ARW WRF Graphical Packages
ARWpost ; NCL and others

Cindy Bruyère

ARWpost

Converter: creates

GrADS ***(.dat)*** and ***(.ctl)*** files, or
Vis5D input files ***(.v5d)***

general

- **Download Code:**
  - http://www.mmm.ucar.edu/wrf/users/download/get_source.html
- **OnLine Tutorial:**
  - http://www.mmm.ucar.edu/wrf/users/graphics/ARWpost/ARWpost.htm

general

- **MUST have WRFV2 compiled**
  (similar to WPS)

- **IF Vis5D files are being created,**
  - Vis5D libraries needed for compilation
  - Vis5D is free and can be downloaded from: http://www.ssec.wisc.edu/~billh/vis5d.html

compilation

./configure

Make sure this is correct. If not, set environment variable NETCDF

```
Will use NETCDF in dir: /usr/local/netcdf-pgi
-----------------------------------------------
Please select from among the following supported
platforms.
1. PC Linux i486 i586 i686, PGI compiler (no
vis5d)
2. PC Linux i486 i586 i686, PGI compiler (vis5d)
3. PC Linux i486 i586 i686, Intel compiler (no
vis5d)
4. PC Linux i486 i586 i686, Intel compiler
(vis5d)

Enter selection [1-4] :
```

compilation

- **configure.arwp,** will be created
- If your WRFV2 code is not compiled under **../WRFV2,** edit **configure.arwp,** and set **"WRF_DIR"** to the correct location of your WRFV2 code
- **./compile**

  This will create **ARWpost.exe**

namelist.ARWpost

- **set times to be processed** *(&datetime)*
  Start & End dates
  Interval in seconds
  (times must be in wrfout file - tacc in next version)

- **set input and output file names and variables to be processed** *(&io)*

- **set levels to interpolate too** *(&interp)*

namelist.ARWpost *(&io)*

io_form_input	2=netCDF, 5=GRIB1
input_root_name	**Path** and **root** name of files to use as input. *Do not use wild characters* *Do not only provide directory name*
output_root_name	Output root name. *output_root_name***.dat** & *output_root_name***.ctl**, OR *output_root_name***.v5d**
output_type	Options are 'grads' (*default*) or 'v5d'
mercator_defs	Set to true if mercator plots are distorted

namelist.ARWpost *(&io)*

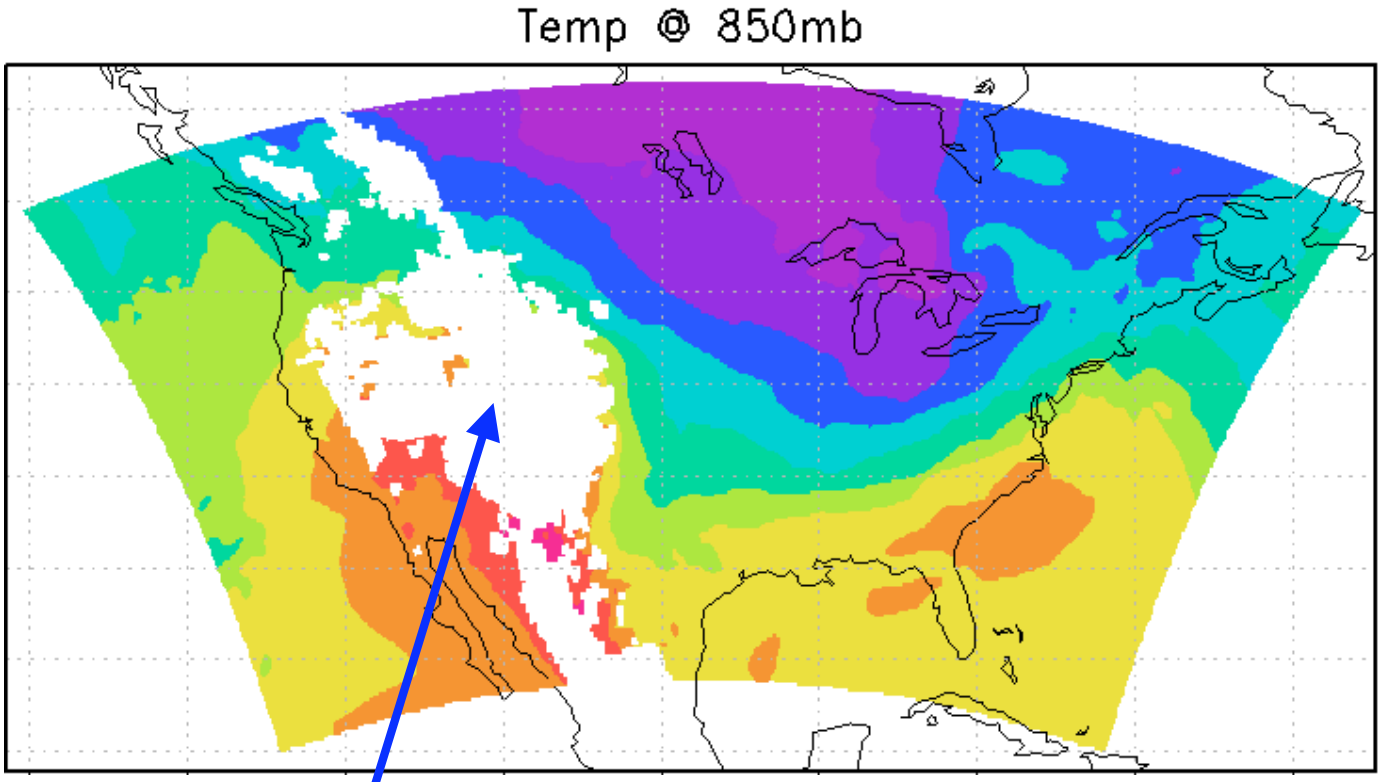
plot	Which fields to process. Options are: **all, basic, list, file, basic_file, basic_list, list_file, all_file, all_list, basic_list_file, all_list_file** Order has no effect, i.e., "all_list" and "list_all" are similar. "**list**" - a list of variables must be supplied under "**fields**" "**file**" - a list of variables must be added to a file, and the filename supplied under "**fields_file**"
fields	Fields to plot. Only used is **list** was used in the "**plot**" variable.
fields_file	File name that contains list of fields to plot. Only used is **file** was used in the "**plot**" variable.

