# Day 3: Analyzing Model Output

Sanjiv Kumar, Rosie Fisher, Kyla Dahlin, Keith Oleson, Peter Lawrence, Will Wieder, and Danica Lombardozzi

**Land Model Working Group**

sanjiv@cola.iges.org

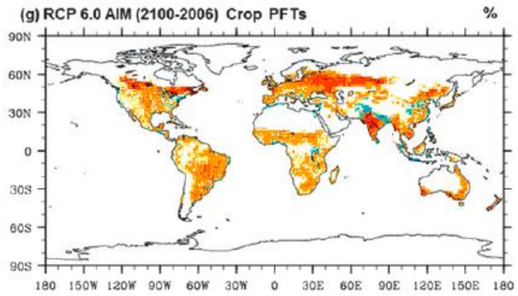
Thanks to Gordon Bonan, David Lawrence, Adam Phillips, Dennis Shea, Susan Bates, Christine Shields, Dave Bailey, and Kathy Pegion for their assistance.

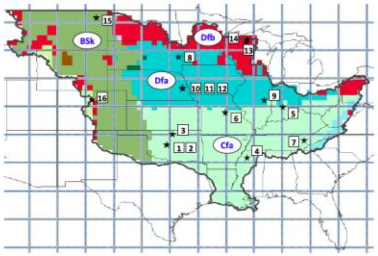
Modified from Adam Phillips' CESM Tutorial presentation

NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

U.S. DEPARTMENT OF ENERGY | Office of Science

# Motivation



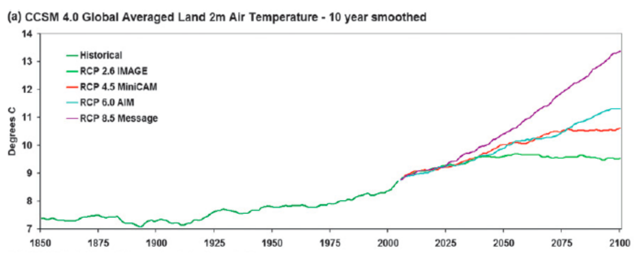



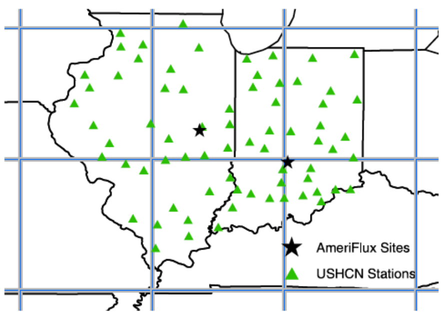

To find out one important thing from the sea of data (GB to TB model outputs)

To compare model outputs with the observations at a point (flux tower), regional (water basin), and global scales.

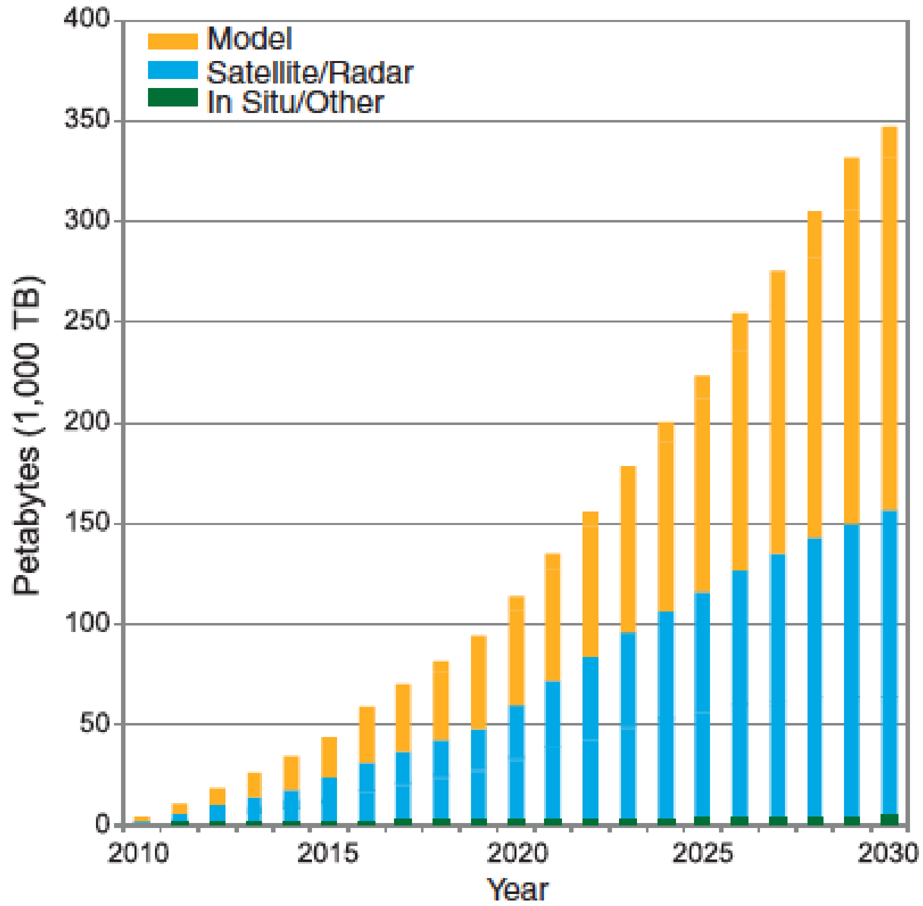
To make a sound scientific argument that is supported by the quantitative analysis of model results and the observations.

Publish your results



# Climate Data Explosion




**Climate Date Volume (worldwide) Projections**

Overpeck et al. 2011, Science

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

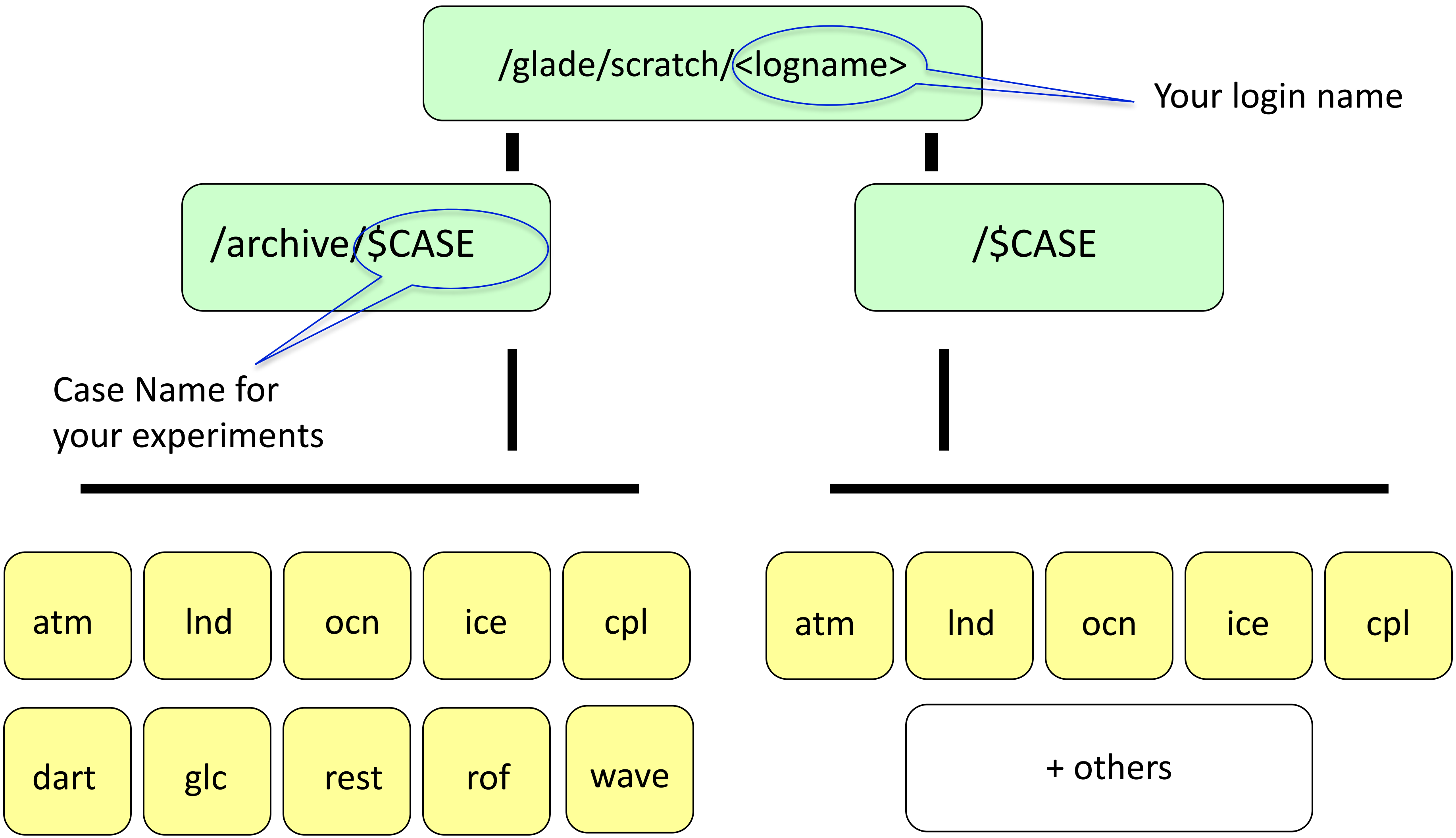
⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

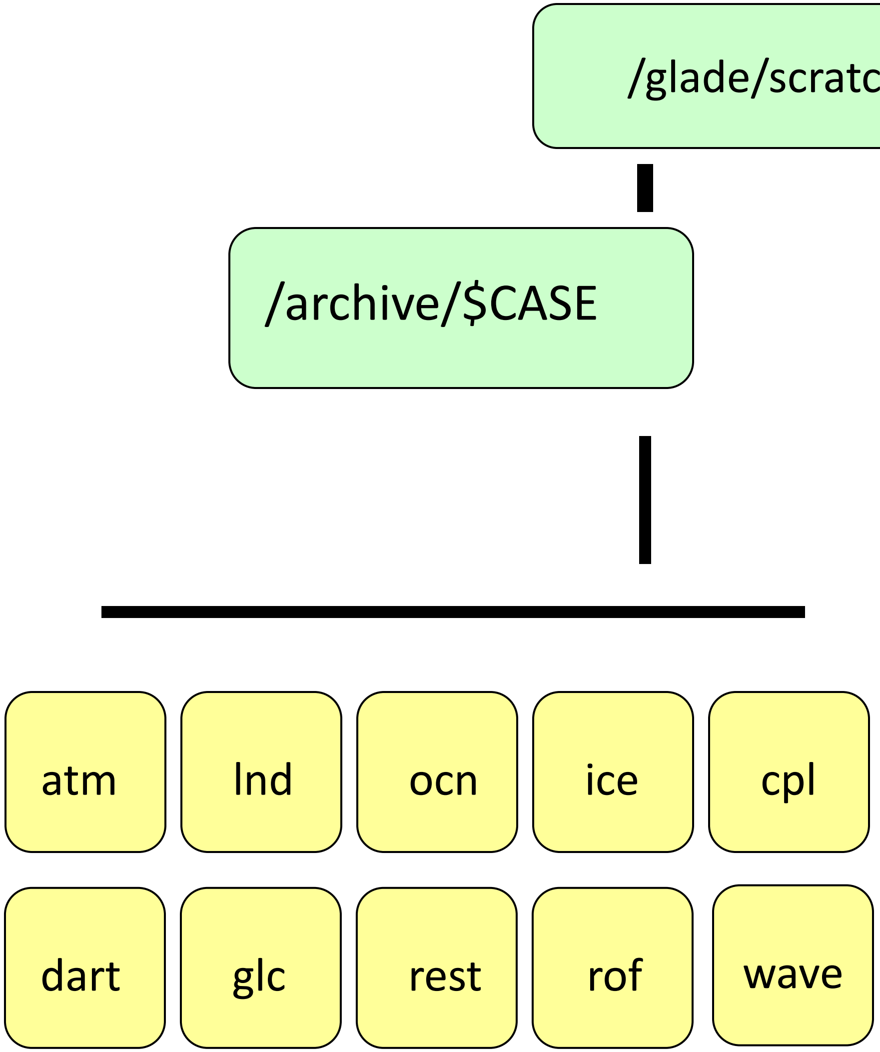
Break

⑧ Help Session/Exercise <60 minutes>

# Short-Term Archive and Runtime Directories



# Short-Term Archive




- By default, short term archiver writes to /glade/scratch/<logname>/archive/$CASE on yellowstone. (Can modify by setting DOUT_S_ROOT 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.*

