ernesto Munoz Acevedo for their assistance



U.S. DEPARTMENT OF
ENERGY

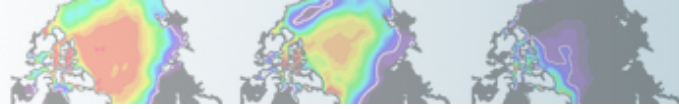
Office of
Science



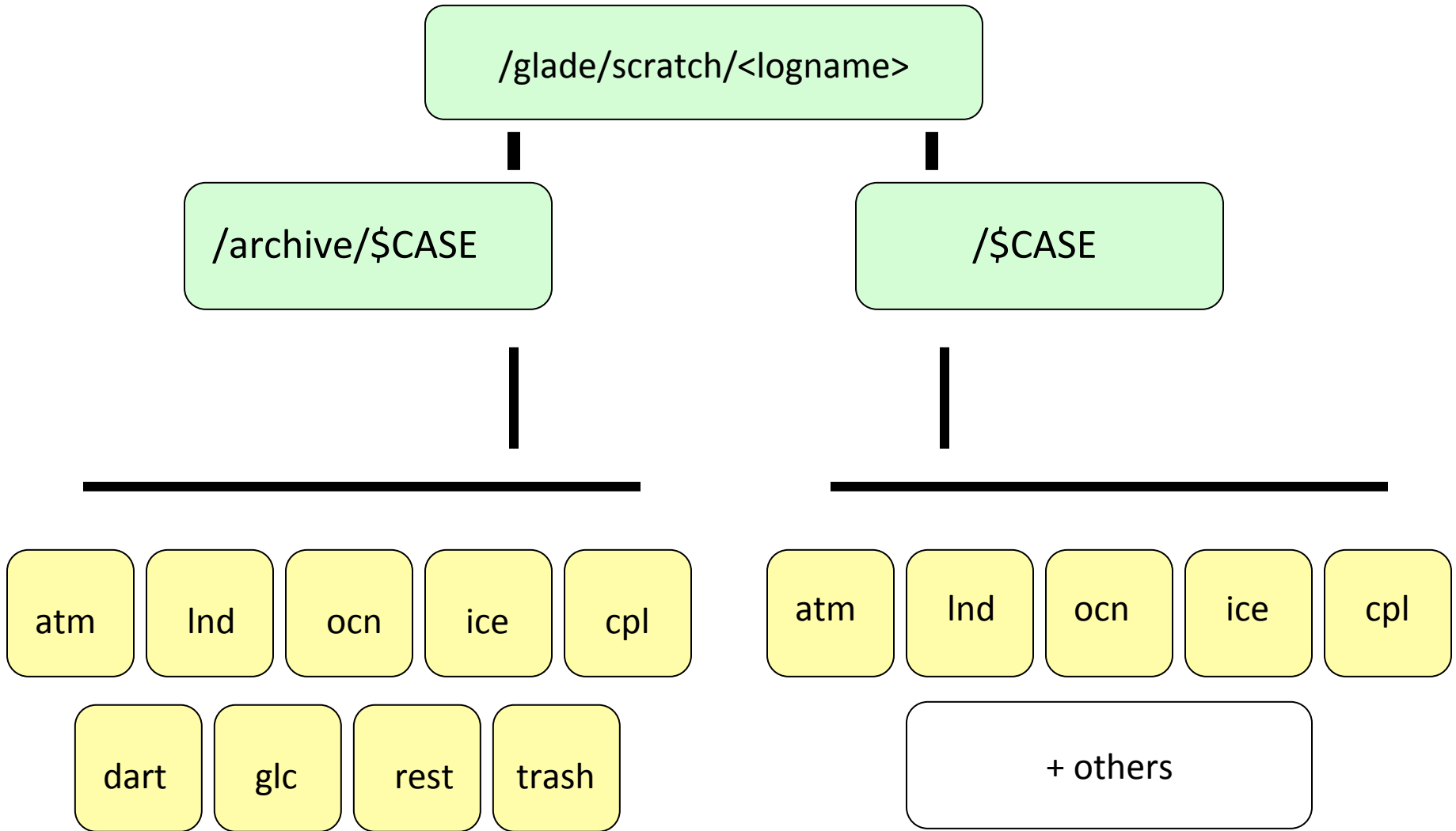
Day 3: Diagnostics and Output

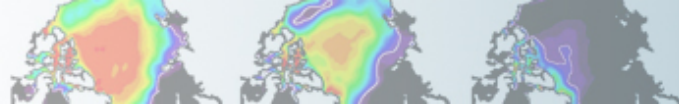
Outline

- I. CESM1.0 directory structures & file-naming conventions
- II. Introduction to the netCDF format, ncdump
- III. netCDF Operators (NCO) / Climate Data Operators (CDO) / ncview
- IV. ImageMagick / ghostview
- V. Introduction to NCL
- VI. Practical Lab #3
 - A. Diagnostics packages
 - B. NCL post-processing scripts
 - C. NCL graphics scripts
 - D. Additional Exercises
 - E. Challenges
- VII. Appendix (NCAR Archival System: The HPSS)

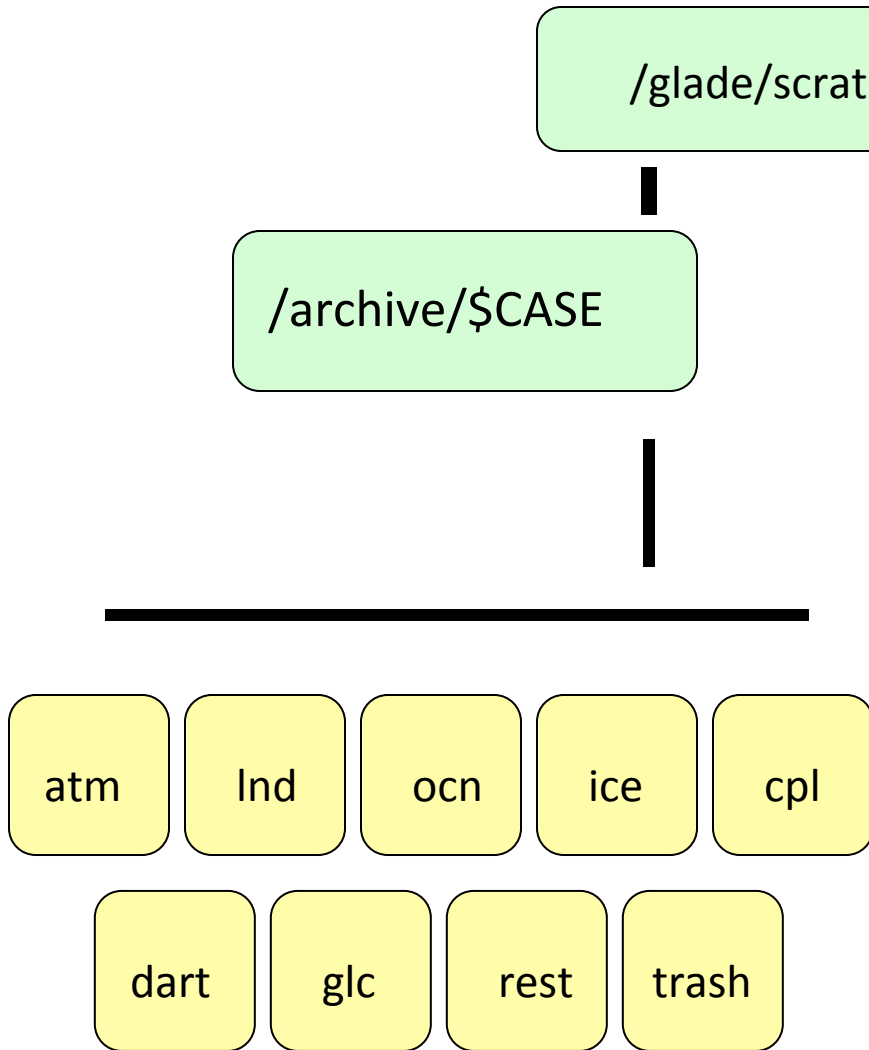


Short-Term Archive and Runtime Directories





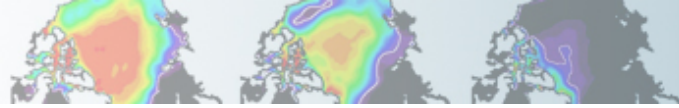
Short-Term Archive



- By default, short term archiver writes to `/glade/scratch/<logname>/archive/$CASE` on bluefire. (Can modify by setting `DOUT_S_MSROOT` in `env_run.xml`)

- Long term archiver is activated when the `env_run.xml` parameter `$DOUT_L_MS = "TRUE"`.