***Available diagnostics*:** height ,theta ,tc, tk, td, rh, umet, vmet, pressure, dbz, max_dbz, u10m, v10m, slp, mcape, mcin, lcl, lfc, cape, cin

namelist.ARWpost *(&interp)*

interp_method	0 = sigma levels, -1 = code defined "nice" height levels, 1 = user defined height or pressure levels
interp_levels	Only used if interp_method=1 Supply levels to interpolate to, in hPa (pressure) or km (height above sea level) Supply levels bottom to top **NOTE**: *NO extrapolation below ground*

interpolation



NO extrapolation below ground

run

- **./ARWpost.exe**
- **Will create either,**

***output_root_name*.dat & *output_root_name*.ctl**

OR

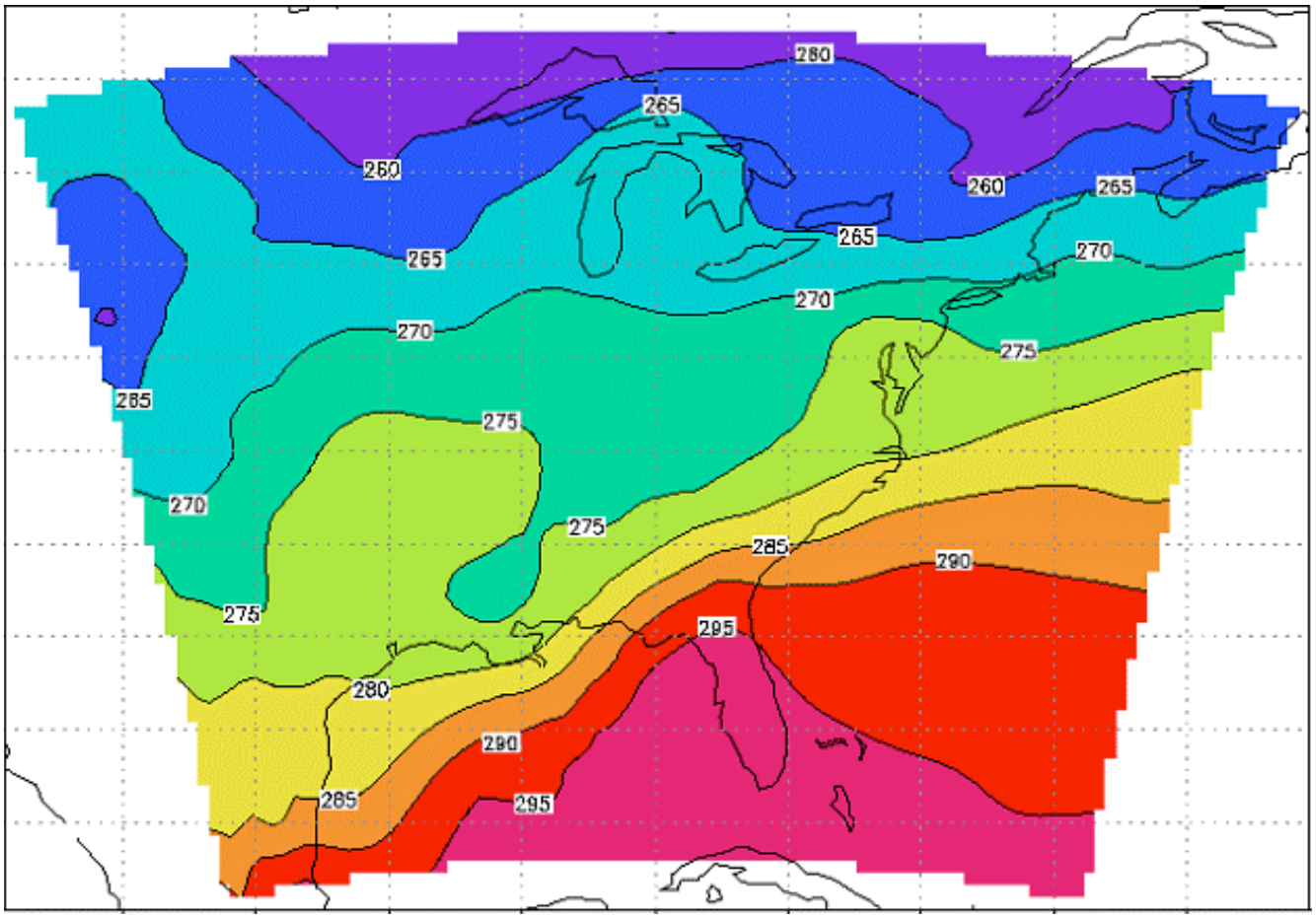
***output_root_name*.v5d**

GrADS specific notes

- **To display images requires GrADS software**
  freely available from
  http://grads.iges.org/grads/grads.html

- **Documentation:**
  http://grads.iges.org/grads/gadoc/index.html

GrADS - projections



GrADS - .ctl file

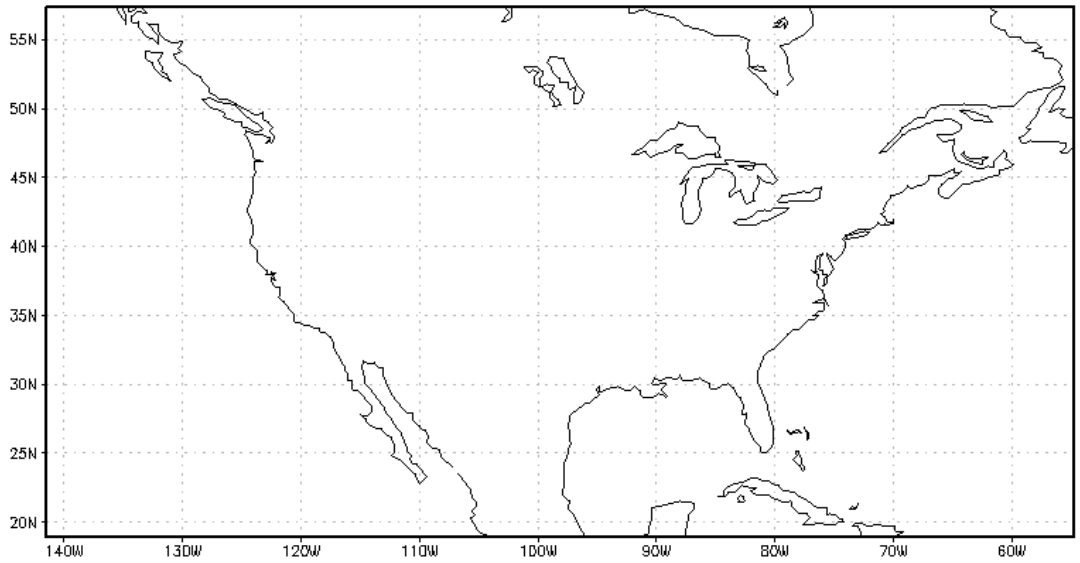
```
dset ^test.dat
options byteswapped
undef 1.e37
title  OUTPUT FROM WRF V2.2 MODEL
pdef  259 163 lcc  40.000  -98.000  130.000   82.000
      60.00000  30.00000  -98.00000  22000.000  22000.000
xdef  877 linear -141.49254   0.09909910
ydef  389 linear   18.88639   0.09909910
```

options byteswapped

Needed on some machines - if you get NaNs when you plot, ***remove*** *this line from .ctl file*