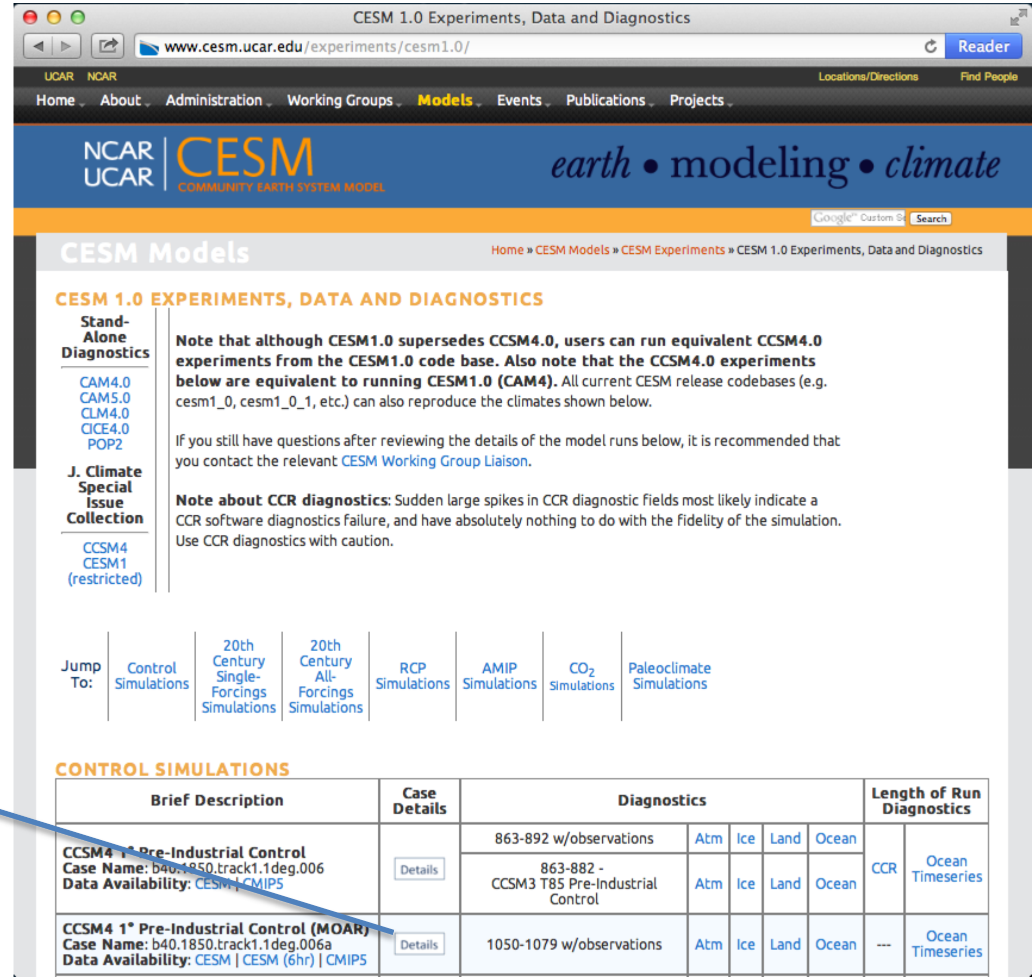
Example history file names:
f40_test.cam2.h0.1993-11.nc
f40_test.clm2.h0.1993-11.nc
f40_test.pop.h.1993-11.nc
f40_test.cice.h.1993-11.nc

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 Experiments Pages

**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, Data and Diagnostics

www.cesm.ucar.edu/experiments/cesm1.0/

UCAR NCAR Locations/Directions Find People

Home About Administration Working Groups Models Events Publications Projects

NCAR UCAR CESM COMMUNITY EARTH SYSTEM MODEL *earth • modeling • climate*

CESM Models

Home » CESM Models » CESM Experiments » CESM 1.0 Experiments, Data and Diagnostics

CESM 1.0 EXPERIMENTS, DATA AND DIAGNOSTICS

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

J. Climate Special Issue Collection: CCSM4, CESM1 (restricted)

**Note that although CESM1.0 supersedes CCSM4.0, users can run equivalent CCSM4.0 experiments from the CESM1.0 code base. Also note that the CCSM4.0 experiments below are equivalent to running CESM1.0 (CAM4).** All current CESM release codebases (e.g. cesm1_0, cesm1_0_1, etc.) can also reproduce the climates shown below.

If you still have questions after reviewing the details of the model runs below, it is recommended that you contact the relevant CESM Working Group Liaison.

**Note about CCR diagnostics**: Sudden large spikes in CCR diagnostic fields most likely indicate a CCR software diagnostics failure, and have absolutely nothing to do with the fidelity of the simulation. Use CCR diagnostics with caution.

Jump To: Control Simulations | 20th Century Single-Forcings Simulations | 20th Century All-Forcings Simulations | RCP Simulations | AMIP Simulations | $CO_2$ Simulations | Paleoclimate Simulations

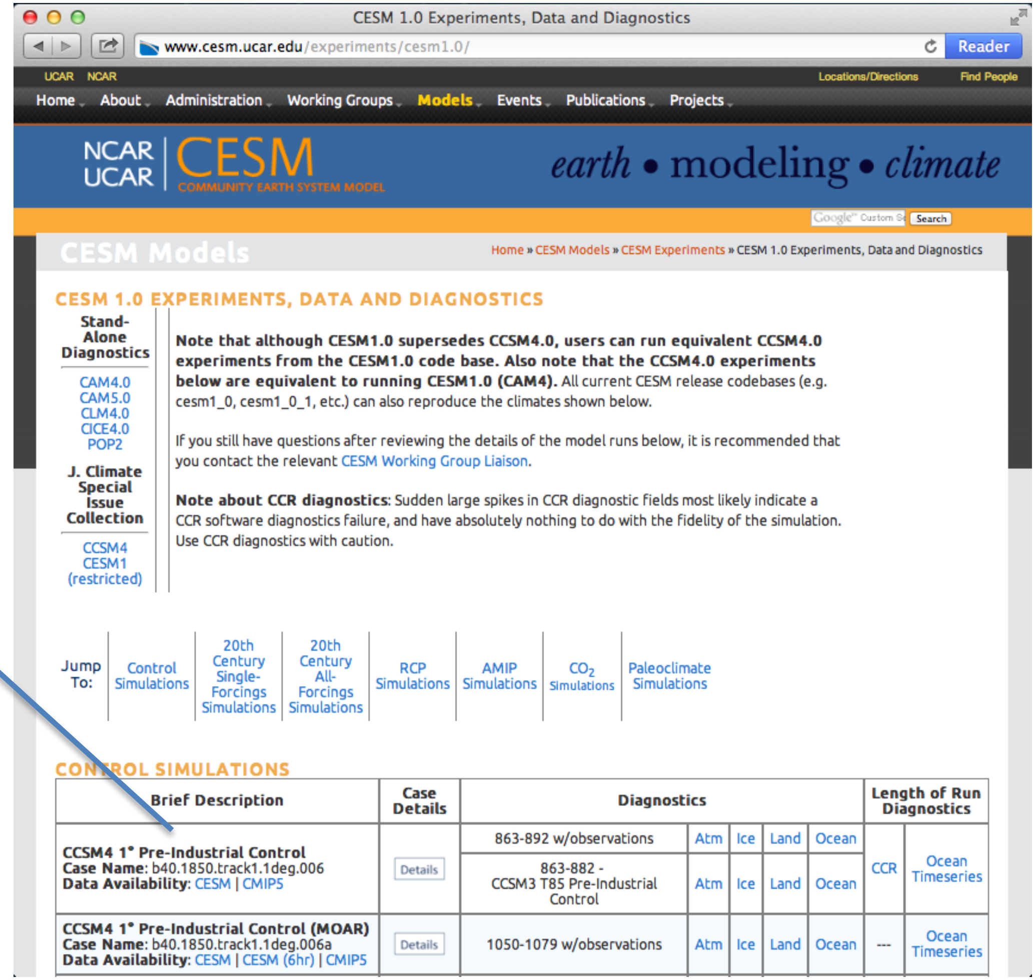
CONTROL SIMULATIONS

| Brief Description | Case Details | Diagnostics | | | | | Length of Run Diagnostics | |
|---|---|---|---|---|---|---|---|---|
| **CCSM4 1° Pre-Industrial Control** **Case Name**: b40.1850.track1.1deg.006 **Data Availability**: CESM \| CMIP5 | 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 Availability**: CESM \| CESM (6hr) \| CMIP5 | Details | 1050-1079 w/observations | Atm | Ice | Land | Ocean | --- | Ocean Timeseries |

http://www.cesm.ucar.edu/experiments/cesm1.0/
http://www.cesm.ucar.edu/experiments/cesm1.1/

# CESM Experiments Pages


Links to the Earth System Grid




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

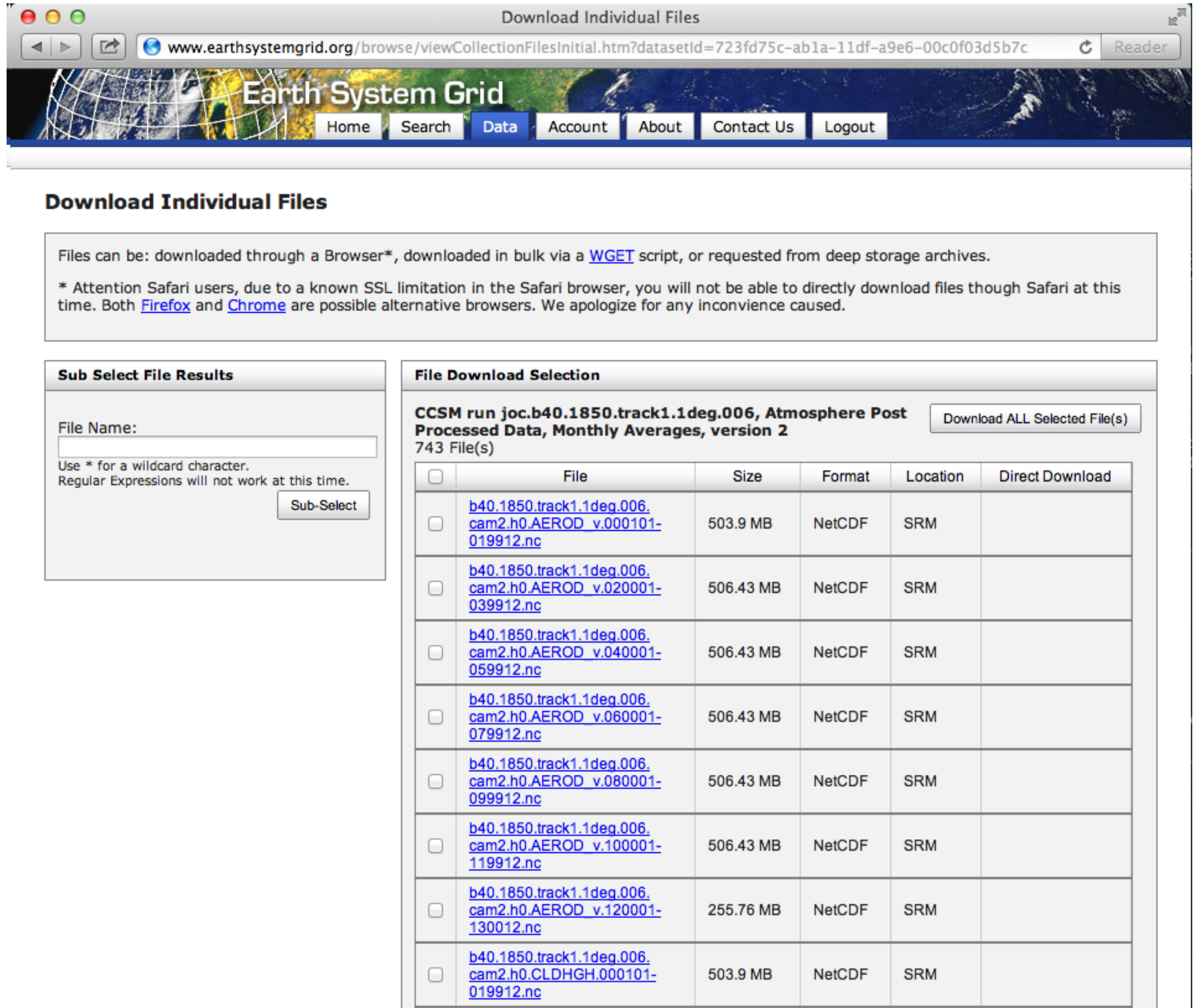
http://www.cesm.ucar.edu/experiments/cesm1.1/

# Earth System Grid

Publicly released CESM data is available via the ESG.