- When `$DOUT_L_MS = "TRUE"`, all files are written from short-term archive directory to the HPSS. Once confirmed that the files are present on the HPSS, *all but the newest files are removed from the short-term archive.*



CESM History File Naming Conventions

All history output files are in “netcdf” format

Location of history files in short-term archive directory:

```
/glade/scratch/<logname>/archive/$case/<component>/hist
```

component = atm, ocn, etc.

CESM distinguishes between different time sampling frequencies by creating distinct history files for each frequency. *Sampling frequencies are set by the user within the namelist.*

Example history file names:

```
f40_test.cam2.h0.1993-11.nc
```

```
f40_test.clm2.h0.1993-11.nc
```

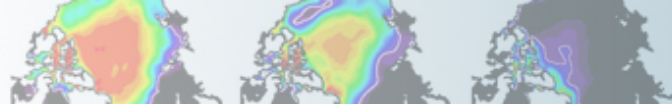
```
f40_test.pop.h.1993-11.nc
```

```
f40_test.cice.h.1993-11.nc
```

By default, h0/h denotes that the time sampling frequency is monthly.

Other frequencies are saved under the h1, h2, etc file names:

```
f40_test.cam2.h1.1993-11-02-00000.nc
```



CESM Diagnostics/Experiments Page

Case Name: b40.1850.track1.1deg.006a
Machine: NCAR:bluefire
CMIP5 ID: ---
Compset: B_1850_CN
Resolution: 0.9x1.25_gx1v6
Years: 953-1108
Initialization: 0953-01-01 of b40.1850.track1.1deg.006
HPSS Location: /CCSM/csm
 /b40.1850.track1.1deg.006a
Case Details: MOAR control simulation. Extra output saved.
Start/End Dates: 12/29/10, 2/2/11
Data Release Date (Full): 7/15/11

CESM 1.0 EXPERIMENTS / DIAGNOSTICS

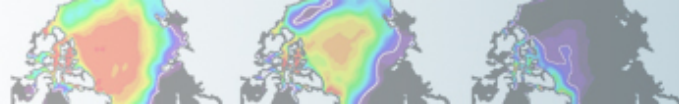
Stand-Alone Diagnostics
 CAM4.0
 CAM5.0
 CLM4.0
 CICE4.0
 POP2

J. Climate Special Issue Collection
 CCSM4

CONTROL SIMULATIONS

Brief Description	Case Details	Diagnostics				Length of Run Diagnostics		
		Atm	Ice	Land	Ocean	CCR	Ocean Timeseries	
CCSM4 1° Pre-Industrial Control Case Name: b40.1850.track1.1deg.006 Data Location: ESG	[Details]	863-892 w/observations	Atm	Ice	Land	Ocean	CCR	Ocean Timeseries
		863-882 - CCSM3 T85 Pre-Industrial Control	Atm	Ice	Land	Ocean		
CCSM4 1° Pre-Industrial Control (MOAR) Case Name: b40.1850.track1.1deg.006a Data Location: ESG	[Details]	1050-1079 w/observations	Atm	Ice	Land	Ocean	---	Ocean Timeseries
CCSM4 2° Pre-Industrial Control Case Name: b40.1850.track1.2deg.003 Data Location: ESG	[Details]	501-530 w/observations	Atm	Ice	Land	Ocean	CCR	Ocean Timeseries
		501-520 - CCSM3 T42 Pre-Industrial Control	Atm	Ice	Land	Ocean		

<http://www.cesm.ucar.edu/experiments/cesm1.0/>



Introduction to netCDF

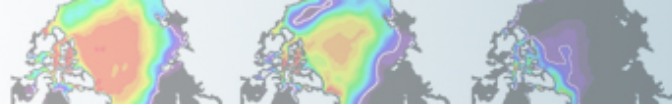
Netcdf stands for “network Common Data Form”

PROS: self-describing, portable, metadata friendly, supported by many languages including fortran, C/C++, Matlab, ferret, GrADS, NCL, IDL; viewing tools like nview / ncdump; and tool suites of file operators (NCO, CDO).

CONS: compression not available until netCDF4, oftentimes requires users to explicitly access information (not true in NCL)

<http://www.unidata.ucar.edu/software/netcdf>

<http://www.unidata.ucar.edu/software/netcdf/docs/BestPractices.html>



Introduction to ncdump

ncdump is a netcdf utility that allows one to dump the contents of the netcdf file to screen or file.

Files are often too big to dump to screen, but one can look at subsets of the file using the different ncdump options.

`ncdump slp.mon.mean.nc`

Dump entire contents of netCDF to screen (generally not used: too much information)

`ncdump -h slp.mon.mean.nc`

Dump header from netCDF file to screen (see next slide)

`ncdump -v slp slp.mon.mean.nc`

Dump the slp variable to the screen, after the header

`ncdump -v time slp.mon.mean.nc | less`

Display the time array using the UNIX command less, which allows one to page up/down using the arrows on the keyboard

<http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/ncdump.html>

Example output using ncdump -h

To view the contents of a netCDF file
we can use the ncdump utility:

```
ncdump -h slp.mon.mean.nc
```

```
netcdf slp.mon.mean {
```

```
  dimensions:
```

```
    lon = 144 ;
```

```
    lat = 73 ;
```

```
    time = UNLIMITED ; // (744 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., 17609832. ;
```

```
    float slp(time, lat, lon) ;
```

```
      slp:long_name = "Sea Level Pressure" ;
```

```
      slp:valid_range = 870.f, 1150.f ;
```

```
      slp:actual_range = 960.1486f, 1082.558f ;
```

```
      slp:units = "millibars" ;
```

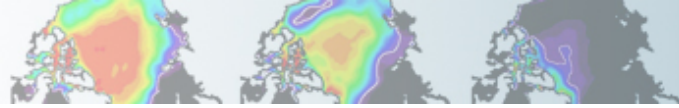
```
      slp:missing_value = -9.96921e+36f ;
```

```
  // global attributes:
```

```
    :title = "Monthly mean slp from the NCEP Reanalysis" ;
```

```
    :description = "Data is from NMC initialized reanalysis\n",  
    "(4x/day). These are the 0.9950 sigma level values." ;
```

```
    :Conventions = "COARDS" ; }
```



Introduction to netCDF operators (NCO)

NCO is a suite of programs designed to perform certain “operations” on netcdf files, i.e., things like averaging, concatenating, subsetting, or metadata manipulation.

Command-line operations are extremely useful for processing model data given that modellers often work in a UNIX-type environment. The NCO’s do much of the “heavy lifting” behind the scenes in the diagnostics packages.

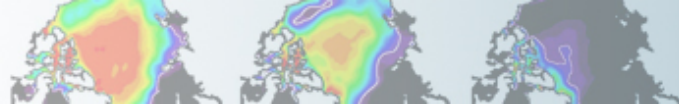
UNIX wildcards are accepted for many of the operators.

The NCO Homepage can be found at <http://nco.sourceforge.net>

The Operator Reference Manual can be found at:

<http://nco.sourceforge.net/nco.html#Operator-Reference-Manual>

Note: There are many other netCDF operators beyond what will be described here.



Introduction to netCDF operators

NCRA (netCDF record averager)

Example: `ncra file1.nc file2.nc avgfile.nc`

`file1.nc` = input model history file, for jan year 1

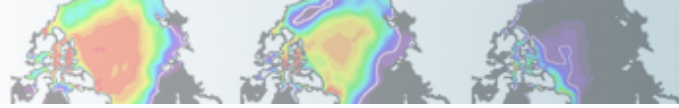
`file2.nc` = input model history file, for feb year 1

`avgfile.nc` = new file consisting of jan/feb averaged data for all fields found in the input model history file.

NCRCAT (netCDF record concatenator)

Examples: `ncrcat file1.nc file2.nc out12.nc`

`out12.nc` = new model history time series file consisting of the months of jan and feb, year 1. Each time-varying field in this file now has 2 time steps.