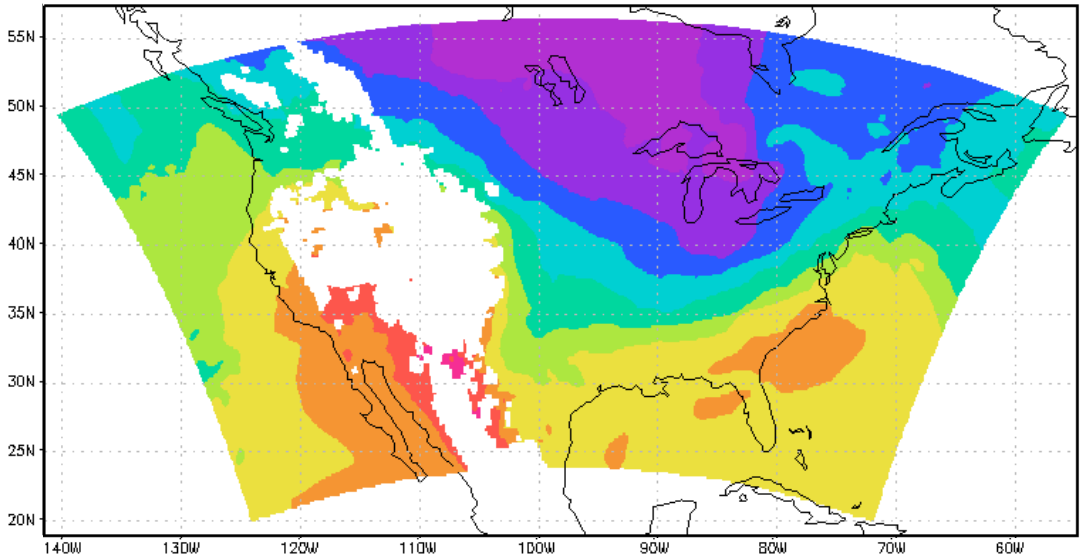
GrADS - .ctl file

```
dset ^test.dat
options byteswapped
undef 1.e37
title  OUTPUT FROM WRF V2.2 MODEL
pdef  259 163 lcc  40.000  -98.000  130.000   82.000
      60.00000  30.00000  -98.00000  22000.000  22000.000
xdef  877 linear -141.49254   0.09909910
ydef  389 linear   18.88639   0.09909910
```



GrADS - .ctl file

dset ^test.dat
options byteswapped
undef 1.e37
title OUTPUT FROM WRF V2.2 MODEL
pdef 259 163 lcc 40.000 -98.000 130.000 82.000
60.00000 30.00000 -98.00000 22000.000 22000.000
xdef 877 linear -141.49254 0.09909910
ydef 389 linear 18.88639 0.09909910

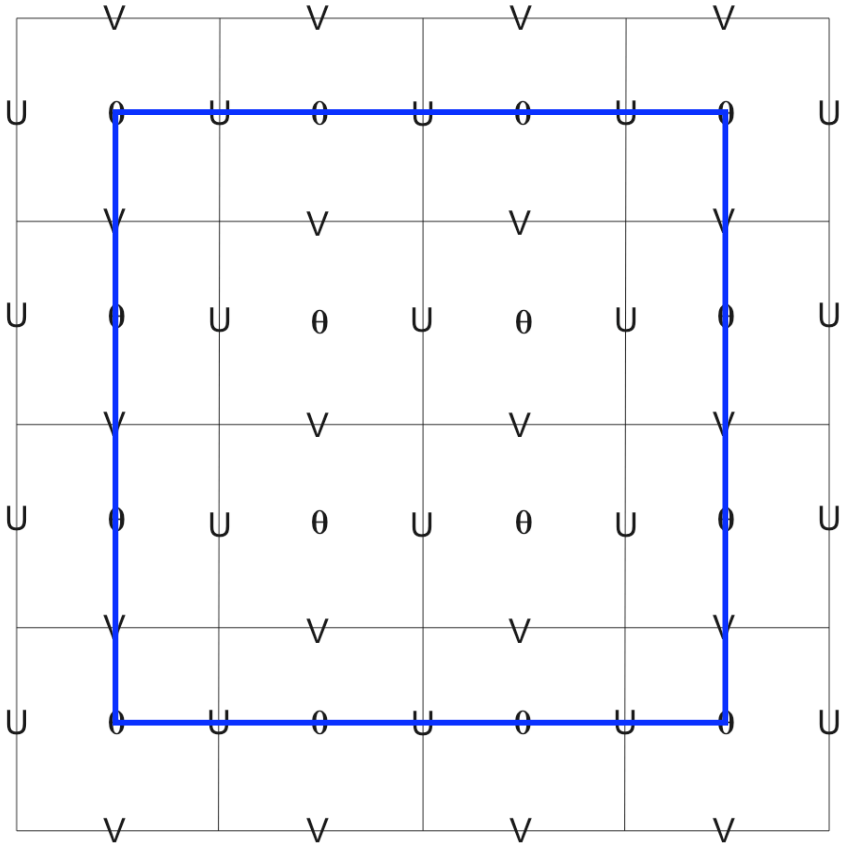


GrADS conversion - question

- **Why is a converter needed if GrADS can display netCDF files?**
  - Can only display model surface coordinates
  - Cannot interpolate to height or pressure levels
  - All diagnostics must be added via GrADS script files
  - *GRIB1 model output can also be read directly by GrADS, but above issues are still valid*
  - *For GRIB1, there is also a stagger problem*

GrADS conversion - question

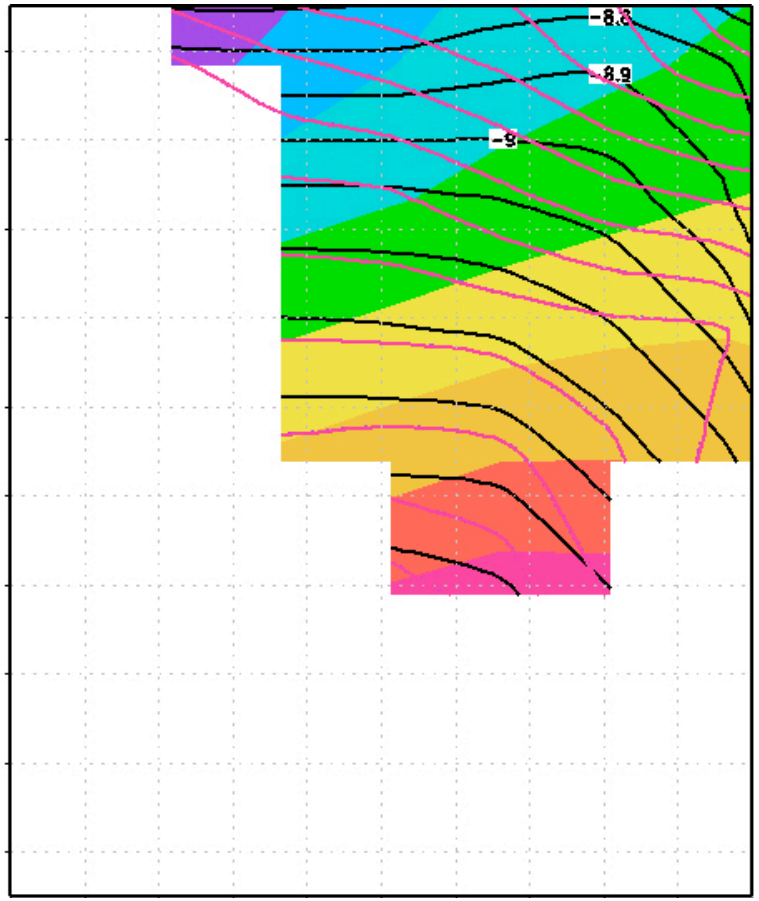
- **Why is a converter needed if GrADS can display netCDF files?**