Registration is quick and easy. NCAR accounts are not required.

Post-processed data in CESM and CMIP formats along with raw history files are provided.




http://www.earthsystemgrid.org

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# Knowing about the data

Change directory to the outputs directory
cd /glade/scratch/$USER/archive/$CASE/lnd/hist

All analysis should be done on 'geyser' machines and not on 'yellowstone'. Yellowtone login nodes and production machines, whereas 'geyser' and 'caldera' is for data analysis and visualizations. Read best practices from CISL@NCAR.

Login to geyser from your Yellowstone login node
bsub -Is -q geyser -W 2:00 -n 1 -P UCGD0001 xterm

# Knowing about the data

```
CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0003-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-02.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0003-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-03.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0003-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-04.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0003-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-05.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0003-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-06.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0003-12.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-07.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-08.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-09.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-10.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-11.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-12.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-01.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-02.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-03.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-04.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-05.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-06.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0004-12.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-07.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-08.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-09.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-10.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-11.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-12.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-01.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-02.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-03.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-04.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-05.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-06.nc  CLM2014Tutorial_5yr1850test.clm2.h0.0005-12.nc
```

# Introduction to netCDF

- netCDF stands for “network Common Data Form”
- All CESM outputs are in netCDF format (.nc)
- self-describing, portable, metadata friendly
- supported by many languages including fortran, C/C++, **Matlab, R**, ferret, GrADS, **NCL**, IDL; viewing tools like ncview / ncdump; and tool suites of file operators (NCO, CDO)
- Data can be stored in several dimensions: ensembles X time X levels X latitudes X longitudes
- Not easy to edit – maintains the original data (you may want to make copy before editing a netCDF file)

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

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

# A Detailed Look at a netCDF Variable

```
>module load ncl
>ncl
>f1 = addfile("./CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc", "r")
>var1 = f1->TSA
>printVarSummary(var1)
```

```
Variable: var1
Type: float
Total Size: 55296 bytes
            13824 values
Number of Dimensions: 3
Dimensions and sizes:   [time | 1] x [lat | 96] x [lon | 144]
Coordinates:
            time: [31..31]
            lat: [-90..90]
            lon: [ 0..357.5]
Number Of Attributes: 5
  long_name :   2m air temperature
  units :       K
  cell_methods :        time: mean
  _FillValue :  1e+36
  missing_value :       1e+36
ncl 3>
```

**Data type**

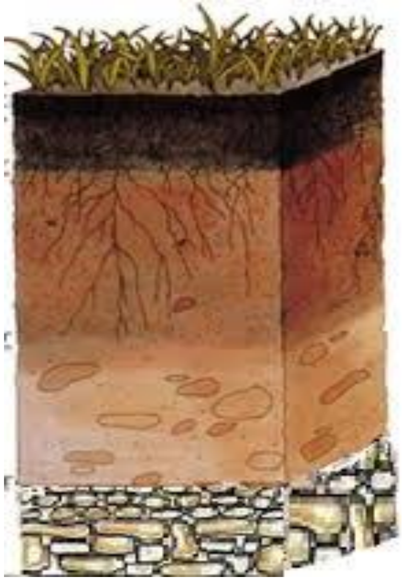
**Dimension and size**

**Coordinates**

**Attributes**

# Exercise in netCDF Variable

- Print Variable Summary for soil moisture (H2OSOI)
- Is there an extra dimension in H2OSOI?
- Explore this extra dimension (coordinates, size, etc.)



# Introduction to ncview

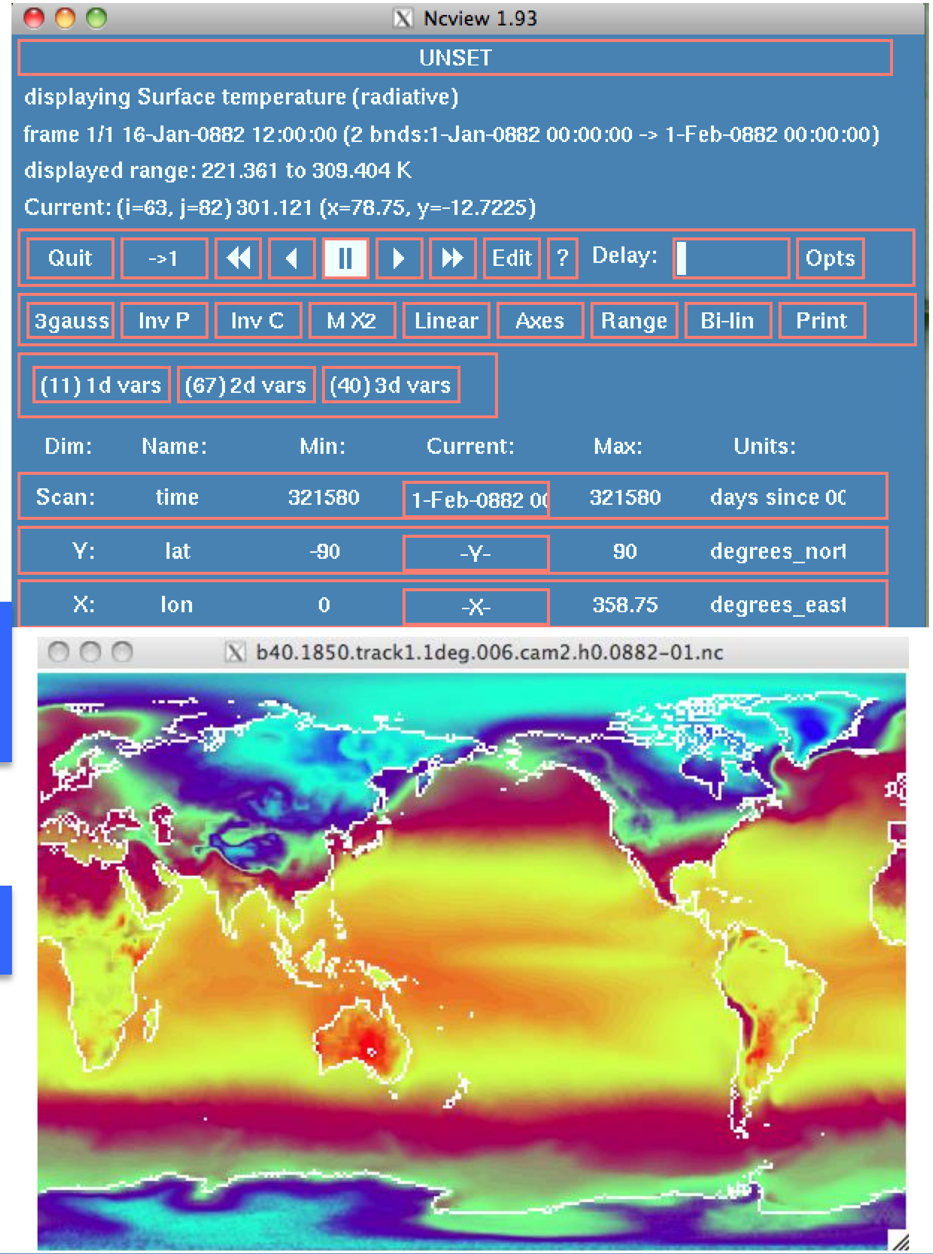
ncview is a graphical interface which allows one to quickly view the variables inside a netCDF file.

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

```
>module load ncview
>ncview CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc
```

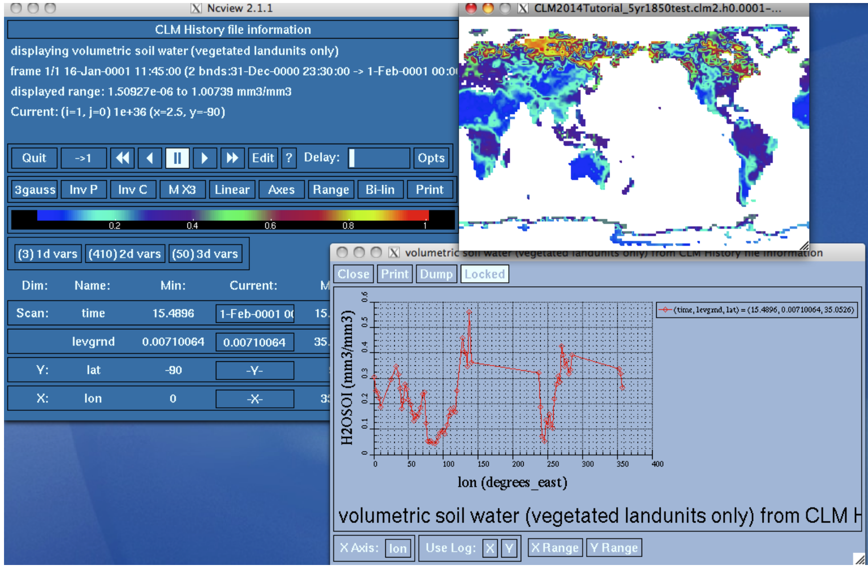
You can also join several netCDF files

```
>ncview *.nc
```



# Exercise in ncview

- Change the variable on the plot
- Click on any location on the plot
- See XY plot
- Change x – axis in the plot



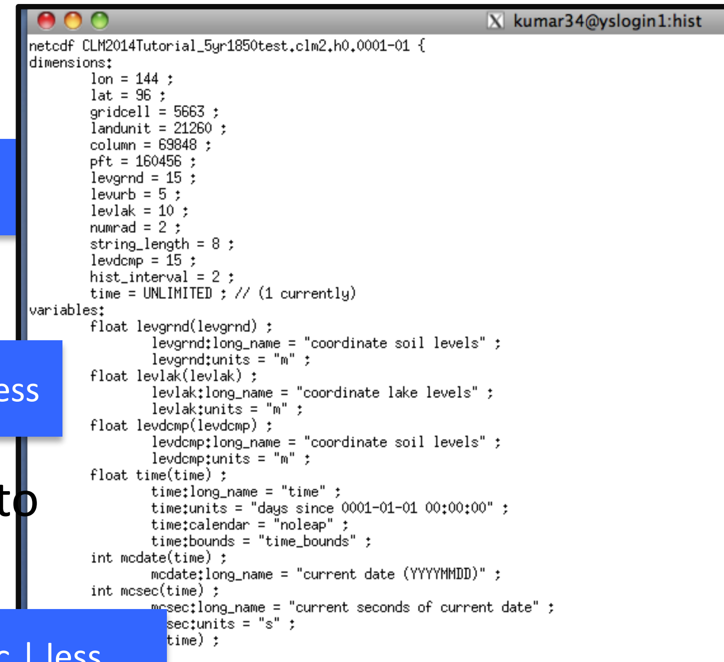

# Extra Credit Question

- Plot depth profile of soil moisture (H2OSOI)

# 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.



```
kumar34@yslogin1:hist
netcdf CLM2014Tutorial_5yr1850test.clm2.h0.0001-01 {
dimensions:
        lon = 144 ;
        lat = 96 ;
        gridcell = 5663 ;
        landunit = 21260 ;
        column = 69848 ;
        pft = 160456 ;
        levgrnd = 15 ;
        levurb = 5 ;
        levlak = 10 ;
        numrad = 2 ;
        string_length = 8 ;
        levdcmp = 15 ;
        hist_interval = 2 ;
        time = UNLIMITED ; // (1 currently)
variables:
        float levgrnd(levgrnd) ;
                levgrnd:long_name = "coordinate soil levels" ;
                levgrnd:units = "m" ;
        float levlak(levlak) ;
                levlak:long_name = "coordinate lake levels" ;
                levlak:units = "m" ;
        float levdcmp(levdcmp) ;
                levdcmp:long_name = "coordinate soil levels" ;
                levdcmp:units = "m" ;
        float time(time) ;
                time:long_name = "time" ;
                time:units = "days since 0001-01-01 00:00:00" ;
                time:calendar = "noleap" ;
                time:bounds = "time_bounds" ;
        int mcdate(time) ;
                mcdate:long_name = "current date (YYYYMMDD)" ;
        int mcsec(time) ;
                mcsec:long_name = "current seconds of current date" ;
                 sec:units = "s" ;
                 time) ;
```

Pipe the ncdump command through Less

>ncdump CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc | less

Dump only header

>ncdump –h CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc | less

Dump only one variable after the header; need to know the variable name

>ncdump –v TSA CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc | less

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

# There are so much data/information in a single file. We need a better way to make sense out of all these files!

## Motivation for writing analysis script

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

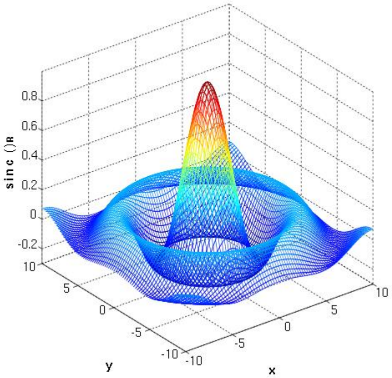
⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# An introduction to Matlab

- Matlab ("MATrix LABoratory") is a proprietorial software package for numerical computation, algorithm development and visualization.
- Designed at the University of New Mexico in ’70s
- 1M users in 2004.