Introduction to netCDF operators

NCEA (netCDF ensemble averager)

Example: `ncea amip_r01.nc amip_r02.nc amip_r03.nc amip_ENS.nc`

`amip_r01.nc` = input file from ensemble member #1

containing monthly Jan-Dec year 1 data

`amip_r02.nc` = same as above but contains data from ensemble member #2

`amip_r03.nc` = same as above but contains data from ensemble member #3

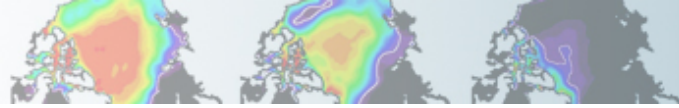
`amip_ENS.nc` = new file consisting of monthly Jan-Dec year 1 data averaged across the 3 ensemble members.

NCDIFF (netCDF differencer)

Examples: `ncdiff amip_r01.nc amip_r02.nc diff.nc`

`diff.nc` = contains the differences between `amip_r01.nc` and `amip_r02.nc`.

Note: Useful for debugging purposes.



Introduction to netCDF operators

NCKS (netCDF “Kitchen Sink” = does just about anything)

Combines various netCDF utilities that allow one to cut and paste subsets of data into a new file.

Example: `ncks -v TEMP f40_test.pop.h.1993-11.nc f40_test.TEMP.199311.nc`

`f40_test.pop.h.1993-11.nc` = input model history file (monthly)

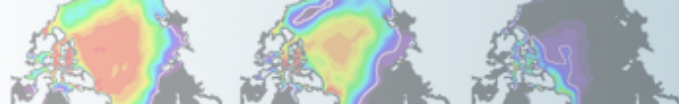
`-v TEMP` = only grab the TEMP variable

`f40_test.TEMP.1993-11.nc` = output file containing TEMP + associated coordinate variables

Note #1: Only those variables specified by `-v` and their associated coordinate variables are included in the output file. As the variables `date`, `TLAT`, and `TLONG` are not coordinate variables of `TEMP`, they won't be copied to the output file unless one does this:

`ncks -v TEMP,date,TLAT,TLONG f40_test.pop.h.1993-11.nc f40_test.T.1993-11.nc`

Note #2: Wildcards not accepted.



Introduction to netCDF operators

netCDF operator options

- v Operates only on those variables listed.

```
ncks -v T,U,PS in.nc out.nc
```

- x -v Operates on all variables except those listed.

```
ncrcat -x -v CHI,CLDTOT 1999-01.nc 1999-02.nc out.nc
```

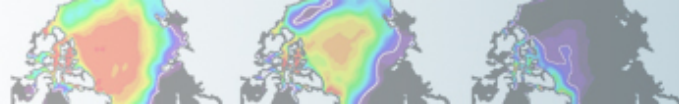
- d Operates on a subset of data.

```
ncks -d lon,0.,180. -d lat,0,63 in.nc out.nc
```

Real numbers indicate actual coordinate values, while integers indicate actual array indexes. In the above example, all longitudes will be grabbed from 0:180E, and the first 64 latitudes indexes will be grabbed.

- h Override automatic appending of the global history attribute with the NCO command issued (which can be very long)

More options exist beyond what was discussed here.



Introduction to Climate Data Operators (CDO)

CDO are very similar to the NCO. They are similar simple command line operators that do a variety of tasks including: detrending, EOF analysis, and similar calculations.

CDO are not currently used in the diagnostics packages, so we will not go into specifics here. We mention the CDO to make you aware of their existence.

The CDO Homepage can be found at:

<https://code.zmaw.de/projects/cdo/>

CDO documentation can be found at:

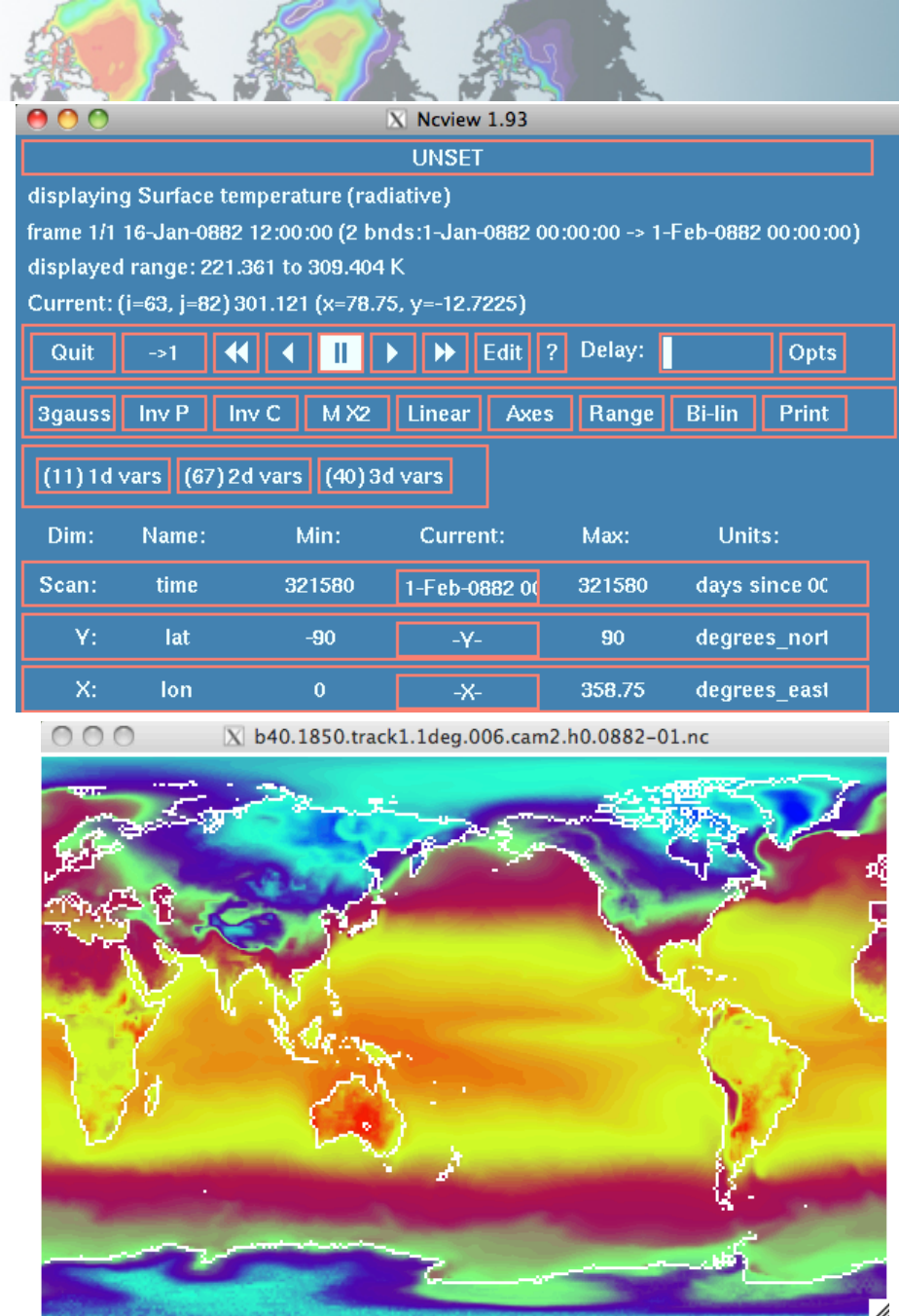
<https://code.zmaw.de/projects/cdo/wiki/Cdo#Documentation>

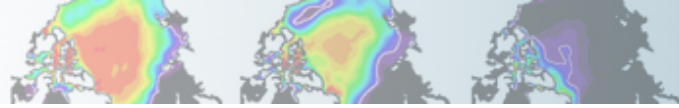
Introduction to nview

ncview is a graphical interface which allows one to quickly view the variables inside a netcdf file.

Example: `ncview file1.nc`

ncview allows you to interactively visualize a selected variable across a selected range (time, spatial).





ImageMagick

ImageMagick is a free suite of software that that can be used to display, manipulate, or convert images. It can also be used to create movies.

There are two ways to use ImageMagick. One way is to simply display the image and alter it using pop-up menus visible after clicking on the image:

```
display plot1.png
```

A second way is to alter an image at the command line, which is usually the faster and cleaner way to do it:

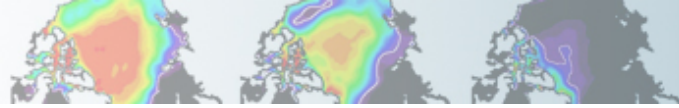
```
convert -density 144 -rotate 270 plot2.ps plot2.jpg
```

(set the resolution to 2x default, rotate the image 270 degrees, and convert to a jpg.)