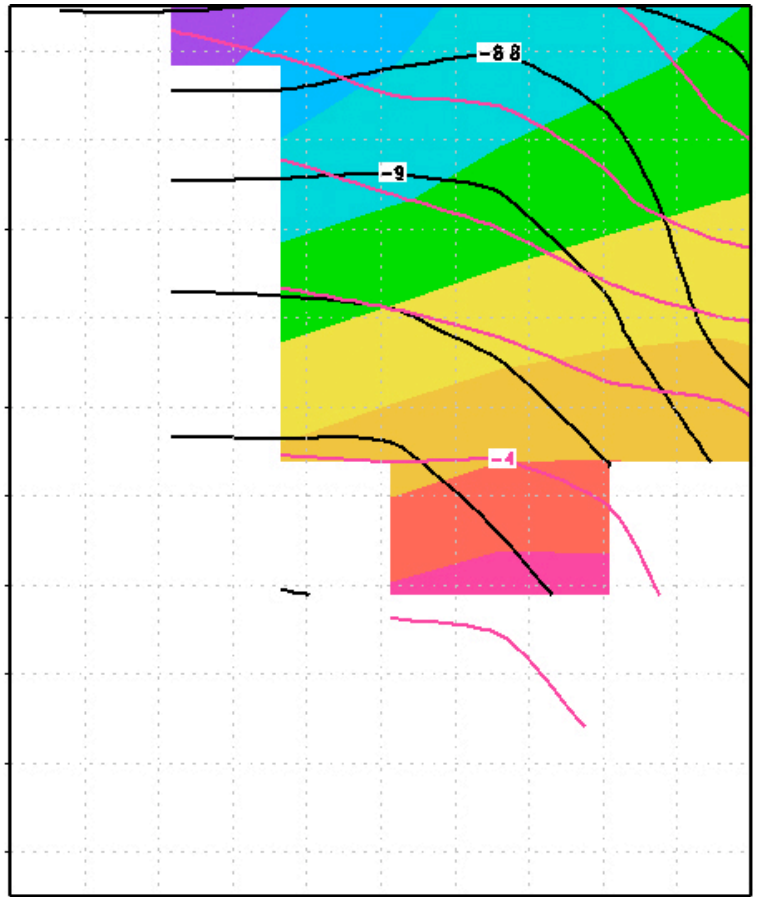


WRF staggered grid

staggering

shaded=T ; black=U ; red=V



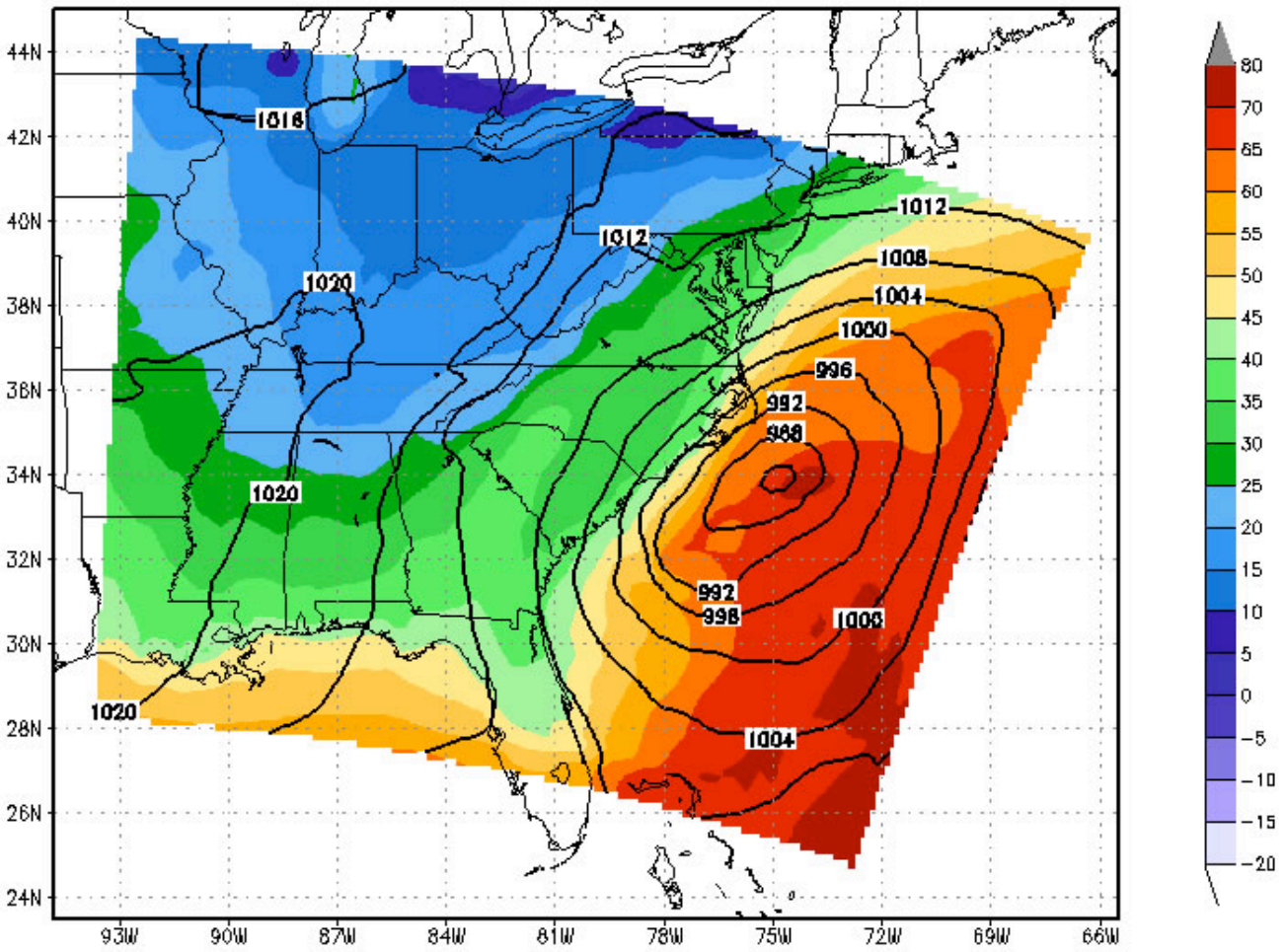


staggering

- **Since GrADS version 1.9**
  - a new **gradsnc** interface is available *created by GrADS developers for WRF*
- **To USE**
  - must create 4 .ctl files (M ; U ; V ; W)
  - must open the all at once
- **Utility**
  - ARWpost/util/WRFnc2ctl.f

creating a plot

Surface Temperature (F)
Sea Level Pressure (mb)



creating a plot

```
open em_real.ctl
set mpdset hires
set display color white

define tf=1.8*tc + 32
set gxout shaded
set z 1
d tf
run cbar.gs

set gxout contour
set ccolor 1
set cint 4
d slvl
```