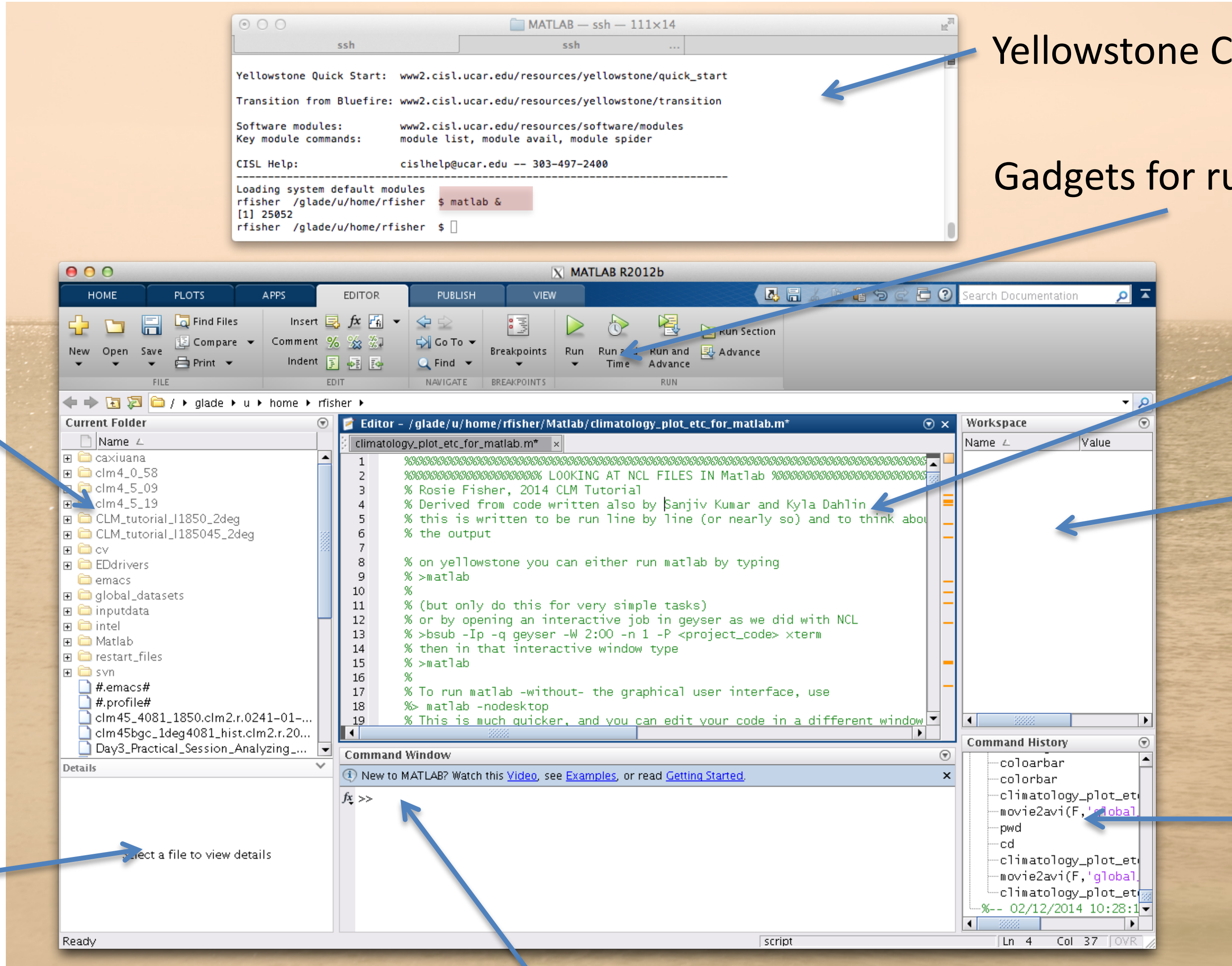


# Good / Bad

- Matlab is easy and quick to learn and to write.
- Graphics are highly customizable
- NetCDF functions are standard
- Like ‘R’, someone has usually done it already
- Getting help is easy (benefit of not being open-source)

- ‘R ‘and NCL have better inbuilt mapping tools
- It isn’t free!
- Organizations differ in their license situation, so you might just be lucky…

# Running Matlab with a Graphical User Interface (GUI)




Yellowstone Command Line

Gadgets for running code

Code

Variables

Command History

File Paths

Not very useful 'preview' feature

Command Line

# Running Matlab from the command line without a GUI

```
*****************************************************************************
*                          Welcome to Yellowstone                           *
*****************************************************************************

Next scheduled downtime:  dailyb.cisl.ucar.edu

Yellowstone Quick Start:  www2.cisl.ucar.edu/resources/yellowstone/quick_start

Transition from Bluefire: www2.cisl.ucar.edu/resources/yellowstone/transition

Software modules:         www2.cisl.ucar.edu/resources/software/modules
Key module commands:      module list, module avail, module spider

CISL Help:                cislhelp@ucar.edu -- 303-497-2400
-----------------------------------------------------------------------------
Loading system default modules
rfisher  /glade/u/home/rfisher  $ matlab -nodesktop

                                                    < M A T L A B (R) >
                                          Copyright 1984-2012 The MathWorks, Inc.
                                            R2012b (8.0.0.783) 64-bit (glnxa64)
                                                     August 22, 2012


To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

>>
```

# Where to look

- http://www.mathworks.com
- http://www.mathworks.com/help/matlab/
- http://www.mathworks.com/matlabcentral/fileexchange/

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# A Very Short Introduction to R

Kyla Dahlin

CLM Workshop

February 20, 2014

# What is R?

- A free and open-source programming language and software environment for statistical computing and graphics
- Developed from another programming language, S.
- ‘Object oriented’ which essentially means it’s structured in a way that’s relatively intuitive

# Good / Bad

- It’s great because there are LOTS of user-submitted packages that expand its tools.
- Nice people sometimes publish their code in association with scientific papers.
- Almost anything you can imagine doing in R someone has already done – you just have to wade through the internet to figure out how to do it.
- Wading through the internet can be frustrating / exhausting. And the letter ‘R’ is not exactly google-able.


BUILD AN R PACKAGE HE SAID
IT'LL TAKE AN HOUR HE SAID
memegenerator.net

# Some Things to Know

- "#" delineates a comment
- Indexes start counting from **1** (ncl and python, among others, start with 0)
- When assigning variables, y <- 4 is the same as y = 4
- When you string things together in R use c(…)
- Logical operators :

  ==, !=, >, <, >=, <=, |, &, is.na(), !is.na()

# Resources

- www.r-project.org
- www.rstudio.com
- www.google.com

For NetCDF in R:

www.image.ucar.edu/GSP/Software/Netcdf

# Intro to Exercise

Where's the script?

**/glade/p/cesm/lmwg/CLM2014_tutorial_space/Day3/climatology_plot_etc_for_R_final.txt**

# Intro to Exercise (load R)

on yellowstone you can either run R by typing

**>module load R**

**>R**

(but only do this for very simple tasks)

or by opening an interactive job in 'geyser' (part of yellowstone)

**>bsub -Ip -q geyser -W 2:00 -n 1 -P UCGD0001 xterm**

then in that interactive window type

**>module load R**

**>R**

# Install Packages

**>install.packages(c("ncdf", "raster", "rasterVis", "rgdal"))**

you'll be asked to select a CRAN mirror (I like CA1, personally) then you'll see lots of text go by

then load the packages

**>library(ncdf)**

**>library(raster)**

**>library(rasterVis)**

**>library(rgdal)**

# Do Stuff

1. Open a .nc file
2. Extract a variable and convince R that it's actually a map of the whole globe
3. Plot it
4. Read in a variable from a time series (bunch of files)
5. Calculate point-wise mean and standard deviation
6. Calculate the hottest year for each gridcell
7. Clip out and look at a region (Australia)
8. Quit R

# Today’s Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# 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:

```
f = addfile("CLM2014Tutorial_20yr1850BGC_n02_clm4_5_57.clm2.h0.0009-01.nc","r")
tsa = f->TSA                              ; all metadata imported
printVarSummary(tsa)

Variable: tsa
Type: float
Total Size: 55296 bytes
        13824 values
Number of Dimensions: 3
Dimensions and sizes:   [time | 1] x [lat | 96] x [lon | 144]
Coordinates:
        time: [3316..3316]
        lat: [-90..90]
        lon: [ 0..357.5]
Number Of Attributes: 5
  long_name :   2m air temperature
  units :       K
  cell_methods :        time: mean
  _FillValue :  1e+36
  missing_value :       1e+36
```

# NCL

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

```
f = addfile("CLM2014Tutorial_20yr1850BGC_n02_clm4_5_57.clm2.h0.0009-01.nc","r")
tsa = f->TSA                              ; all metadata imported
wks = gsn_open_wks("x11","test")
gsn_define_colormap(wks,"cosam")
res = True
 res@mpOutlineBoundarySets = "Geophysical"
 res@mpGeophysicalLineColor = "black"
 res@mpUSStateLineColor = "black"
 res@cnFillOn = True
 res@cnLinesOn = False
 res@cnFillMode = "RasterFill"
 res@cnLineLabelsOn = False
 res@gsnSpreadColors = True
…

 plot = new(2, graphic)                   ; if you have more than one plot then this is useful

 plot(0) = gsn_csm_contour_map(wks, tsa, res)
```

# The NCL website: Alphabetical listing

Alphabetical listing of NCL Functions

http://www.ncl.ucar.edu/Document/Functions/list_alpha.shtml

Yahoo! Google Maps YouTube Wikipedia News (580) Popular

NCAR CISL VETS Download Contrib

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

NCAR Command Language NCAR advanced

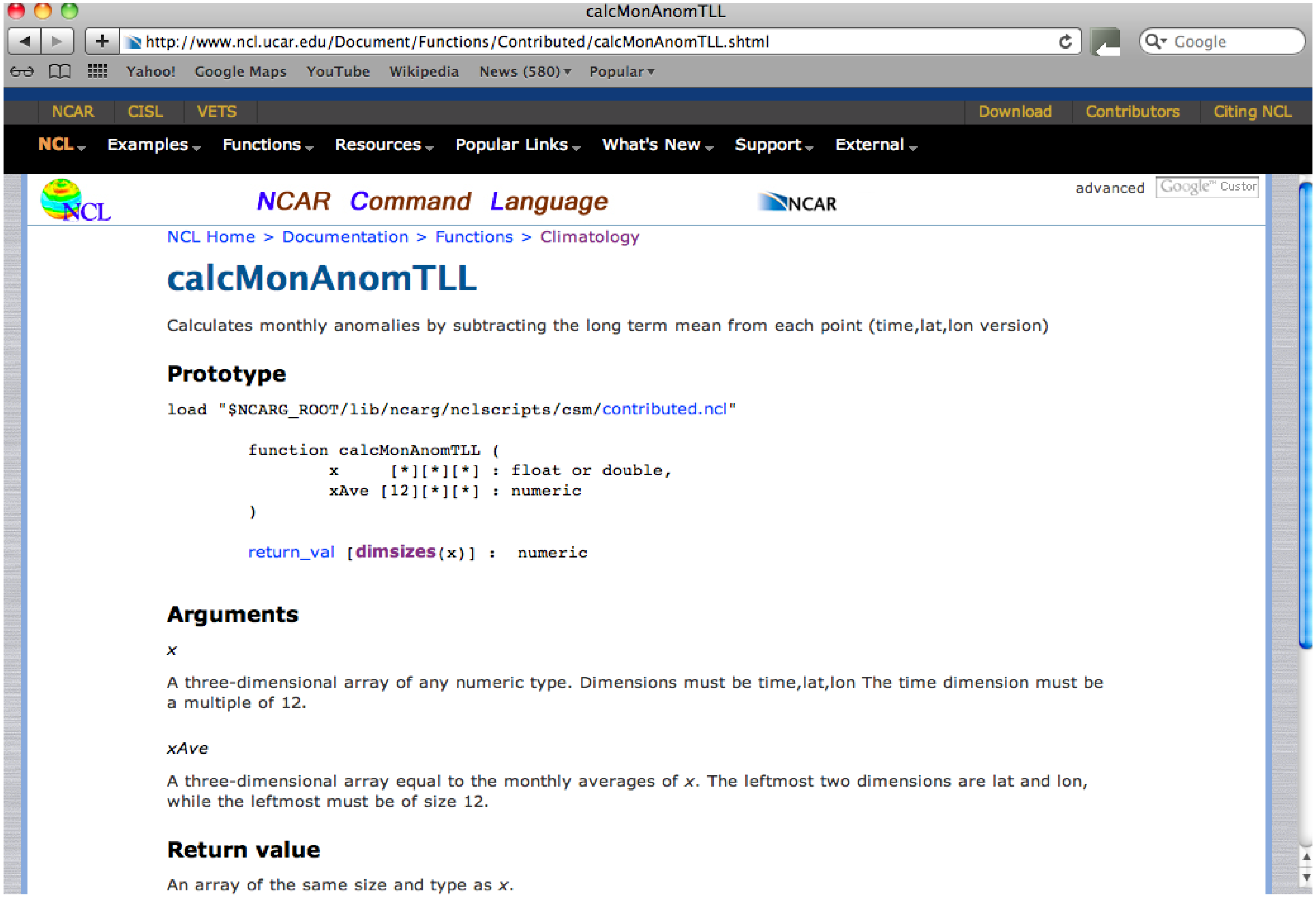
## Alphabetical listing of NCL Functions

