



# **ARW WRF Graphical Packages**

## ***ARWpost & NCL***

*Cindy Bruyère*



# ARWpost

---

**Converter, creates**

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

# download ARWpost

---

- From wrf-model.org web site  
(*ARWpost.tar*):

[http://www.mmm.ucar.edu/wrf/users/download/get\\_source.html](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>

# code

---

README  
**arch/**  
**clean\***  
**compile\***  
**configure\***  
fields.plt  
gribinfo.txt  
**gribmap.txt**  
myLIST  
**namelist.ARWpost**  
scripts/  
src/  
util/

# 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*
- **set input and output file names and variables to be processed** (*&io*)
- **set levels to interpolate too** (*&interp*)

# namelist.ARWpost (&io)

---

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

# namelist.ARWpost (&io)

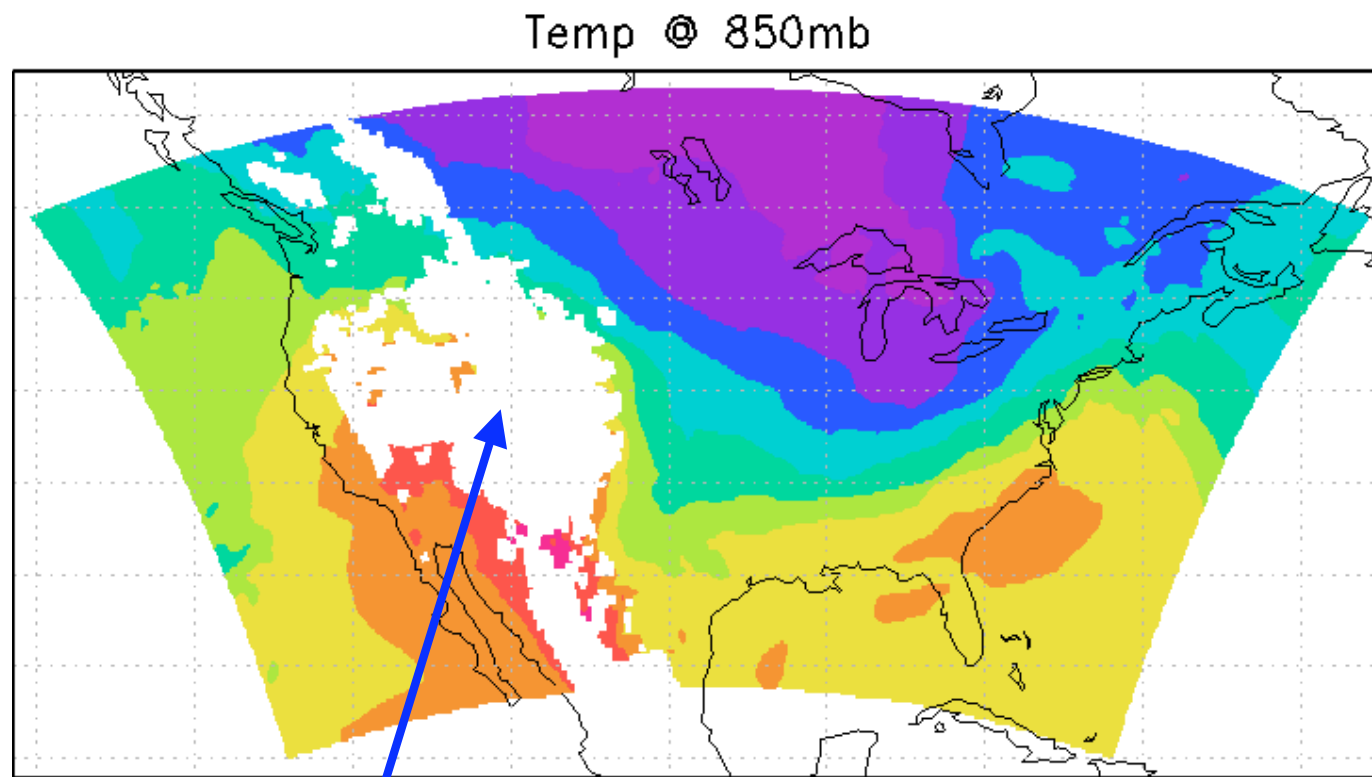
|                    |                                                                                                                                                                                                                                                                                      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