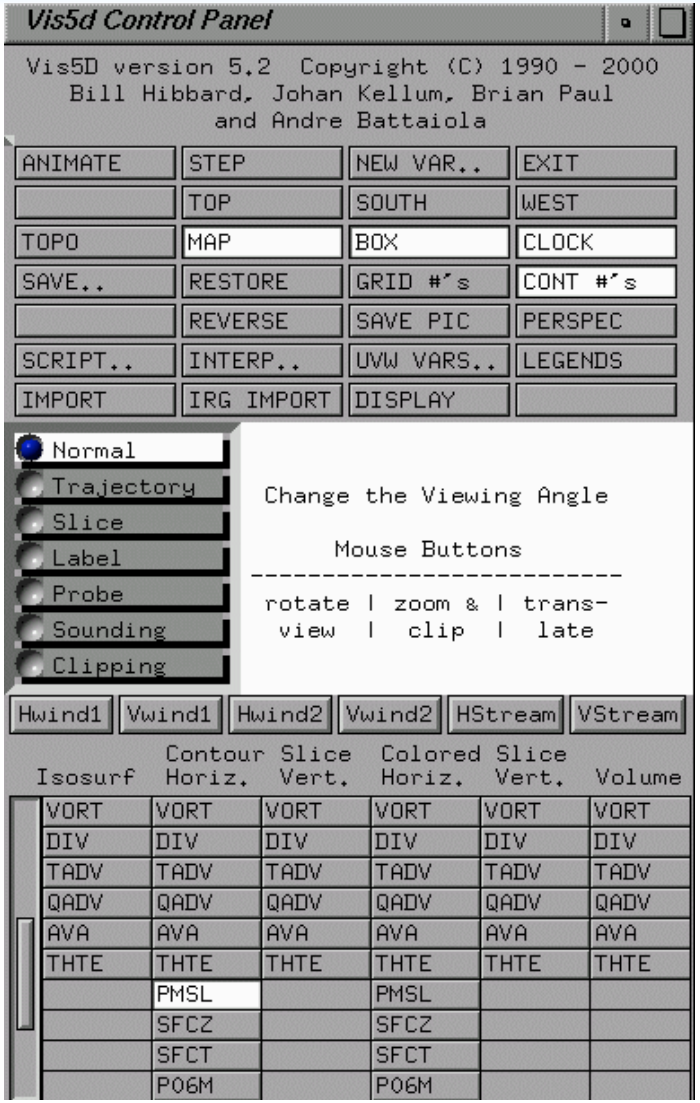
vis5d specific notes

- **vis5d is a three-dimensional visualization software**
- **vis5d is free and can be downloaded from:** http://www.ssec.wisc.edu/~billh/vis5d.html
- **Run**