Category **listing** | Function type **listing** | Browsable **listing**

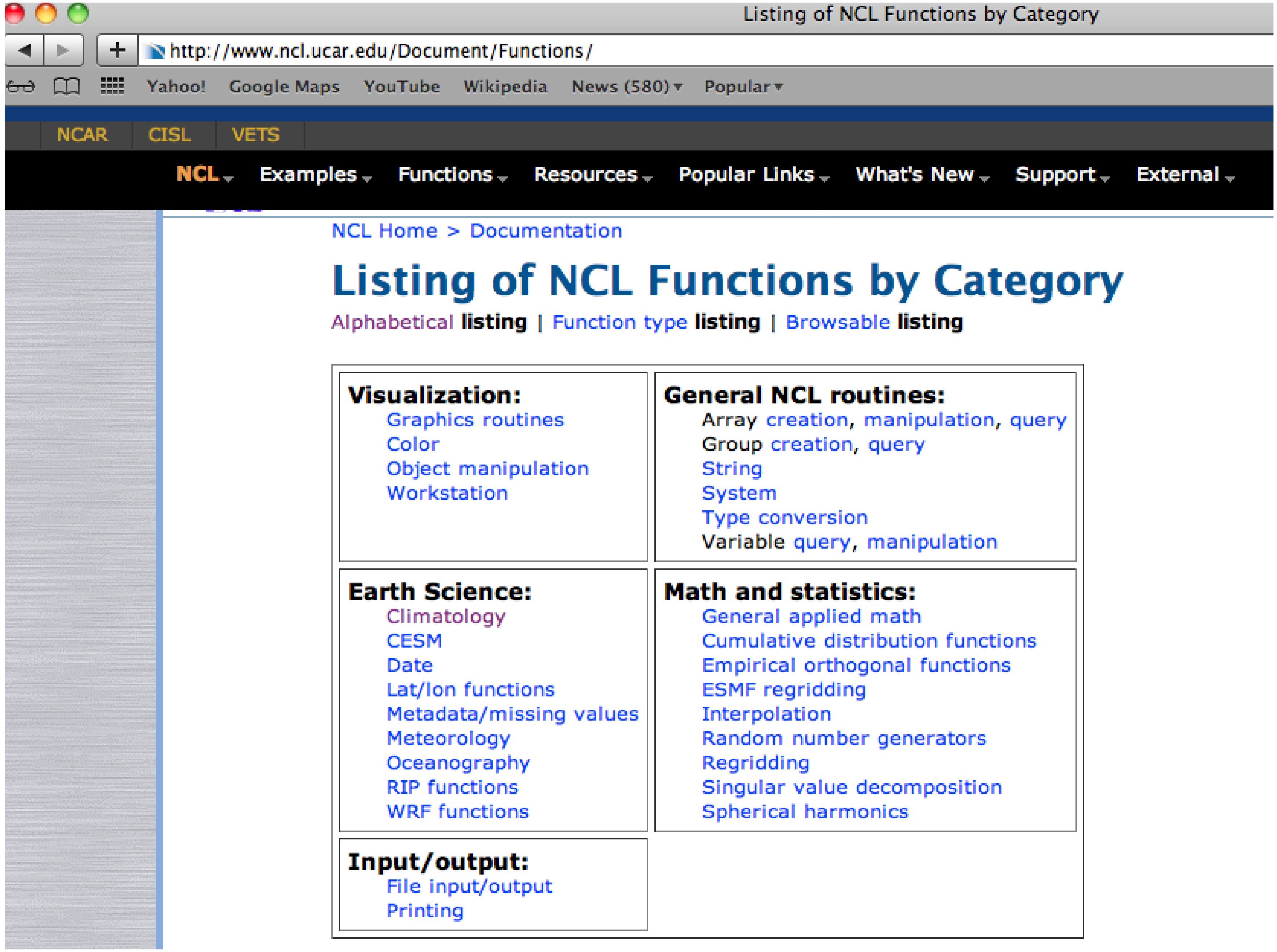
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

| | |
|---|---|
| abs | Returns the absolute value of numeric data. |
| acos | Computes the inverse cosine of numeric types. |
| add90LatX | Adds two fake pole points (90S and 90N) to the rightmost dimension of the given data. |
| add90LatY | Adds two fake pole points (90S and 90N) to the leftmost dimension of the given data. |
| addfile | Opens a data file that is (or is to be) written in a supported file format. |
| addfiles | Creates a reference that spans multiple data files. |
| addfiles_GetVar | Creates a reference that spans multiple data files and returns metadata. (deprecated: see **addfiles**) |
| all | Returns True if all the elements of the input evaluate as True. |
| angmom_atm | Calculates the atmosphere's relative angular momentum. |

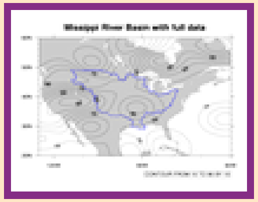
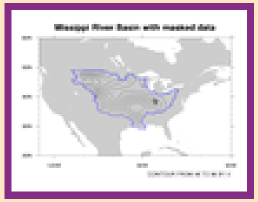
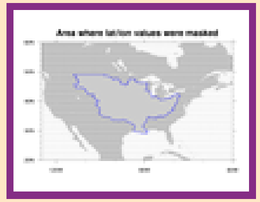
# The NCL website: NCL Function



# The NCL website: Function By Category



# The NCL website: Examples

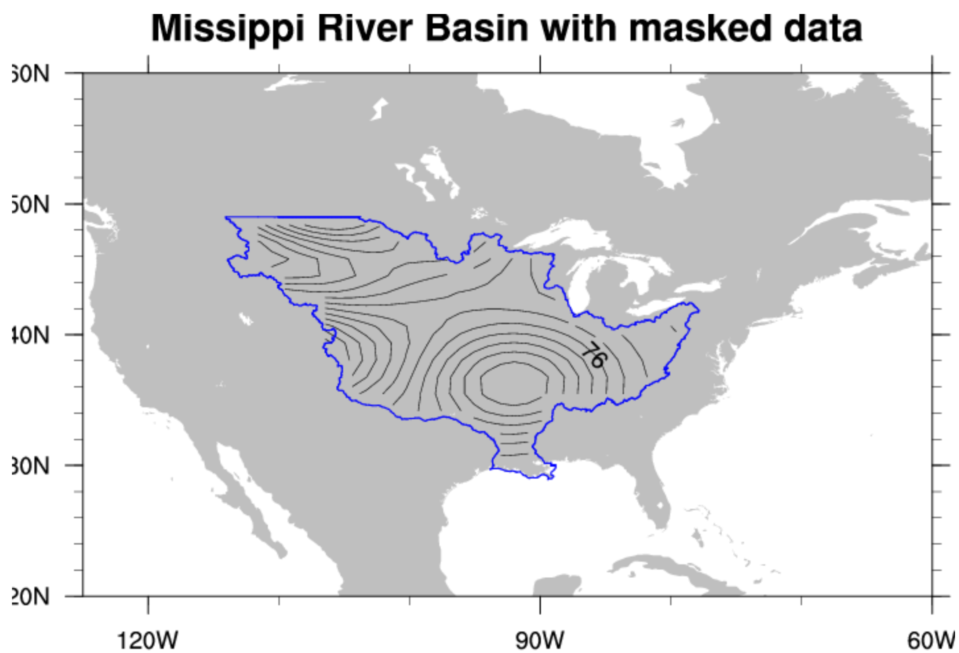
shapefiles_4.ncl: Demonstrates using **gc_inout** to mask an area in your data array using a geographical outline.

This particular example reads a shapefile to get an outline of the Mississippi River Basin. You then have the option of masking out all areas inside or outside this outline.

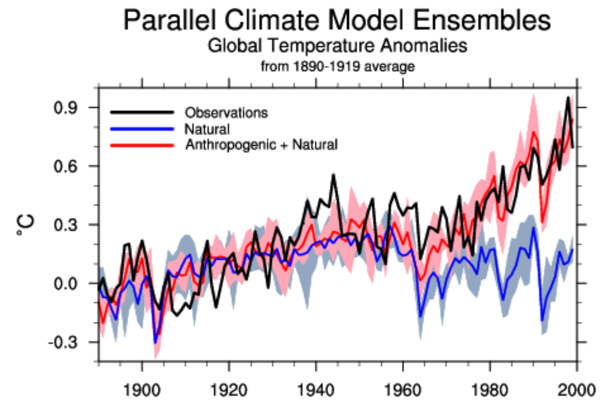
The "mrb.*xxx*" data files for this example can be found on the example datasets page.