There are many options available when using convert, some of which you may need to use depending on your version of ImageMagick:

```
convert -trim +repage -background white -flatten plot2.ps plot2.png
```

(crop out all the possible white space, reset various settings, set the background to white, create a canvas based on white background and merge layers, and convert to a png.)



ImageMagick

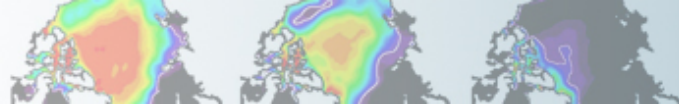
ImageMagick is a free suite of software that that can be used to display, manipulate, or convert images. It can also be used to create movies.

To create a movie from the command line:

```
convert -loop 0 -adjoin -delay 45 *.gif movie.gif
```

(loop through the movie once, create the movie (-adjoin),
and increase the time between slides (-delay 0 is the default))

<http://www.imagemagick.org>



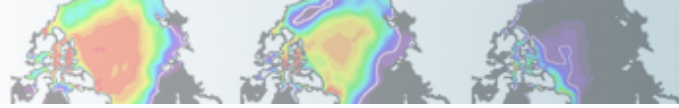
gv (Ghostview)

gv and Gnome Ghostview are simple programs that allow one to view postscript files:

`gv plot4.ps` (gv) or `gcv plot4.ps` (gnome ghostview)
(gv is installed on mirage cluster)

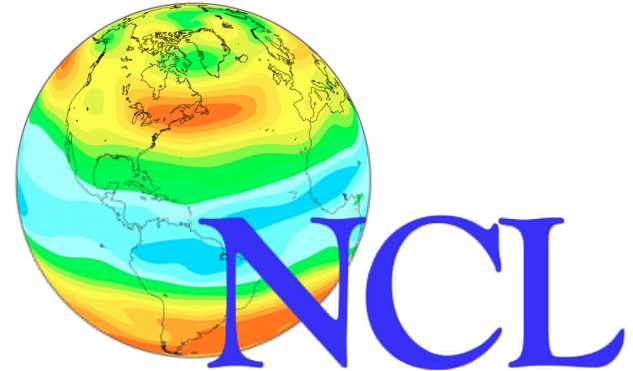
Once displayed, one can alter the orientation of the image, or change its' size, or print specific pages amongst a group of pages. For viewing postscript (or encapsulated postscripts), gv should be used.

<http://pages.cs.wisc.edu/~ghost/gv/index.htm>



NCL

NCL is an interpreted language designed for data processing and visualization. NCL is free, portable, allows for the creation of excellent graphics, can input/output multiple file formats, and contains numerous functions and procedures that make data processing easier.

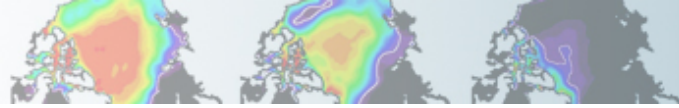


<http://www.ncl.ucar.edu>

Support: Postings to the ncl-talk email list concerning the NCL language are often answered within 24 hours by the NCL developers or by other NCL users.

Many downloadable examples are provided.

NCL is the official CESM processing language.



NCL

NCL easily reads in netCDF files:

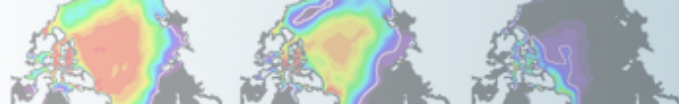
```
a = addfile("b40.1850.track1.1deg.006.0100-01.nc","r")
z3 = a->Z3 ; all metadata imported
```

NCL specializes in regridding (more on this later), whether from one grid to another:

```
lat = ispan(-89,89,2)
lon = ispan(0,358,2)
z3_rg = linint2(z3&lon,z3&lat,z3,True,lon,lat,0) ; regrid to 2x2
```

or from CAM's hybrid sigma levels to pressure levels.

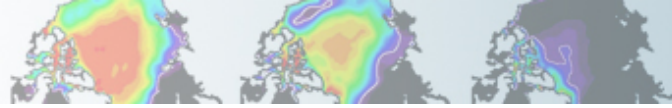
```
lev_p = (/ 850., 700., 500., 300., 200. /)
P0mb = 0.01*a->P0
tbot = T(klev-1,,:)
Z3_p = vint2p_ecmwf(z3,hyam,hybm,lev_p, \
                    PS,1,P0mb,1,True,-1,tbot,PHIS)
```



NCL

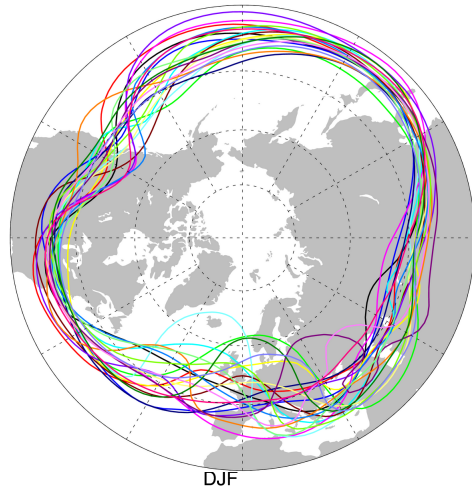
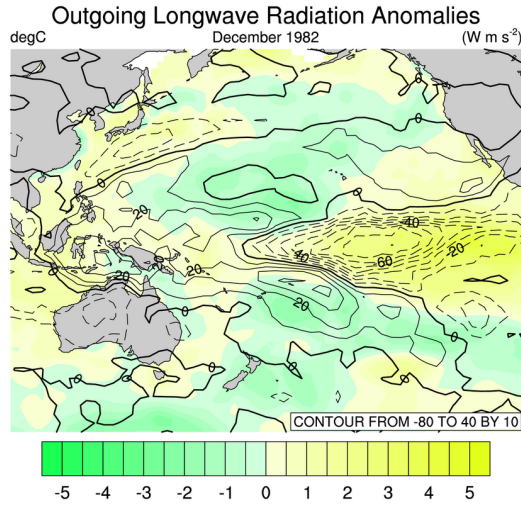
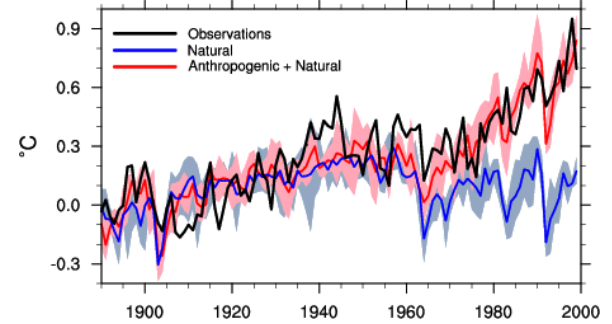
NCL's graphics package is exceptionally flexible. There are thousands of plot options (called resources) available that allow one to customize plots:

```
a = addfile("b40.1850.track1.1deg.006.0100-01.nc","r")
ts = a->TS(0,,:,:)
wks = gsn_open_wks("ps","test")
gsn_define_colormap(wks,"amwg")
res = True
res@mpCenterLonF = 180.
res@mpProjection = "WinkelTripel"
res@mpOutlineOn = True
res@mpGeophysicalLineColor = "gray70"
res@cnFillOn = True
res@gsnSpreadColors = True
plot = gsn_csm_contour_map(wks,ts,res)
```