  vis5d *output_root_name*.v5d

- **Graphical Interface**

vis5d graphical interface

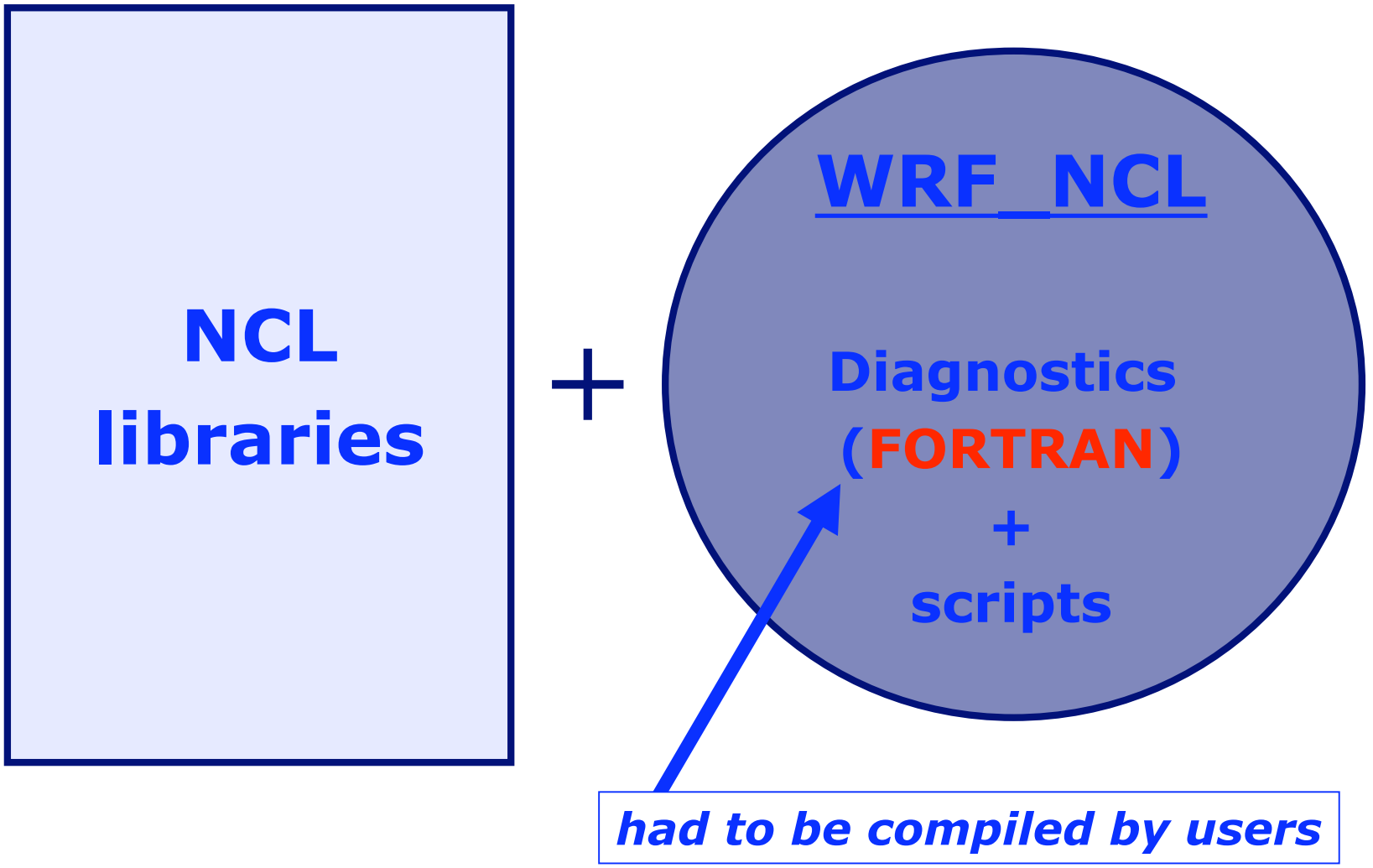


NCL

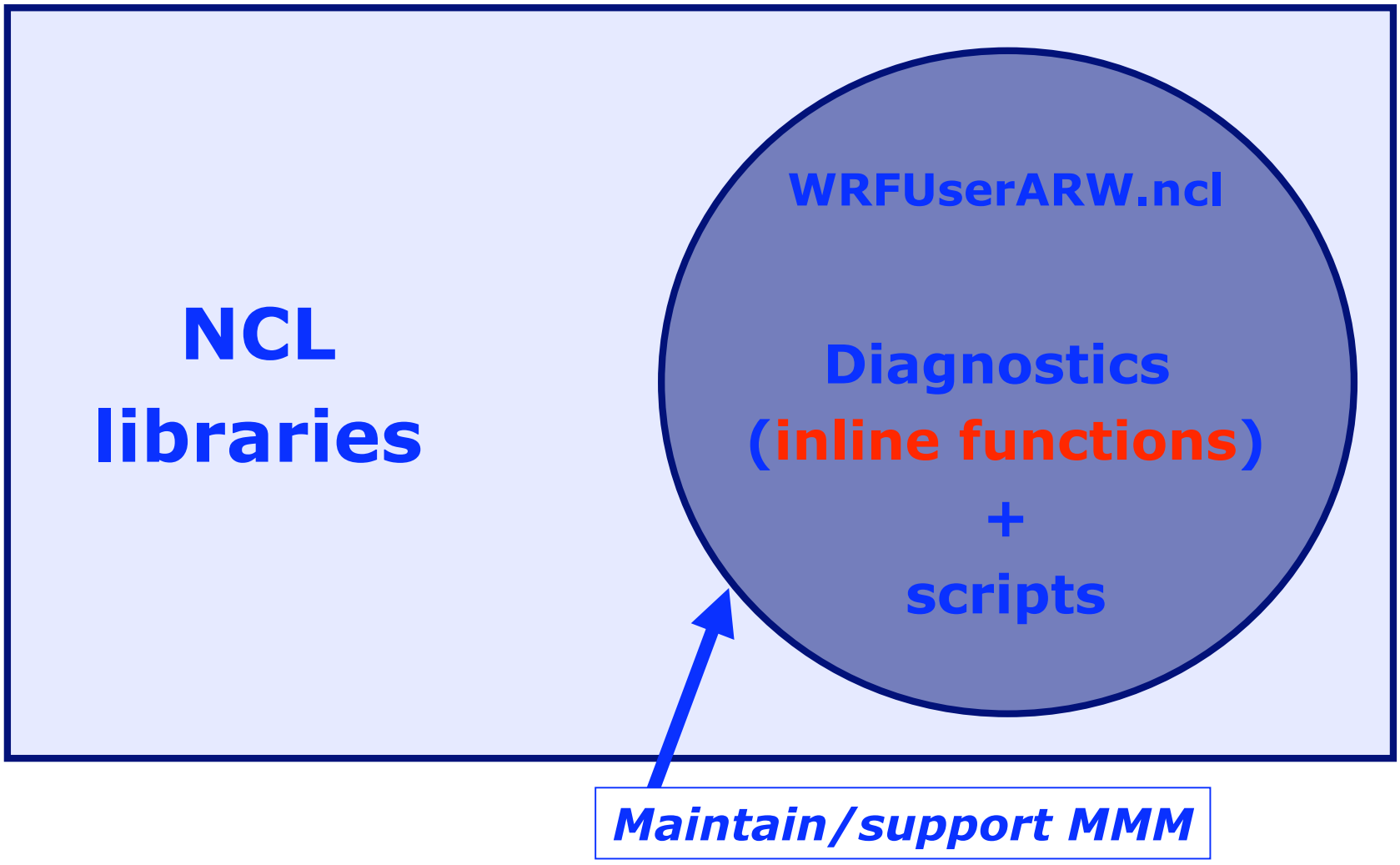
NCL stands for *NCAR Command Language*

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

NCL & WRF *(before July 2007)*



NCL & WRF *(July 2007 - ncl v4.3.1)*



download NCL

- **Go to:**
  - **http://www.ncl.ucar.edu/Download/**
    Read and agree to GPL license
    Fill out short registration form
    Download binaries
- **Set NCARG_ROOT environment variable:**
  - *setenv NCARG_ROOT /usr/local/ncl*
- ***NCL version 4.3.1 or 5.0.0***

Home Directory → ~/.hluresfile ← Very Important

- **Required by NCL libraries**
- **Must be in your "~/" directory** ***(home directory)***
- **Control**
  - color table ; font
  - white/black background
  - size of plot
  - control characters

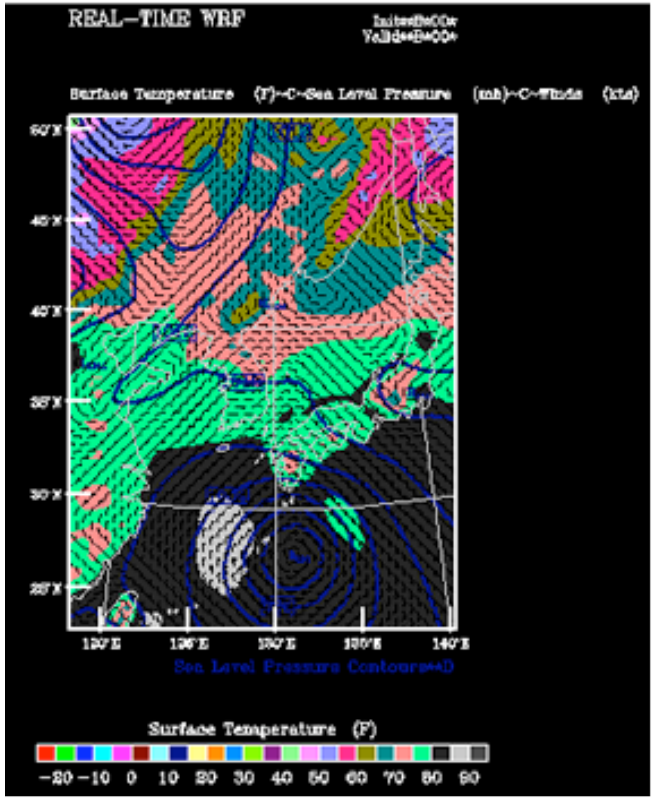
- ***http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml***

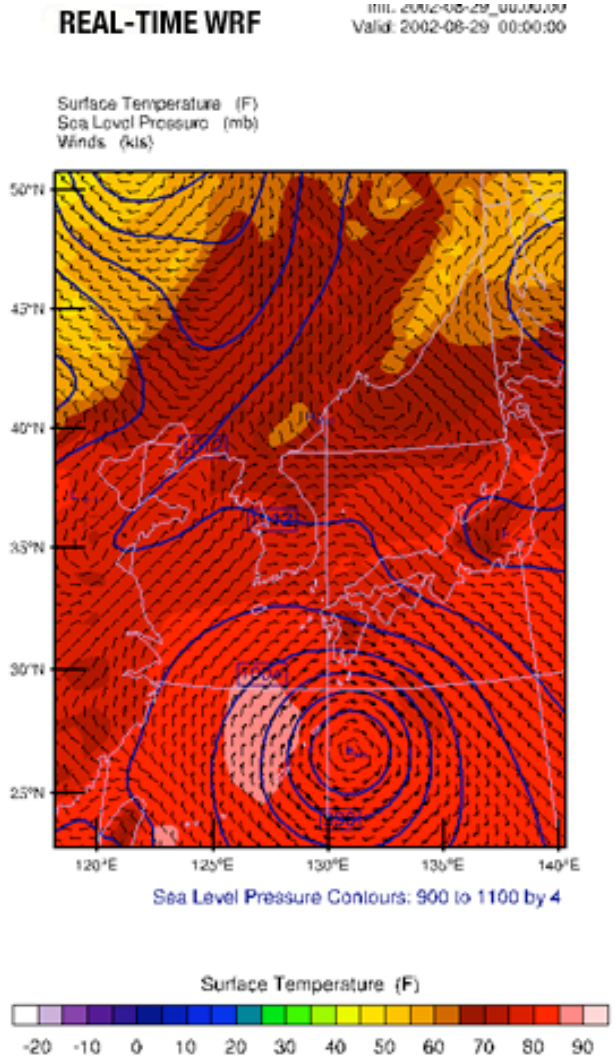
~/.hluresfile

```
*wkColorMap              : WhViBlGrYeOrReWh
*wkBackgroundColor       : white
*wkForegroundColor       : black
*FuncCode                : ~
*TextFuncCode            : ~
*Font                    : helvetica
*wkWidth                 : 900
*wkHeight                : 900
```