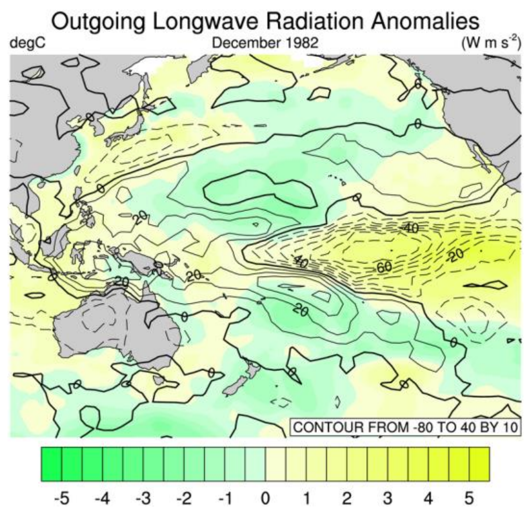
## NCL Help

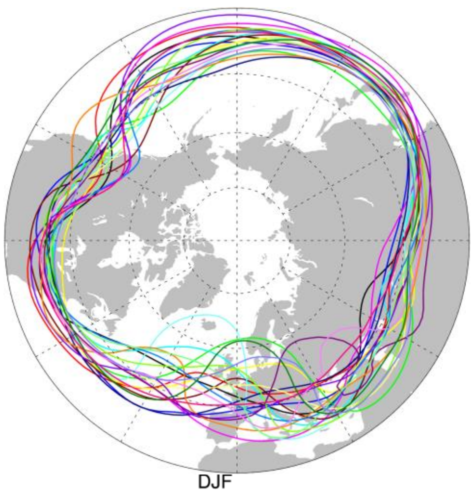
ncl-talk@ucar.edu

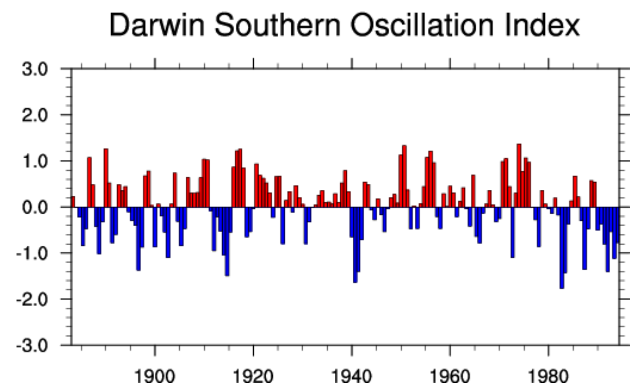


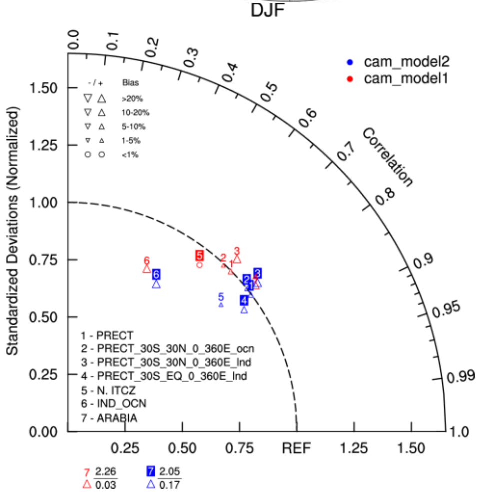
# NCL Example Graphics

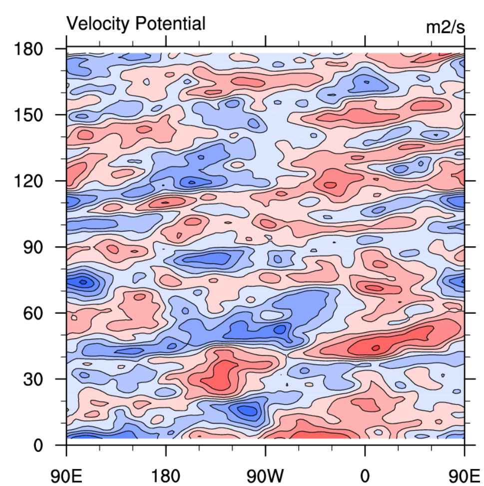


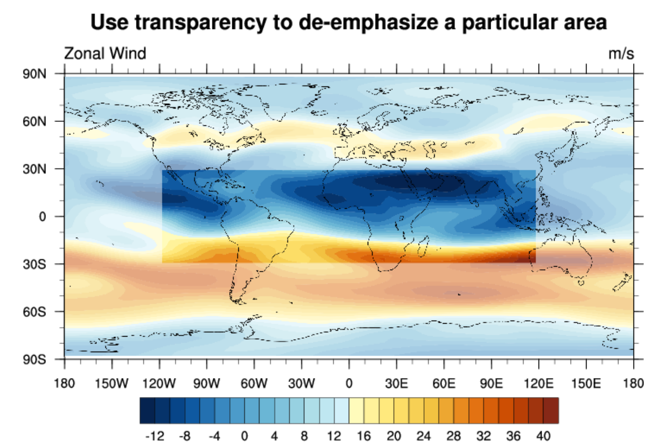



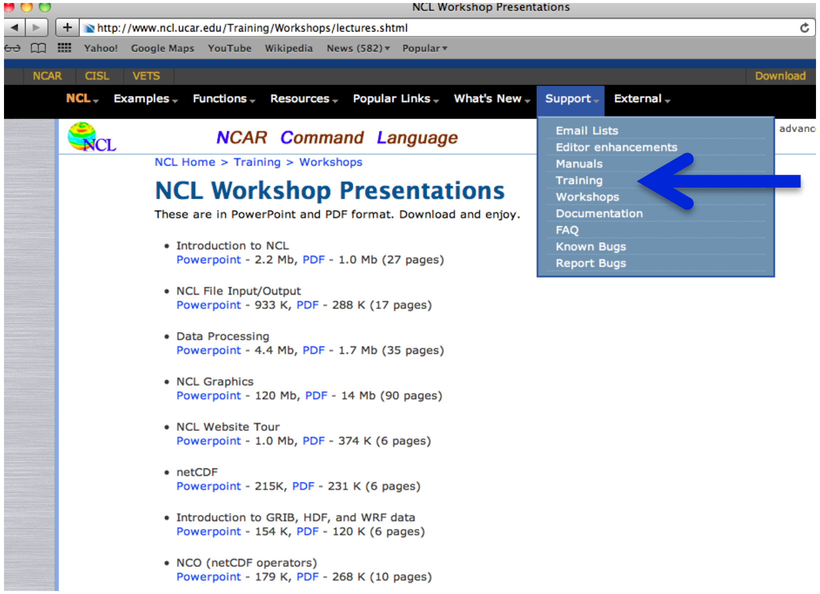



# NCL

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/



# Today’s Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# Exercise in NCL

(1) Aggregating variable across several different files into a single local variables

(2) Climatology Plot, overlaying standard deviation over the climatology plot

(3) Difference map between two climates

(4) Time series/seasonal cycle for a region

(5) Time series/seasonal cycle at a point

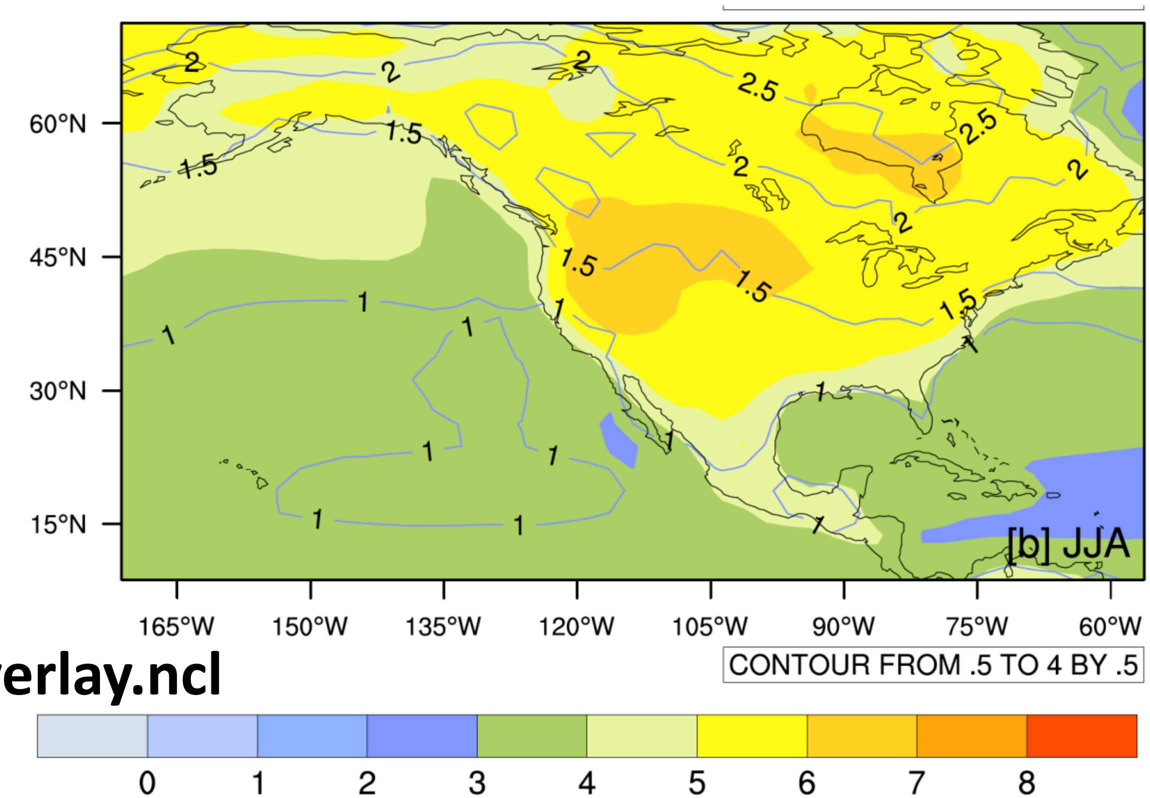
(6) Comparing global model runs and point scale runs with a point scale observations (Univ. of Mich. Biological Station)

Note: Your are provided with NCL scripts for each of these tasks. Copy NCL scripts to your home directory otherwise you may not be able to modify. Instruction are provided in the Day3_Cheat_Sheet.docx

# NCL Exercise 1

**NCL Script 1**
**climatology_plot1.ncl**

- Aggregating the data
- Climatology Calculations
- Climatology Plot
- plot resources




**NCL Script 2**
**climatology_plot1_std_overlay.ncl**

- Plotting contour lines
- Plot overlaying

# Aggregating the data


Step1: Make the list of files

```
flis = systemfunc("ls "+modInDir+"/"+caseName+".clm2.h0.*.nc")
print(flis)
```

```
CLM2014Tutorial_5yr1850test.clm2.h0.0001-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0001-12.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0002-12.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0003-12.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0004-12.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-01.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-02.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-03.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-04.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-05.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-06.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-07.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-08.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-09.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-10.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-11.nc
CLM2014Tutorial_5yr1850test.clm2.h0.0005-12.nc
```

Step2: Add these files

```
f1 = addfiles(flis, "r")
ListSetType(f1, "cat")
print(f1)
```

Step3: Import into a local variable

```
var1 = f1[:]->TSA
printVarSummary(var1)
```

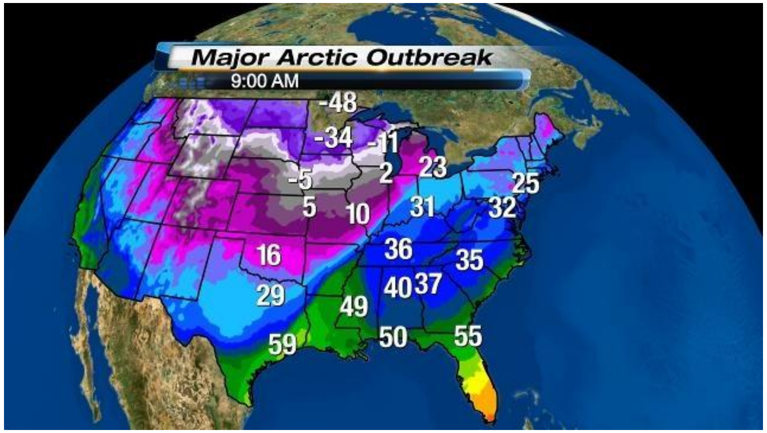
```
Variable: var1
Type: float
Total Size: 3317760 bytes
            829440 values
Number of Dimensions: 3
Dimensions and sizes:   [time | 60] x [lat | 96] x [lon | 144]
Coordinates:
            time: [31..1825]
            lat: [-90..90]
            lon: [ 0..357.5]
Number Of Attributes: 5
  long_name :   2m air temperature
  units :       K
```

# Climatology Calculation

Monthly average and standard deviation

```
var1_mon_mean = clmMonTLL(var1)
printVarSummary(var1_mon_mean)

var1_mon_std = stdMonTLL(var1)
printVarSummary(var1_mon_std)
```




Annual average and standard deviation

```
var1_ann = var1(0:4, :, :)  ;***this way new variables have necessary dimensions e.g. 5 years x lat x lon
                            ;*** you can change time dimensions depending upon the number of years for your run
do yearnum = 0, 4
   stmonth= yearnum*12
   endmonth= stmonth + 11
   var1_ann(i, :, :) = (/dim_avg_n_Wrap(var1(stmonth:endmonth, :, :), 0)/)
end do

var1_ann_avg = dim_avg_n_Wrap(var1_ann, 0)     ;**** _Wrap is necessary because that way it also copies the metadata
printVarSummary(var1_ann_avg)

var1_ann_std = dim_stddev_n_Wrap(var1_ann, 0)
printVarSummary(var1_ann_std)
```

# Climatology Plot

- Open a work station
- Define plot resources
- Use the appropriate function to plot a two dimensional data (lat X lon)