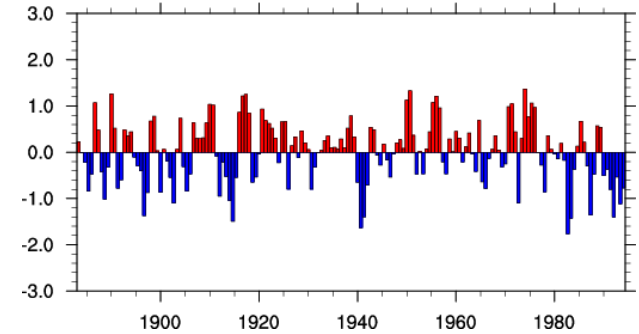


NCL Example Graphics

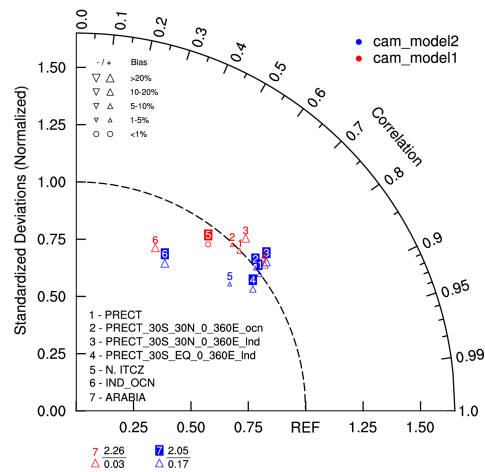
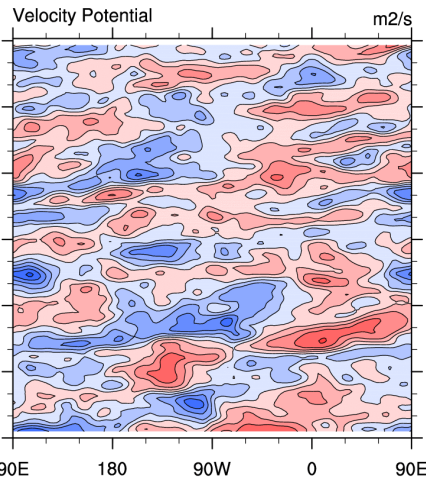
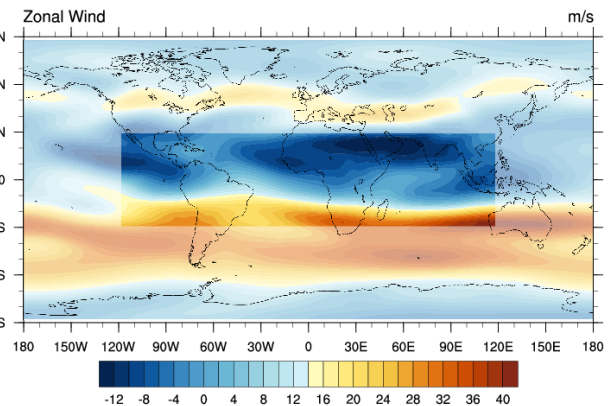
Parallel Climate Model Ensembles
Global Temperature Anomalies
from 1890-1919 average

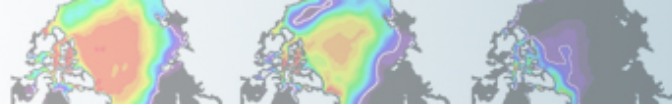


Darwin Southern Oscillation Index



Use transparency to de-emphasize a particular area





The NCL website: Examples

Chrome File Edit View History Bookmarks Window Help

Category List of NCL Application Examples

www.ncl.ucar.edu/Applications/

NCL Examples Functions Resources Popular Links What's New Support External

NCL Command Language

Category List of NCL Application Examples

[List of tips, resources, functions/procedures and the example scripts they appear in.]

[Example datasets | Templates]

This page contains links to hundreds of NCL scripts, and in most cases, a link to the graphic produced by that script. You can download any one of these scripts and plug in your own datasets. You can also download all of the scripts as a compressed tar file. (Use "tar -xvf ApplicationScripts.tgz" to extract the scripts.)

Most of the examples on this page use the GSUN interfaces. For best results, install the .hluresfile in your home directory before running any of these scripts.

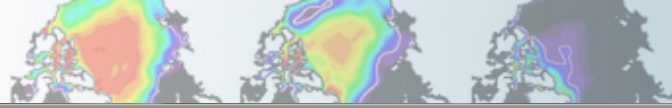
Click on any green category for a separate page of that category's listings only, and possibly more information.

[Alphabetical listing]

<p>Datasets</p> <ul style="list-style-type: none"> ARM AVHRR classification data: <ul style="list-style-type: none"> vegetation/cloud COAMPS COADS CRU EASE ERA40 GODAS HDF/HDF4/5-EOS: <ul style="list-style-type: none"> AIRS,HIRDLS,MLS,MOPITT MODIS,OMI,TES,TRMM High Res Precipitation MCSST NARR NIC snow & ice data NOGAPS seaWIFs shapefiles station data 	<p>Map projections</p> <ul style="list-style-type: none"> Maps only Map lat/lon grids General Coastlines Cylindrical <ul style="list-style-type: none"> equidistant Lambert conformal (masked) Lambert conformal (native) Native grid Polar stereographic Satellite Rotated lat- lon Tickmarks: map 	<p>Models</p> <ul style="list-style-type: none"> COAMPS CCCMA: CRCM CESM: Ice CESM: Land CESM: POP HOMME (SEAM) ICON NOGAPS Ocean: HYCOM Ocean: NCOM Ocean: NLOM Ocean: ORCA Ocean: ROMS Paleoclimatology PIPS Regional Climate Model TIGGE Project WRF 	<p>Data Analysis</p> <ul style="list-style-type: none"> Anomalies Binning Climate Indices Climatology COADS Complex Coefficients (GRIB) Correlations Divergent and rotational wind components EOFs ESMF regriding Filters GRADS CTL files Grid_filling Isentropic levels Local minima and maxima Meridional overturning MJO Clivar NCL system interactions and scripting Omega One-dimensional interpolation PDF: Probability Distributions Random Data to Grid Regression Regridding Sigma coordinate interpolation
<p>Plot Types</p> <ul style="list-style-type: none"> Bar charts Box plots Contours: no maps Contours: cylindrical equidistant Contours: native grid Contours: polar stereographic Contours: satellite 	<p>Plot Techniques</p> <ul style="list-style-type: none"> 2-dimensional vertical coordinates Animations Annotations Axes Coastlines New color capabilities 	<p>Special Plots</p> <ul style="list-style-type: none"> Evans plots Histograms Iso levels Meteograms Pie charts Skew-T Taylor diagram T-S diagram 3D plots 	

Models

- COAMPS
- CCCMA: CRCM
- CESM: Ice
- CESM: Land
- CESM: POP



The NCL website: Examples

NCL Graphics: CCSM Ice Model

http://www.ncl.ucar.edu/Applications/ice.shtml

Most Visited Getting Started Latest Headlines CISL's NCAR Comma... How to change Logi...

NCL Graphics: CCSM Ice Model

NCAR CISL VETS

NCL - Examples - Functions - Resources - Popular Links - What's New - Support -

NCAR Command Language

NCAR Home > Application examples > Models || Data files for some examples

Example pages containing: tips | resources | functions/procedures

CCSM Ice Model

Ice thickness

1 2 3 4 5 6 7 8 9 10

These plots use the `gsn_csm_contour_map_polar` high-level plot interface. More examples of its use are available on the [polar](#) example page.

1D coordinate variables

`ice_1.ncd`: A simple plot showing ice thickness with a spectrum of color.

`gsn_csm_contour_map_polar` is the plot interface that draws contours on a polar stereographic map.

`cnfWOn` = True, Turns on the color fill.

1D coordinate variables

`ice_2.ncd`: A polar vector plot of ice velocity.

`gsn_csm_vector_map_polar` is the plot interface that draws vectors on a polar stereographic map.

`vcMonoLineArrowColor` = False, Turns on color vectors.

To turn on curly vectors, set the resource `vcGlyphStyle` = "CurlyVector"

2D coordinate variables

`ice_3.ncd`: Demonstrates plotting ice variables from file that contain 2D coordinate variables. There are many more example of this technique on the [pop ocean](#) page.

As of NCL version 4.2.0.a025, data that has 2D lat/lon coordinates can be plotted directly in physical space. To do this, simply read in the coordinates and assign them the following attributes:

```
lat2d = f->TLAT
lon2d = f->TLONG
+8lon2d = 1- lon2d
```

Done

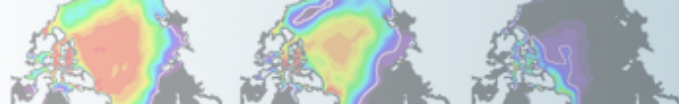
1D coordinate variables

`ice_2.ncd`: A polar vector plot of ice velocity.

`gsn_csm_vector_map_polar` is the plot interface that draws vectors on a polar stereographic map.

`vcMonoLineArrowColor` = False, Turns on color vectors.