http://www.mmm.ucar.edu/wrf/users/graphics/NCL/.hluresfile

~/.hluresfile


REAL-TIME WRF
Surface Temperature (F)
Sea Level Pressure Contours
Surface Temperature (F)
-20 -10 0 10 20 30 40 50 60 70 80 90


REAL-TIME WRF
Valid: 2002-08-29 00:00:00
Surface Temperature (F)
Sea Level Pressure (mb)
Winds (kts)
50°N
45°N
40°N
35°N
30°N
25°N
120°E
125°E
130°E
135°E
140°E
Sea Level Pressure Contours: 900 to 1100 by 4
Surface Temperature (F)
-20 -10 0 10 20 30 40 50 60 70 80 90

Generate PLOTS

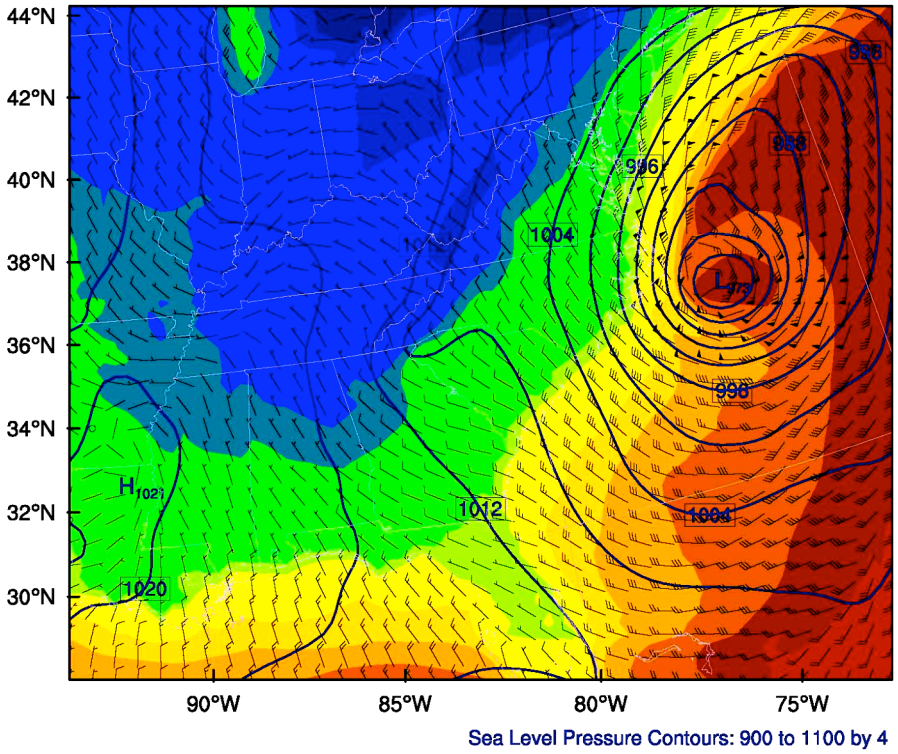
- **OnLine Tutorial:**

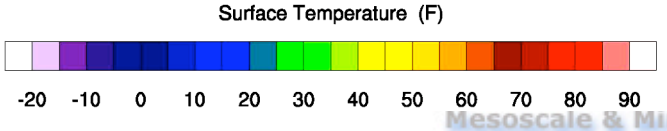
  http://www.mmm.ucar.edu/wrf/users/graphics/NCL/NCL.htm

Contain sample scripts

creating a plot

Surface Temperature (F)
Sea Level Pressure (mb)
Winds (kts)





creating a plot : NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"

begin

  ; Open graphical output
  ; Open input file(s)

  ; Read variables

  ; Set up plot resources & Create plots
  ; Output graphics

end
```

creating a plot : NCL script

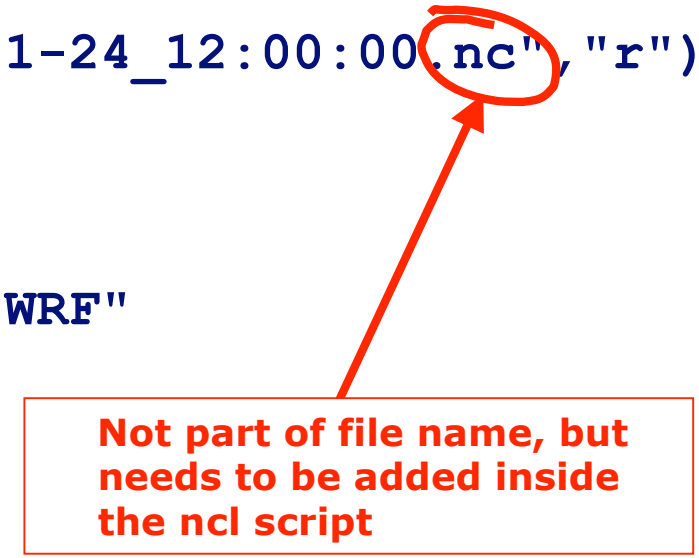
```
begin

; Open graphical output
 wks = gsn_open_wks("x11","wrfout")   ; x11, ps, pdf

; Open input file
 a = addfile("wrfout_d01_2000-01-24_12:00:00.nc","r")

 ; Set up plot resources
 ARWres = True
 ARWres@MainTitle = "REAL-TIME WRF"
 ARWres@Footer = False

 ; Create Map background
 mpres = True
 map = wrf_map(wks,a,mpres)
```



creating a plot : NCL script

```
; Read variables

slp = wrf_user_getvar(a,"slp",5)
tc2 = wrf_user_getvar(a,"T2",5)
      tf2 = 1.8 * (tc2-273.16) + 32.
      tf2@description = "Surface Temperature"
      tf2@units = "F"
u10 = wrf_user_getvar(a,"U10",5)
v10 = wrf_user_getvar(a,"V10",5)
      u10 = u10*1.94386
      v10 = v10*1.94386
      u10@units = "kts"
      v10@units = "kts"
```

Function from: WRFUserARW.ncl

Function will call NCL diagnostics if needed

Diagnostics:

uvmet, ua, va, wa, pressure, z, th, tk, tc, td, td2, slp, rh

creating a plot : NCL script