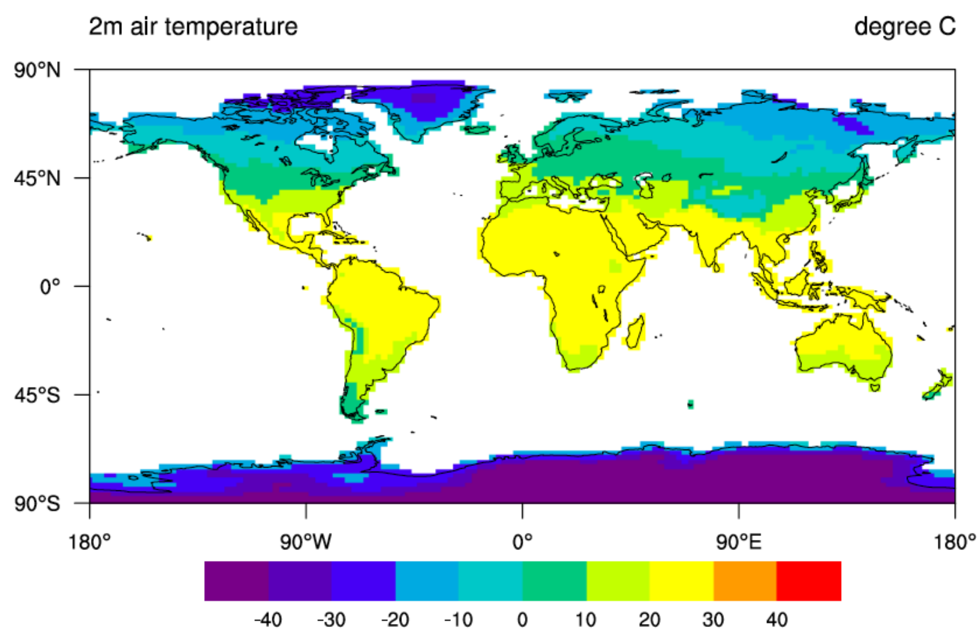
```
wks = gsn_open_wks("x11", "climatology_plot1")
gsn_define_colormap(wks, "cosam")
res = True
…
res@mpOutlineBoundarySets = "Geophysical”
res@mpGeophysicalLineColor = "black”
res@mpUSStateLineColor = "black”
….
….

plot = gsn_csm_contour_map(wks, var1_ann_avg, res)
```

## Extra Credit Work

- Overlay standard deviation on mean
- Draw a contour line plot for standard deviation
- Overlay the contour line with mean climate plot

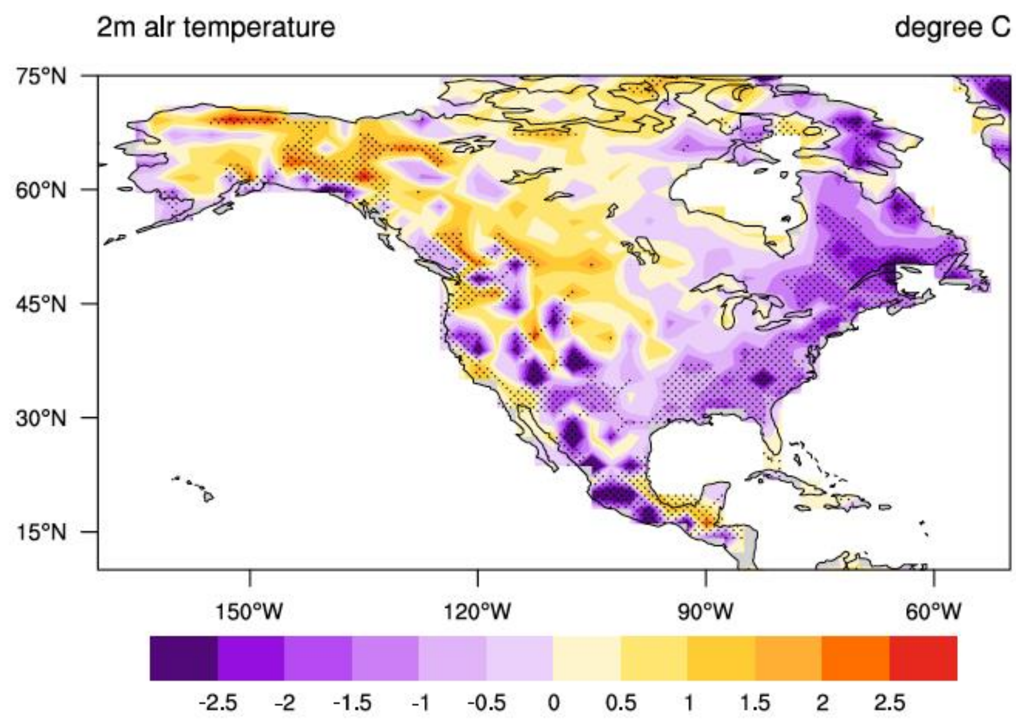
climatology_plot1_std_overlay.ncl



# Exercise 2

**NCL Script 3**
**Climatology_diff_plot1_sig_overlay.ncl**

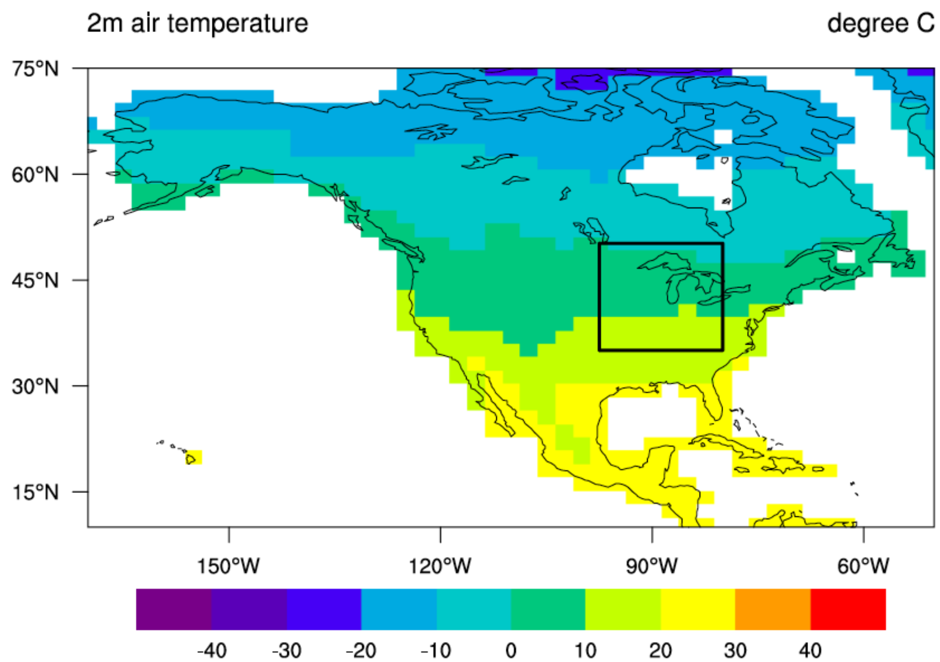
- Importing two data sets
- Calculating difference
- Calculating difference's statistical significance using student t-test [extra credit work]
- Plotting the difference
- Overlaying statistical significance on the difference plot [extra credit work]



# Exercise 3

**NCL Script 4**
**climtology_plot1_regional_extract.ncl**

- Determining the region's extent
- Plotting area polygon on the map
- Calculating area average for a given region
- Writing a text file

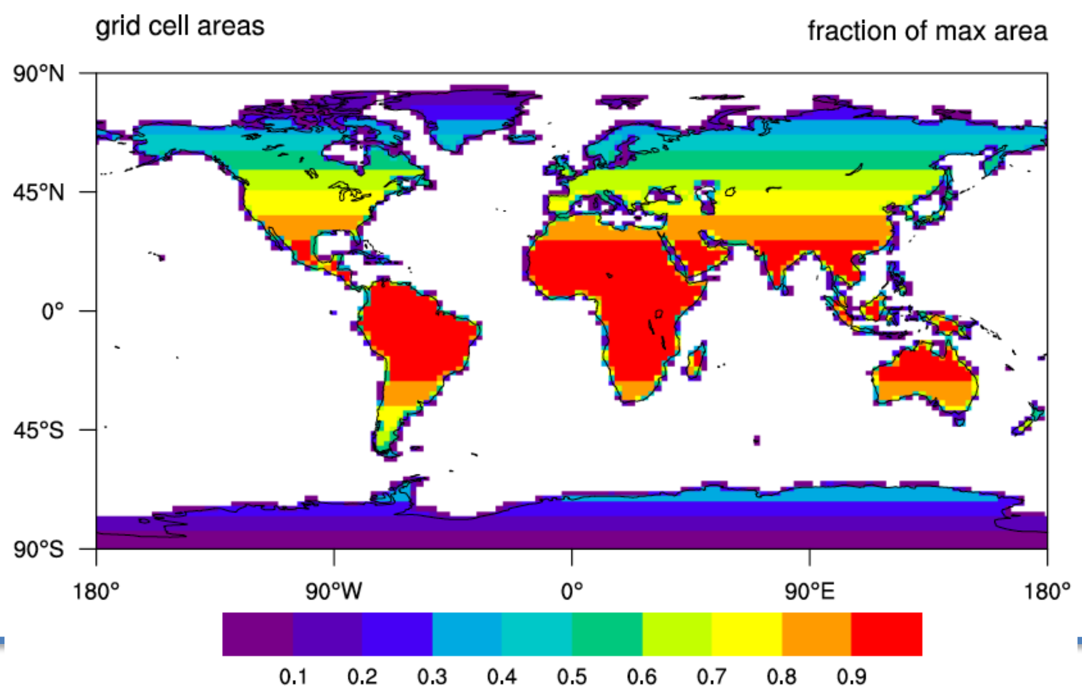


# Area Average Calculation

**Step 1**

- Getting area and land fraction for each grid cell
- Making area weight for each grid cell

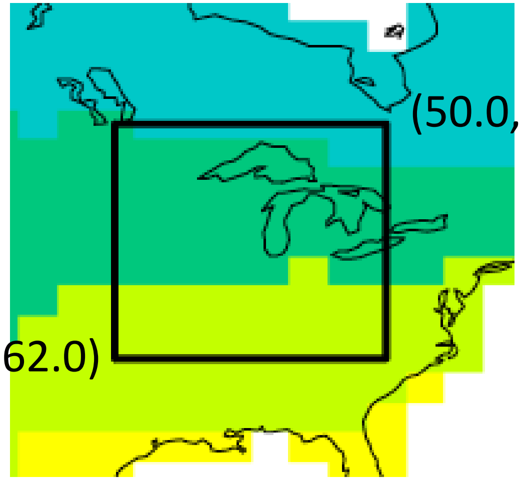
```
f3 = addfile(modInDir+"/"+caseName+".clm2.h0.0001-01.nc", "r")
ar = f3->area
lf = f3->landfrac
arwt = ar      ; create a weight variable with the meta data
arwt = ar * lf ; incorprates coatal areas land fraction effects
arwt = arwt / max(arwt) ; normalizes by maximum value (0-1)
```



# Area Average Calculation

## Step 2

- Define your region of interest
- Calculate area average using "wgt_aeraave2" fuction




```
nReg = 1

lonRegL = 262.0
lonRegR = 280.0

latRegL = 35.0
latRegH = 50.0

P1 = new((/12, 2/), "float", -9999.0)
P1(:, 0) = wgt_areaave2(var1_mon_mean(:, {latRegL:latRegH}, {lonRegL:lonRegR}), arwt({latRegL:latRegH}, {lonRegL:lonRegR}), 0)
P1(:, 1) = wgt_areaave2(var1_mon_std(:, {latRegL:latRegH}, {lonRegL:lonRegR}), arwt({latRegL:latRegH}, {lonRegL:lonRegR}), 0)
```

# Writing a text file

- Writes two dimensional formatted text file (nice looking)
- Text file name and format/precision are specified (by you) in the script
- Easy to open in excel

```
opt1 = True
opt1@fout = "mw_usa_temp_mon_climo.txt"
fmtx = "2f15.8"
write_matrix(P1, fmtx, opt1)
```

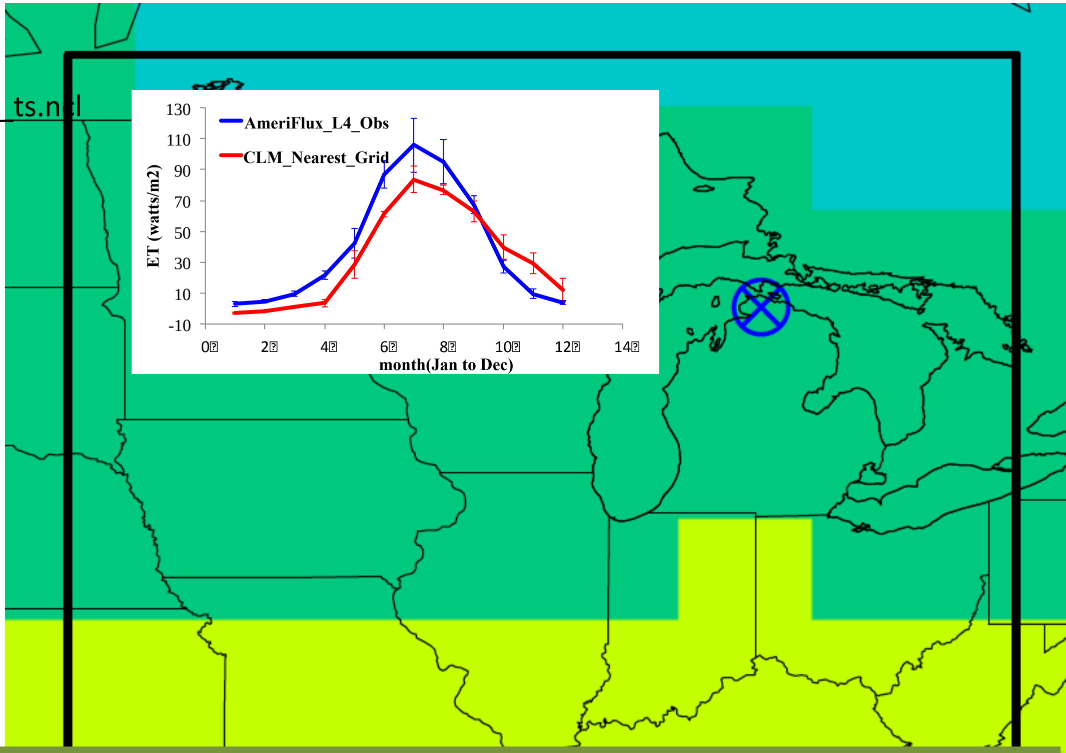
```
 6.22234201      2.66885042
-4.53278351      1.88590217
 0.00369003      1.05542839
 8.57405186      2.27336407
15.04171562      0.93946189
20.34946251      1.39838469
22.61991310      1.31776381
21.26465225      1.25616324
16.55312920      1.31188774
10.58961678      1.92898667
 2.74516439      2.23010802
-3.37048554      2.13963842
```