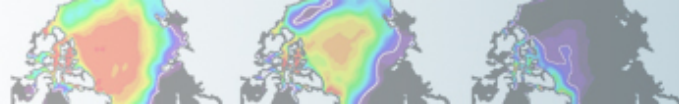
To turn on curly vectors, set the resource `vcGlyphStyle` = "CurlyVector"



NCL

Notable features added to NCL in the past year:

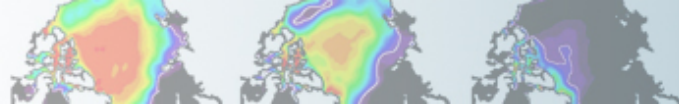
- NCL is now 64-bit capable. Array sizes are unlimited on 64-bit systems.
- New color display model: Cairo. Allows transparent colors, along with the use of more than 256 colors in your colormap.
- New regridding capabilities with integrated Earth System Modeling Framework (ESMF) regridding software. One can now regrid rectilinear (example: finite volume), curvilinear (ex. POP), and unstructured data (ex. CAM-SE) to any other type of grid. Three different interpolation methods can be used: bilinear, patch, and conservative.



NCL

For more information, or to get started learning NCL:

- http://www.ncl.ucar.edu/get_started.shtml
- Take the NCL class (information available on NCL website)
- Page through the NCL mini-language and processing manuals
<http://www.ncl.ucar.edu/Document/Manuals/>

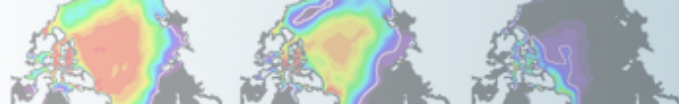


Using NCL in Practical Lab #3

Within the lab, you are going to be provided NCL scripts that post-process the monthly model data that you created and draw simple graphics.

What is meant by post-processing: Convert the model history data from one time step all variables on one file to all time steps, one variable per file. (Also convert CAM 3D data from hybrid-sigma levels to selected pressure levels.)

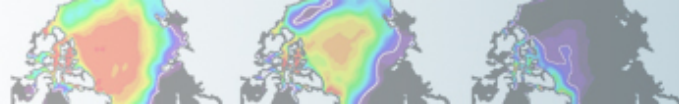
The diagnostic script suites all use NCL, and you will have the opportunity to run these as well.



Diagnostics Package Updates

There have been some significant changes to the atmosphere and ocean diagnostics packages in the past 7 months.

- The ocean package now uses NCL exclusively. The code has also been updated to use the newest and most efficient versions of various NCL functions. This enhances portability, results in a modest-speed up in processing time, and upgrades the quality of the output graphics.
- Both the atmosphere and ocean packages now have a built-in option to enlist the use of Swift, a software package that allows for task multi-processing. Turning Swift on can significantly reduce the wall clock time needed to run the diagnostic packages.
- However, we will not be enabling the use of Swift here due to the newness of this capability, the length of time needed to set up your accounts to use Swift, and most importantly the lack of need due to the limited time span of your model simulations.
- The Swift version of the ocean package now allows model simulation data to be on a local disk. Previously data had to be on a long-term archiving system (NCAR's HPSS), or one had to average the data by hand and place the necessary files in an exact spot.



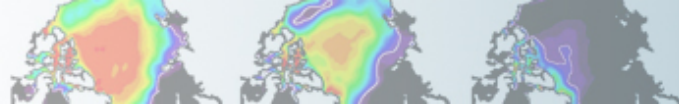
Diagnostics Packages

The 4 diagnostics packages are all available off of the SVN repository.

The diagnostics packages were built to be flexible. Many comparisons are possible using the options provided. Here, we have you set a few options to compare observations to your model run. You can also use the diagnostics to compare model runs to one another, regardless of model version.

The atmospheric, land, and ice packages each have one script that can do comparisons vs. observations or vs. another model run. The ocean diagnostics have three main scripts: popdiag (for comparison to observations), popdiagdiff (for comparison to another model run), and popdiagts (calculates various ocean time series).

Typically, 20 or 30 year time slices of data are analyzed using the diagnostics. (Exception: the popdiagts script is usually run on the entire run.) Here, you only have ~2 years of data, so that's what we will use.



Diagnostics Packages

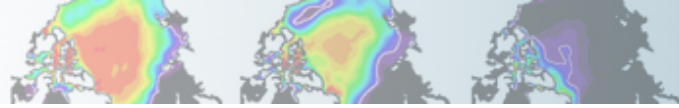
Each diagnostics package has different requirements in terms of the minimum amount of data required for them to run. (Ocean: 12 months, Atmosphere, Land: 14 months, Ice: 24 months) If you do not have the amount of data needed to run a specific diagnostics package, there is a directory set up with years 510-512 of CCSM4 pre-industrial control data here: `/glade/scratch/asphilli/archive/b40.1850.track1.2deg.003`

(Path also given in `test_data_location.txt` file found in `scripts/` directory.)

Note #1: Each diagnostics package will take around a 1/2hr to run. It is suggested that you start one of these packages first, and then move on to the post-processing or NCL graphics scripts.

Note #2: If you wish to take these diagnostics packages back with you to your home institution, you will need to have the netCDF operators and NCL installed. You will also need to modify the `.csh` scripts to work on your local machine.

Note #3: When submitting the diagnostics scripts, the syntax “`>&!`” is used. What this means: `>` = pass the on-screen output to a file, `&` = run in background mode, and `!` = overwrite the existing output file if necessary.



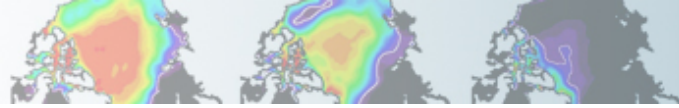
Practical Lab #3

Within the lab, you will have the opportunity to play with the CESM history files that you created. There are 4 sets of diagnostics scripts, 4 NCL post-processing scripts, and 7 NCL graphics creating scripts. You will also be able to try out the various software packages discussed earlier (ncview, ImageMagick, etc.).

The following slides contain information about how to run the various scripts on CISL's mirage system, along with exercises that you can try. It is suggested that you first focus on running those scripts written for the model component that you're most interested in. For instance, if you're an oceanographer, try running the ocean diagnostics script, along with the ocean post-processing script and ocean graphics NCL scripts.

Once you've completed running the scripts for your favorite component, take a run at the other model component scripts, or try the exercises or challenges on the last slide.

(Note that within a few months bluefire/mirage will be replaced by yellowstone/caldera, and that as such these directions might not be applicable to running on the new systems.)



Getting Started

There are a number of scripts that you will need to copy to your own directory. Here's what you will need to do to get set up:

- 1) The diagnostics packages, post-processing scripts, and plotting scripts will all be run on the CISL mirage cluster. Logon to mirage0, mirage1, mirage2, or mirage5:

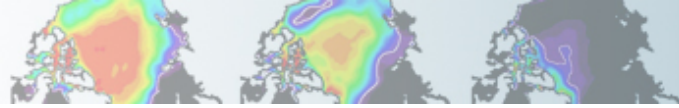
```
ssh -Y mirage5.ucar.edu
```

- 2) Create a new directory in your home called scripts, and cd into it:

```
mkdir scripts  
cd scripts
```

- 3) Copy all the necessary files over to your scripts directory, and move the hluresfile (sets NCL defaults) to your home directory and rename in .hluresfile:

```
cp -R /glade/home/asphilli/CESM_tutorial/* .  
mv hluresfile ../.hluresfile
```

Getting Started

- 4) Within your `.cshrc` or `.tcshrc` files, make sure NCL is loaded in your path:

```
cd
```

```
nedit .tcshrc (or use xemacs, vi, etc.)
```

add the following (if these lines are not there):

```
setenv NCARG_ROOT /fs/local
```

```
setenv PATH $NCARG_ROOT/bin:$PATH
```

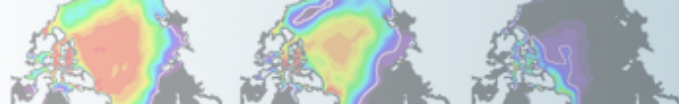
(If you did need to add the above two lines, make sure you source the `.tcshrc` file after modifying and saving it: `source .tcshrc`)

A sample `tcshrc` file is provided in the `scripts` directory if needed. To use it, do the following:

```
mv tcshrc ../.tcshrc
```

```
source .tcshrc
```

- 5) Reminder: Your model output will be under:
`/glade/scratch/<logname>/archive/<run>`



AMWG Diagnostics Package

To run the atmospheric diagnostics script:

(Note that setting a 0/1 option to 0 turns that option ON.)