**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*)

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

# interpolation



**NO extrapolation below ground**

# run

---

- **./ARWpost.exe**

- **Will create either,**

***output\_root\_name.dat & output\_root\_name.ctl***

**OR**

***output\_root\_name.v5d***

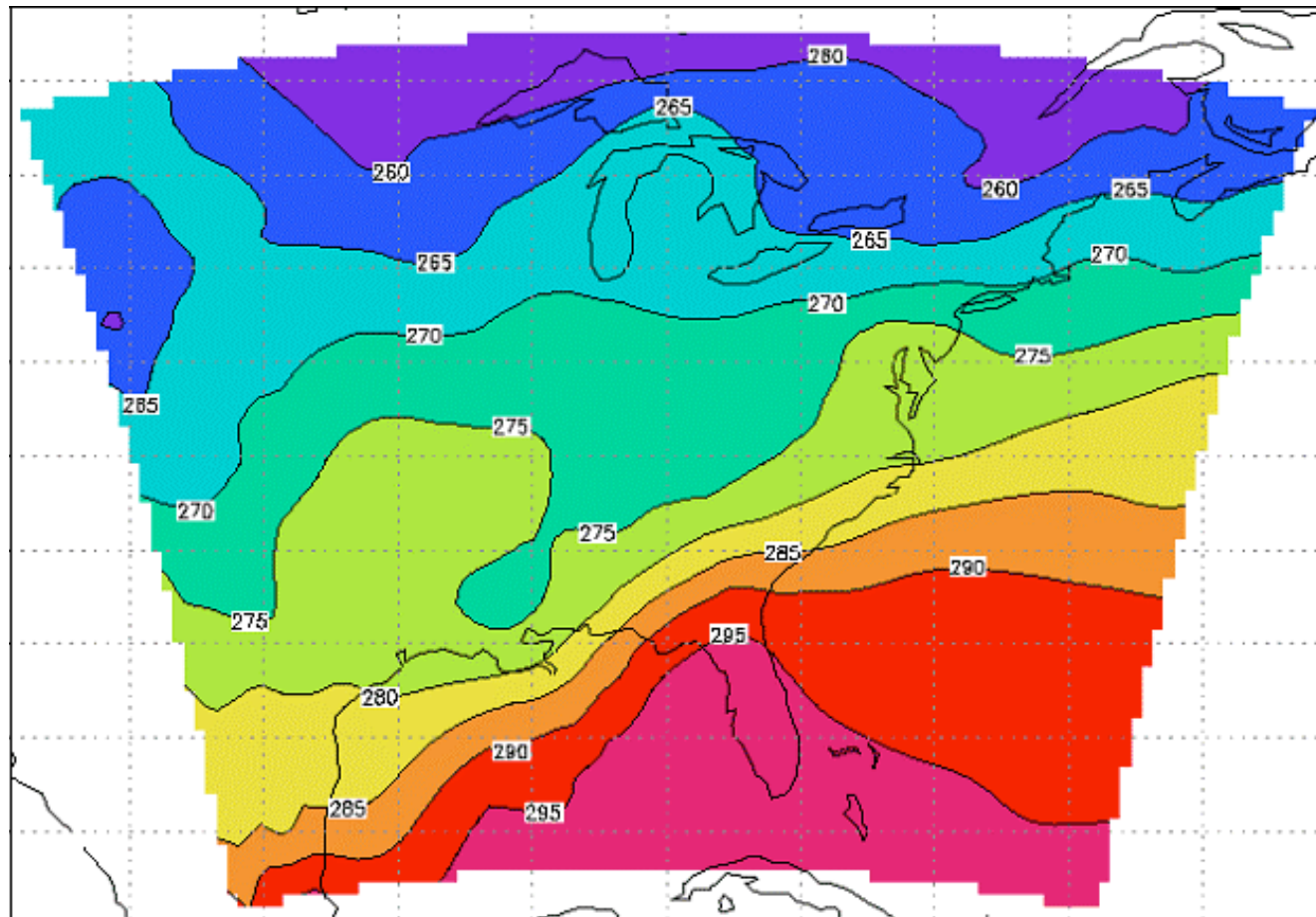


# GrADS specific notes

---

- **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

---

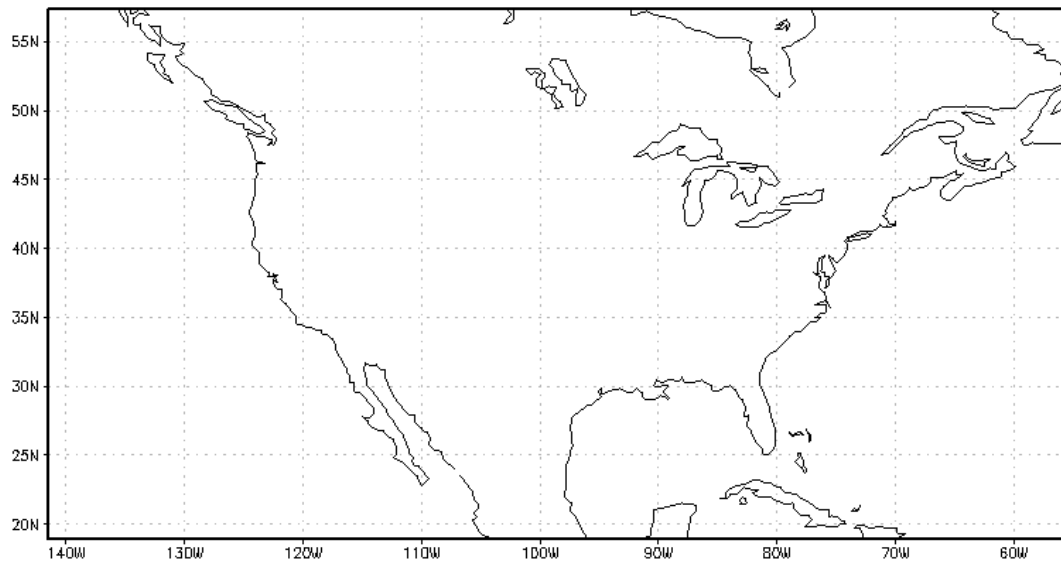