```
; Create plots
 opts_psl = ARWres
 opts_psl@cnLineColor          = "NavyBlue"
 c_slp = wrf_contour(a,wks,slp,opts_psl)

 opts_tc = ARWres
 opts_tc@cnFillOn              = True
 c_tc = wrf_contour(a,wks,tf2,opts_tc)

 opts_vct = ARWres
 opts_vct@FieldTitle           = "Wind"
 opts_vct@NumVectors           = 47
 vec = wrf_vector(a,wks,u10,v10,opts_vct)

 wrf_map_overlay(wks,map,(/c_tc,c_slp,vec/),False)

end
```

creating a plot : NCL script

- **Create a script**
  wrf_real.ncl

- **Set NCARG_ROOT environment variable:**
  setenv NCARG_ROOT /usr/local/ncl

- Ensure you have an ***~/.hluresfile*** file

- **Run NCL script**
  ncl wrf_real.ncl

Adding FORTRAN code

- **Can link NCL to existing FORTRAN code**
- **Code must be in F77**
- **Code must contain special *NCL START* and *END* markers**
- **Must create a shared object library from these routines in order for NCL to recognize the code.**

myTK.f

```
subroutine compute_tk (tk,pressure,theta, nx, ny, nz)
  implicit none
  integer nx,ny,nz
  real    pi, tk(nx,ny,nz)
  real    pressure(nx,ny,nz), theta(nx,ny,nz)


 integer i,j,k

  do k=1,nz
    do j=1,ny
    do i=1,nx
       pi=(pressure(i,j,k) 100000.)**(287./1004.)
       tk(i,j,k) = pi*theta(i,j,k)
    enddo
    enddo
  enddo

  end
```

myTK.f

```
C NCLFORTSTART
      subroutine compute_tk (tk,pressure,theta, nx, ny, nz)
        implicit none
        integer nx,ny,nz
        real    pi, tk(nx,ny,nz)
        real    pressure(nx,ny,nz), theta(nx,ny,nz)
C NCLEND

        integer i,j,k

        do k=1,nz
          do j=1,ny
          do i=1,nx
             pi=(pressure(i,j,k) 100000.)**(287./1004.)
             tk(i,j,k) = pi*theta(i,j,k)
          enddo
          enddo
        enddo

        end
```

myTK.so

WRAPIT myTK.f

setenv NCARG_ROOT */usr/local/ncl*

/usr/local/ncl/bin/WRAPIT myTK.f

cp /usr/local/ncl/bin/WRAPIT .
edit your copy
./WRAPIT myTK.f

myTK.so - use in NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
external myTK "./myTK.so"

begin

      t = wrf_user_getvar(a,"T",5)
      t = t + 300
      p = wrf_user_getvar(a,"pressure",5)

      dim = dimsizes(t)
      tk = new( (/ dim(0), dim(1), dim(2) /), float)

      myTK :: compute_tk (tk,p,t,dim(2),dim(1),dim(0))

end
```

other

- **Some of the more popular graphical tools used, not support by NCAR**
  - GEMPAK
  - IDL
  - MatLab

- **VAPOR**
  - supported by the vapor development group

VAPOR
Visualization and Analysis Platform for Oceanic, atmospheric and solar Research

Alan Norton
alan@ucar.edu
vapor@ucar.edu
National Center for Atmospheric Research

WRF in VAPOR

- Goal: enabling interactive visualization in support of analysis of WRF data
- WRF functionality has been added in v1.2
- Available free on Linux, Windows, Mac
- Installed on NCAR visualization systems
- Interactive rendering and animation (using GPU acceleration)
- Simple 2-step data conversion from WRF output to VAPOR
- Volume rendering
- Intuitive color/opacity editor
- Isosurface rendering

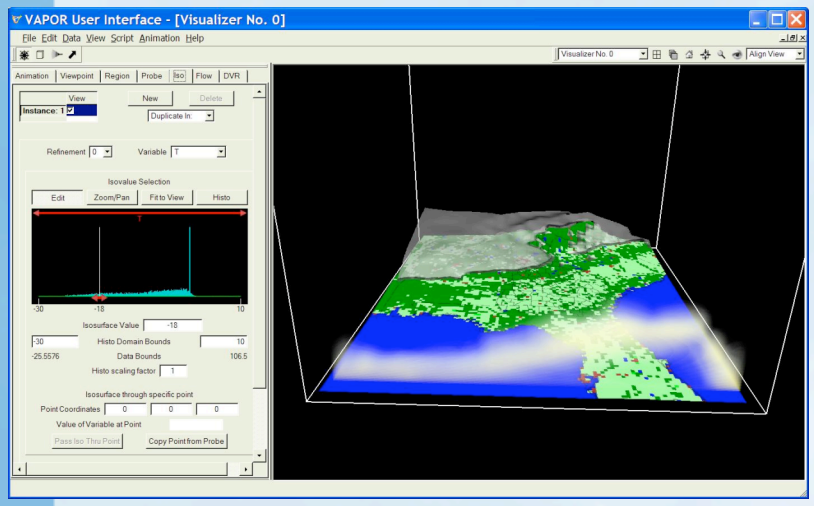
WRF in VAPOR

- Steady and unsteady flow integration
- Interactive seed placement
- Data probing
- Contour plotting
- Terrain surface image render
- Interactive performance on terabyte datasets

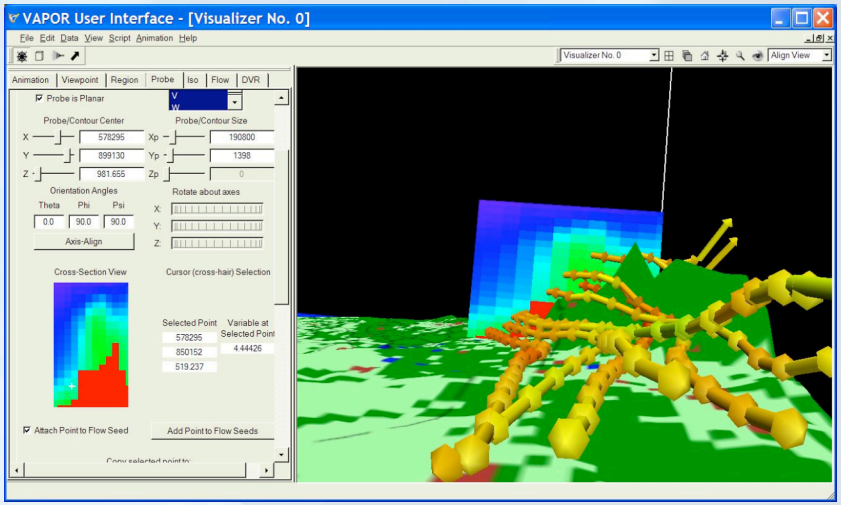
- Downloads, documentation, examples at: http://www.vapor.ucar.edu

- http://www.vapor.ucar.edu/doc/WRFsupport.pdf

WRF in VAPOR



Volume rendering with isosurface above terrain image



Interactive placement of seeds for particle tracing