1) `cd` to your scripts directory, then into `atm_diag`:

```
cd (changes to your home directory)
cd scripts/atm_diag
```

2) Open up the file `diag120131.csh` using your favorite text editor:

```
nedit diag120131.csh (or use xemacs, vi, etc.)
```

3) Modify the following lines and save the file:

line 105 Enter your run name

lines 107/108 Change “user” to your logname

line 120 Enter the model year you wish to start the diagnostics on

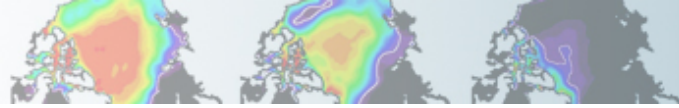
line 121 Enter the # of years you wish to analyze

4) Make the necessary atmospheric diagnostics directories:

```
mkdir /glade/scratch/<logname>/amwg
```

```
mkdir /glade/scratch/<logname>/amwg/climo
```

```
mkdir /glade/scratch/<logname>/amwg/diag
```

AMWG Diagnostics Package

- 5) Submit the job, let it run in background mode, and write the output to a file named atm.out:

```
./diag120131.csh >&! atm.out &
```

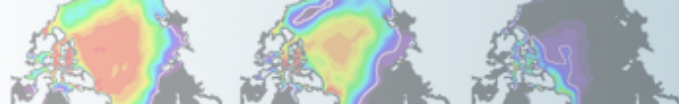
- 6) If the diagnostics package errors out, check the output file atm.out, and correct the script.

- 7) Once the diagnostics script has successfully completed, a tar file should have been created here:

```
/glade/scratch/<logname>/amwg/diag/<run>
```

- 8) cd to your diag directory, create a new directory called html, move the tar file to the html directory, and untar it:

```
cd /glade/scratch/<logname>/amwg/diag/<run>  
mkdir html  
mv *.tar html/  
cd html  
tar -xf *.tar
```



AMWG Diagnostics Package

- 8) cd into the new directory, fire up a firefox window, and open up the index.html file:

```
cd <run>-obs  
/usr/bin/firefox &  
(File->Open File) then choose index.html
```

For reference: Your atmospheric diagnostics web files are located here:
`/glade/scratch/<logname>/amwg/diag/<run>/html/<run>-obs/`

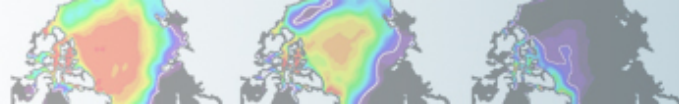
For more information about the AMWG Diagnostics Package:

<http://www.cgd.ucar.edu/amp/amwg/diagnostics/index.html>

If you wish to enable Swift by setting `use_swift` to 0, please see the website:

<http://www.cgd.ucar.edu/amp/amwg/diagnostics/news.html>

(See the 2012/01/31 section)

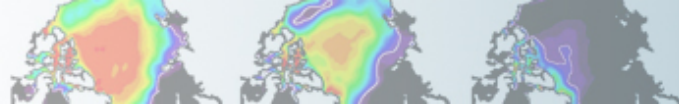


LMWG Diagnostics Package

To run the land diagnostics script:

(Note that setting a 0/1 option to 1 turns that option ON.)

- 1) `cd` to your scripts directory, then into `Ind_diag`:
`cd` (changes to your home directory)
`cd scripts/Ind_diag`
- 2) Open up the file `Ind_diag4.1.csh` using your favorite text editor:
`nedit Ind_template4.9.csh` (or use `xemacs`, `vi`, etc.)
- 3) Modify the following lines:
lines 66,67,68 Enter your run name
lines 95,97 Change “user” to your logname
line 163 Enter the model year you wish to start the diagnostics on
line 164 Enter the # of years you wish to analyze
line 173 Set to same value as line 162 (can be different though)
line 174 Set to same value as line 163 (again, can be different)



LMWG Diagnostics Package

- 4) Submit the job, let it run in background mode, and write the output to a file named Ind.out:

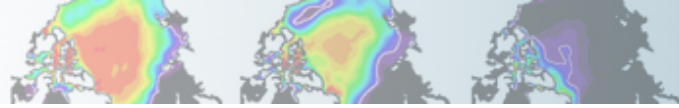
```
./Ind_template4.9.csh >&! Ind.out &
```

- 5) If the diagnostics package errors out, check the output file Ind.out, and correct the script.

- 6) Once the diagnostics script has successfully completed, a tar file should have been created in the following directory:
/glade/scratch/<logname>/Ind_diag/<run>

- 7) cd to the new directory in /glade/scratch, create a new directory called html, move the tar file to the html directory, and untar it:

```
cd /glade/scratch/<logname>/Ind_diag/<run>  
mkdir html  
mv *.tar html/  
cd html  
tar -xf *.tar
```



LMWG Diagnostics Package

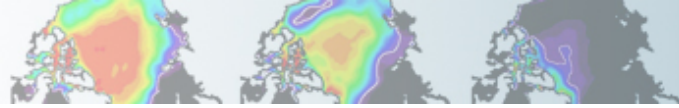
- 8) cd into the new directory, fire up a firefox window, and open up the setsIndex.html file:

```
cd <run>-obs  
/usr/bin/firefox &  
(File->Open File) then choose  
setsIndex.html
```

For reference: Your land diagnostics web files are located here:
/glade/scratch/<logname>/lnd_diag/<run>/html/<run>-obs/

For more information about the LMWG Diagnostics Package:

http://www.cgd.ucar.edu/tss/clm/diagnostics/webDir/lnd_diag4.1.htm



PCWG Diagnostics Package

To run the polar diagnostics script:

(Note that setting a 0/1 option to 1 turns that option ON.)

1) Copy your ice history files to a specified location on /glade/scratch:

```
cd /glade/scratch/<logname>
mkdir diags
mkdir diags/<run>
cd diags/<run>
cp ../../archive/<run>/ice/hist/*.h.* .
```

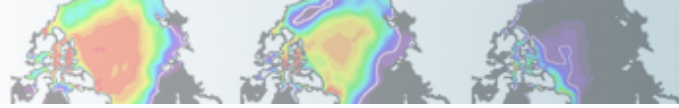
(the above command will take a bit)

2) cd to your scripts directory, then into ice_diag:

```
cd (changes to your home directory)
cd scripts/ice_diag
```

3) Open up the file ice_diag.csh using your favorite text editor:

```
nedit ice_diag.csh (or use xemacs, vi, etc.)
```

PCWG Diagnostics Package

- 7) cd to the new directory in /glade/scratch, create a new directory called html, and move the tar file to the html directory:

```
cd /glade/scratch/<logname>/web_plots/<run>  
mkdir html  
mv <run>.tar html/
```

- 8) cd into the /html directory, untar the tar file, and cd to the directory containing the index.html file:

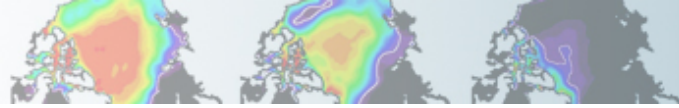
```
cd html  
tar -xf <run>.tar  
cd <run>/ice/yrs<startyr>-<endyr>
```

- 9) Fire up a firefox window, and open up the index.html file:

```
/usr/bin/firefox &  
(File->Open File) then choose index.html
```

For reference: Your ice diagnostics web files are located here:

```
/glade/scratch/<logname>/ (path continued below)  
web_plots/<run>/html/<run>/ice/yrs<startyr>-<endyr>
```

OMWG Diagnostics Package

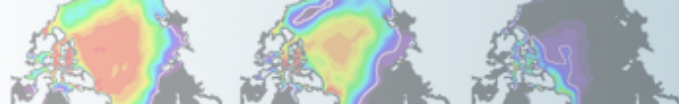
There are multiple oceanic diagnostics scripts. Here we will run the basic popdiag script, which compares your run to observations. The ocean diagnostics scripts were designed to access data off of the NCAR archival system (the HPSS). When the data is only available locally (as your model data is), we have to do a few things manually.

(Note that setting a 0/1 option to 1 turns that option ON.)

- 1) First, you need to create a couple of directories, and manually create the TAVG file. In the coding below, it is assumed that your model data starts in January year 1 and ends no sooner than December year 2.

```
cd /glade/scratch/<logname>
mkdir popdiag
mkdir popdiag/<run>
cd popdiag/<run>
ncra /glade/scratch/<logname>/archive/<run>/ocn/hist/*.h.
000{1,2}-??.nc tavg.1.2.nc
```

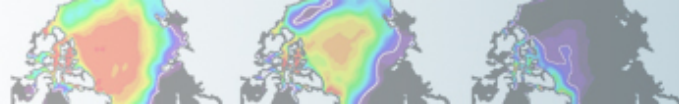
(What we are trying to do: average all the pop history files for model years 1 and 2. If you have years 1,2 and 3, then adjust the “000{1,2}-??.nc tavg.1.2.nc” syntax to “000{1-3}-??.nc tavg.1.3.nc”. If you need to use the 2° control run data, note that the data spans years 510-512, and you will need to adjust the ncra syntax accordingly.)