# Exercise 4

## NCL Script 5

climatology_plot1_regional_extract_point_location_monthly_ts.ncl

- Lat and Lon for the point
- Plotting station marker on the map
- Index for nearest grid point
- 'do loop' for arranging




Science Question: Scaling issue – point observations versus gridded model output

Home Work : Compare it with point scale CLM run

# Getting nearest grid point data

```
;*****************Seasonality Extract from Point Location
 nSt = 1
 StLat = 45.5598
 StLon = 275.2862

 ilat = ind_nearest_coord(StLat, lat, 0)
 ilon = ind_nearest_coord(StLon, lon, 0)

 print("ilat="+ilat+" StLat ="+StLat+" lat(ilat) = "+lat(ilat))
 print("ilon = "+ilon+" StLon = "+StLon+" lon(ilon) = "+lon(ilon))

 P2 = new((/12, 2/), "float", -9999.0)
 P2(:, 0) = var1_mon_mean(:, ilat, ilon)
 P2(:, 1) = var1_mon_std(:, ilat, ilon)
```

First get lat and lon for the station

Find nearest lat and lon indices

Print nearest lat and lon indices

Store data corresponding to the point location in a new 2D array

Print P2 to a text file

# do loop in NCL

```
obsName = "AMF_USUMB"
obsVersion = "L4_h_V002"
obsDir = "/glade/p/cesm/lmwg/ObsData/TowerData/"+obsName
AnaYr = (/"2000", "2001", "2002", "2003", "2004"/)
mon_Ndat = (/31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/)
mon_Sday = (/0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334/)
P3 = new((/12, 5/), "float", -9999.0)

do yearnum = 0, 4
   obsFileName = obsName+"_"+AnaYr(yearnum)+"_"+obsVersion+".nc"
   print(obsFileName)
   fo = addfile(obsDir+"/"+obsFileName, "r")
   varObs1 = fo->Le_f
   printVarSummary(varObs1)
   do monthnum = 0, 11
     sti = mon_Sday(monthnum) * 48
     eni = sti + mon_Ndat(monthnum) * 48 -1
     P3(monthnum,yearnum) = (/dim_avg_n(varObs1(sti:eni), 0)/)
   end do
   delete(fo)
   delete(varObs1)
   delete(obsFileName)
end do

 P4 = P1
 P4(:, 0) = dim_avg_n(P3, 1)
 P4(:, 1) = dim_stddev_n(P3, 1)
```

- Flux Tower data (observations) are at half hourly time step
- Compare monthly model outputs with flux tower data

**Tired of writing the Matlab/R/NCL scripts!!**
**Help is on the way .....**

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# Land Model Diagnostic Package

Kieth Oleson

# LMWG Diagnostics Package (latest version: lnd_diag4.2.26)

- The land diagnostics package post-processes land history files into monthly, seasonal, and annual climatologies and means and produces a series of plots using NCL that gives you a comprehensive look at your simulation(s).

- An overview, instructions on how to obtain the package, and a User's Guide can be found at:

http://www.cesm.ucar.edu/models/cesm1.2/clm/clm_diagpackage.html

- The diagnostics package can either be run in Swift or Non-Swift mode.
  - Swift is a software package that allows for task multi-processing. Turning Swift on can significantly reduce the wall clock time needed to run the diagnostic package.
  - BUT, Swift currently only works with the NCAR "geyser" machine.

# After You Get Home

- If you wish to use the diagnostics package and if you have access to the NCAR “geyser” machine, follow the provided instructions for setting up your environment to run the Swift version of the diagnostics package. (see the document lnd_diag_swift4.2.XXXXXX.pdf available on the land diagnostics web page). If you have problems you can contact me at oleson@ucar.edu

- If you wish to install the land diagnostics package on your home institution’s computers, you or your system administrator will need to have the netCDF operators (NCO), NCL, perl, and the “convert” tool from ImageMagick installed. You will also need to modify the lnd_template4.2.26.csh script to work on your local machine.

- If you wish to use Swift on your home institution’s computers, make sure the non-Swift version works on your computer, then contact me (I will contact somebody who knows how to install it).

# LMWG Diagnostics Package

- An example run script that compares two of the tutorial simulations (2000SP – 1850SP) (this is called a model to model comparison) can be found at:

/glade/p/cesm/lmwg/CLM2014_tutorial_space/Day3/Diag/Script/lnd_template4.2.26_CLM2014Tutorial.csh

- Netcdf files are created that contain the climatologies and means. The netcdf files for the tutorial simulations can be found at:

/glade/p/cesm/lmwg/CLM2014_tutorial_space/Day3/Diag/OUTPUT/Tcase2_prefix
/glade/p/cesm/lmwg/CLM2014_tutorial_space/Day3/Diag/OUTPUT/Tcase1_prefix

- An html file containing plots from the land diagnostics package for the tutorial simulations can be found at:

/glade/p/cesm/lmwg/CLM2014_tutorial_space/Day3/Diag/OUTPUT/Tcase2_prefix/Tcase2_prefix-Tcase1_prefix/setsIndex.html

Tcase2_prefix: CLM2014Tutorial_20yr2000SP_n02_clm4_5_57_11-20
Tcase1_prefix: CLM2014Tutorial_20yr1850SP_n02_clm4_5_57_11-20

# LMWG Diagnostics Package

**Typical work flow** to run the land diagnostics package on “geyser”:

1) Create a lnd_diag directory in your home directory, cd to lnd_diag, create a run directory, cd to run, and put the template script (lnd_template4.2.26.csh) in there (you only have to do this once):

```
cd /glade/u/home/<logname>
mkdir lnd_diag
cd lnd_diag
mkdir run
cd run
cp /glade/p/cesm/lmwg/diag/lnd_diag4.2/lnd_template4.2.26.csh .
```

3) Open up the file lnd_template4.2.26.csh using your favorite text editor and modify for your simulations:

```
vi  lnd_template4.2.26.csh
```
(or use nedit, xemacs, etc.)

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

```
./lnd_template4.2.26.csh  >&!  lnd.out  &
```

Note: When submitting the diagnostics scripts, the syntax “>&!” is used. What this means: > = pass the on-screen output to a file, & = pass standard error/out to the file, ! = overwrite the existing output file if necessary. The second & runs the job in the background

# LMWG Diagnostics Package

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

6) Once the diagnostics script has successfully completed (you should get an email), a tar file (*.tar; containing html files and plots in *.gif format) will have been created in the specified directory

7) cd to this directory and untar the tar file:

cd /glade/p/cesm/lmwg/CLM2014_tutorial_space/Day3/Diag/OUTPUT/Tcase2_prefix

tar xvf *.tar

8) cd to the untarred directory, fire up a browser window, and point to the setsIndex.html file

cd Tcase2_prefix-Tcase1_prefix
firefox &
In firefox: File->Open File…->setsIndex.html

Tcase2_prefix: CLM2014Tutorial_20yr2000SP_n02_clm4_5_57_11-20
Tcase1_prefix: CLM2014Tutorial_20yr1850SP_n02_clm4_5_57_11-20

# Today’s Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

Break

⑧ Help Session/Exercise <60 minutes>

# Today's Goal

① Where are model outputs? <10 minutes>

② Knowing about the data (netCDF, ncview, ncdump) <20 minutes>

③ Analysis in Matlab <30 minutes>

④ Analysis in R <30 minutes>

Break

⑤ Introduction to NCL (**N**CAR **C**ommand **L**anguage) <10 minutes>

⑥ Analysis in NCL <30 minutes>

⑦ Running Land Model Diagnostic Package <30 minutes>

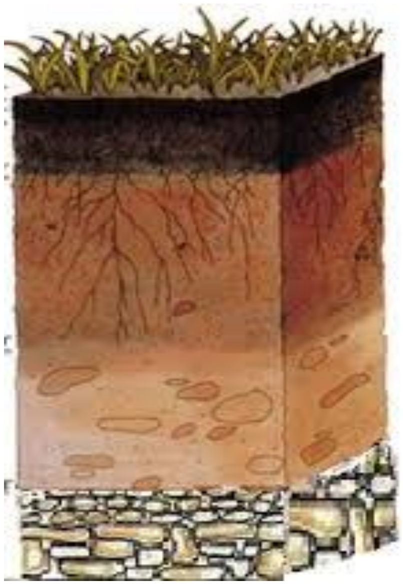
Break

⑧ Help Session/Exercise <60 minutes>

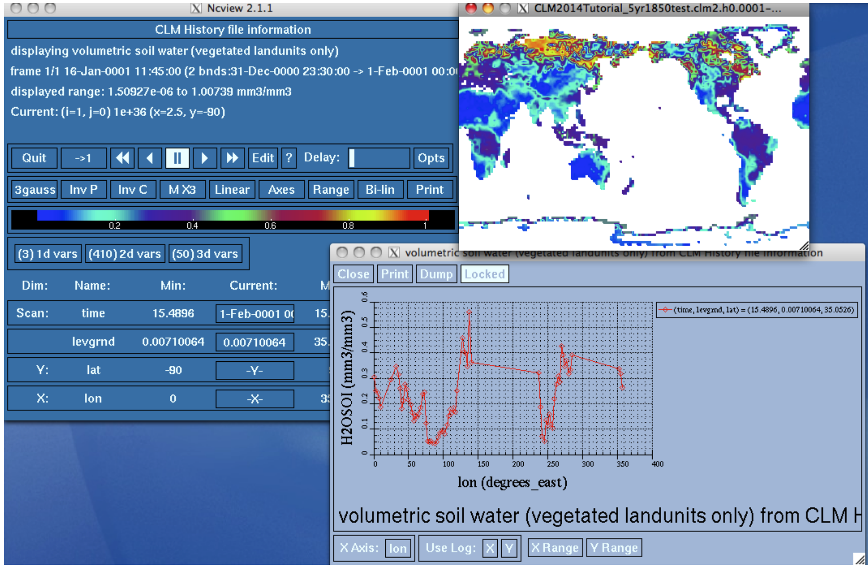
# Exercise or Help Session!

# Exercise for netCDF Variable

- Print Variable Summary for soil moisture (H2OSOI)
- Is there an extra dimension in H2OSOI?
- Explore this extra dimension (coordinates, size, etc.)



# Exercise in ncview




- ◆Change the variable on the plot
- ◆Click on any location on the plot
- ◆See XY plot
- ◆Change x – axis in the plot

# Extra Credit Question

- ◆ Plot depth profile of soil moisture (H2OSOI)

# Exercise in R

1. Open a .nc file
2. Extract a variable and convince R that it's actually a map of the whole globe
3. Plot it
4. Read in a variable from a time series (bunch of files)
5. Calculate point-wise mean and standard deviation
6. Calculate the hottest year for each gridcell
7. Clip out and look at a region (Australia)
8. Quit R

# Exercise in NCL

(1)Aggregating variable across several different files into a single local variables

(2)Climatology Plot, overlaying standard deviation over the climatology plot

(3)Difference map between two climates

(4)Time series/seasonal cycle for a region

(5)Time series/seasonal cycle at a point

(6)Comparing global model runs and point scale runs with a point scale observations (Univ. of Mich. Biological Station)

Note: Your are provided with NCL scripts for each of these tasks. You may want to take help of these scripts while writing your own script