OMWG Diagnostics Package

- 2) `cd` to your scripts directory, then into `ocn_diag`:
`cd` (changes to your home directory)
`cd scripts/ocn_diag`
- 3) Open up the file `popdiag.csh` using your favorite text editor:
`nedit popdiag.csh` (or use `xemacs`, `vi`, etc.)
- 4) Modify the following lines:
line 13 Enter your run name
line 14 Change to `gx3v7` if analyzing your run. If analyzing
 `b40.1850.track1.1deg.003` leave set to `gx1v6`
line 15 Alter the first number to the start model year (Do not enter
line 16 Alter the first number to the end model year (leading 0's.)
line 68 replace `CESM_tutorial` with `scripts`

NOTE: The start/end years entered in lines 15 and 16 need to match the years specified in the `ncra` command shown in the previous slide. Thus, if you performed the `ncra` command on years 1 and 2, you would enter 1 on line 15 and 2 on line 16.



OMWG Diagnostics Package

- 5) Submit the job, let it run in background mode, and write the output to a file named ice.out:

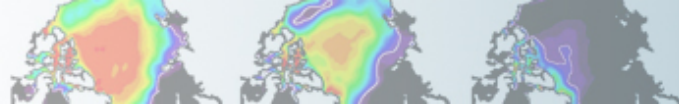
```
./popdiag.csh >&! ocn.out &
```

- 6) cd to the popdiag scratch directory, start firefox, and look at popdiag.html:

```
cd /glade/scratch/<logname>/popdiag/<run>  
/usr/bin/firefox &  
(File->Open File) then choose popdiag.html
```

For reference: Your ocean diagnostics web files are located here:
`/glade/scratch/<logname>/popdiag/<run>`

Note: In the future if you wish to turn Swift on, make sure you read the README-SWIFT.pdf file before setting `Swift = 1` in any of the `popdiag*.csh` scripts.



NCL post-processing scripts

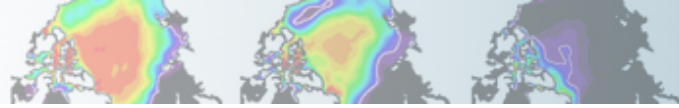
All 4 post-processing scripts are quite similar, and are located in your scripts directory. To list them, type: `ls *create*` . If these scripts are used for runs other than the tutorial runs, note that the created netCDF files may get quite large (especially pop files). This can be mitigated by setting `concat` and `concat_rm = False`.

To set up the post-processing scripts, alter lines 7-15 (7-17 for atm). There are comments to the right of each line explaining what each line does.

To run the atm script (for example), type the following:

```
ncl atm.create_timeseries.ncl
```

All 4 scripts will write the post-processed data to `work_dir` (set at top of each script)/`processed/<run>`. Once the post-processing is complete, we can use the new files in our NCL graphics scripts, or view them via `ncview`.



NCL Graphics Scripts

These scripts are set up so that they can read either raw history files from your archive directory (lnd,ice,ocn history files) or the post-processed files after they've been created by the NCL post-processing scripts.

You will need to modify the user defined file inputs at the top to point to your data files, either your raw history files or your newly created post-processed files.

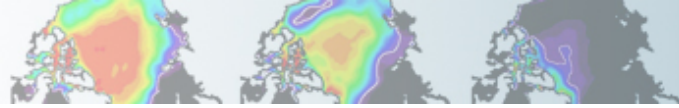
Once the files are modified, to execute the scripts, simply type (for example):

`ncl atm_latlon.ncl` . To see the script output use gv: `gv atm_latlon.ps`

There are 7 NCL graphics scripts available for you to run:

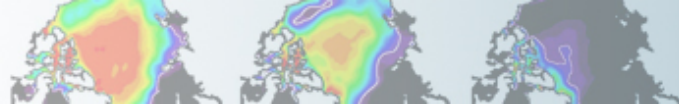
<code>atm_latlon.ncl</code>	<code>atm_nino34_ts.ncl</code>	<code>ice_south.ncl</code>
<code>ice_north.ncl</code>	<code>lnd_latlon.ncl</code>	<code>ocn_latlon.ncl</code>
<code>ocn_vectors.ncl</code>		

The `ocn_vectors.ncl` allows you to compare one ocean history file to another, and is more complicated (you can modify the first 50 lines) than the other 6 scripts. To run them, simply set the options at the top of the script.



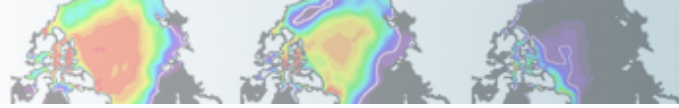
Exercises

- 1) Use `ncdump` to examine one of the model history files. Find a variable you've never heard of, then open up the same file using `ncview`, and plot that variable.
- 2) Modify one of the NCL scripts to plot a different variable.
- 3) Use the `netCDF` operators to difference two files. Plot various fields from the difference `netCDF` file using `ncview`.
- 4) Convert the output from one of the NCL scripts from `.ps` to `.jpg`, and crop out the white space. Import the image into Powerpoint.
- 5) Use the `netCDF` operators to concatenate sea level pressure and the variable `date` from all the monthly atmospheric history files (`.h0.`) into one file.
- 6) Same as 5), but only do this for the Northern Hemisphere.
- 7) Same as 6), but don't append the global history file attribute.



Challenges

- 1) Modify one of the NCL scripts to alter the look of the plot. Use the NCL website's Examples page to assist.
- 2) Add a variable or 3 to one of the post-processing scripts, then modify one of the NCL scripts to plot one of the new variables.
- 3) Use the atmospheric diagnostics package to compare your simulation against the simulation here:
`/glade/scratch/asphilli/archive/b40.1850.track1.2deg.003`
Make sure you compare the same number of years.

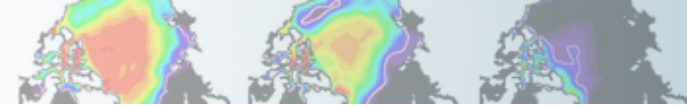


Introduction to the NCAR Archival System

The High Performance Storage System (HPSS)

- Tape-based archival system (same back end as MSS)
- FTP-like interface
- Connected to most CISL/CGD systems
- Reading and writing methods [*]nix-like: cp, mv, put, get,...
- Files do not expire, and can be as large as 1TB
- By default, 1 tape copy of each file is created.

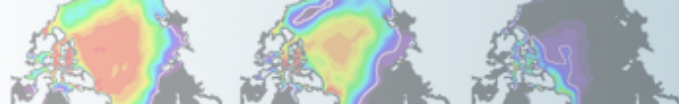
<http://www2.cisl.ucar.edu/docs/hpss-guide>



HPSS Commands

Key commands:

- `ls -l <full pathname for file>` : shows to screen specified file
- `get <source file>` : copies file(s) from HPSS
- `put <source file> <destination>` : copies file(s) to HPSS
- `lcd <local directory>` : changes your local directory
- `prompt` : do not prompt for input



HPSS Access

There are two ways to access data on the HPSS. One way is to interactively enter the HPSS from the command line by using HSI, and then using various ftp commands:

`hsi` (You may be prompted for your UCAR UCAS password)

`cd /CCSM/csm/b40.1850.track1.1deg.006/atm/hist`

`ls -l *.h0.1000*`

`prompt` (To turn off prompting)

`get *.h0.1000*` (All files will be written from the HPSS to whichever local directory you were in when you started your hsi session)

The other way to access the HPSS is from the command line:

`hsi -q 'lcd /glade/scratch/user; prompt; get /CCSM/csm/b40.1850.track1.1deg.006/atm/hist/*.h0.1000*'`

(Put the files on the local directory /glade/scratch/user, don't prompt, and grab all atmospheric history files that have ".h0.1000" in their name.)

`hsi -q 'lcd /glade/scratch/user; cd /USER; put atm.nc'`

(write /glade/scratch/user/atm.nc to the HPSS directory /USER; lcd unneeded if already in /glade/scratch/user)