```
dset ^test.dat
options byteswapped
undef 1.e37
title OUTPUT FROM WRF V2.2 MODEL
pdef 259 163 lcc 40.000 -98.000 130.000 82.000
      60.00000 30.00000 -98.00000 22000.000 22000.000
xdef 877 linear -141.49254 0.09909910
ydef 389 linear 18.88639 0.09909910
```

## **options byteswapped**

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

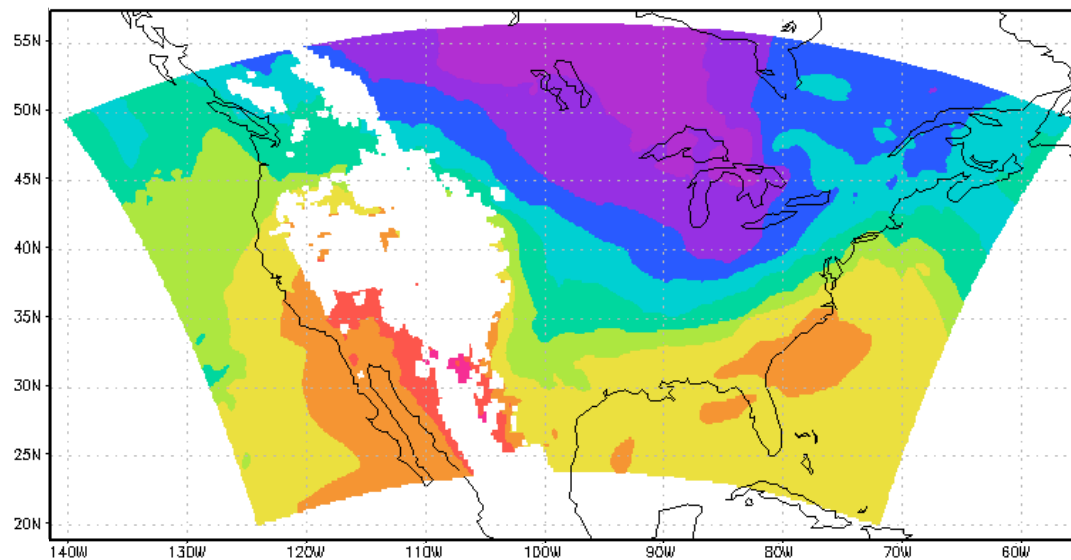
# GrADS - .ctl file

```
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
```





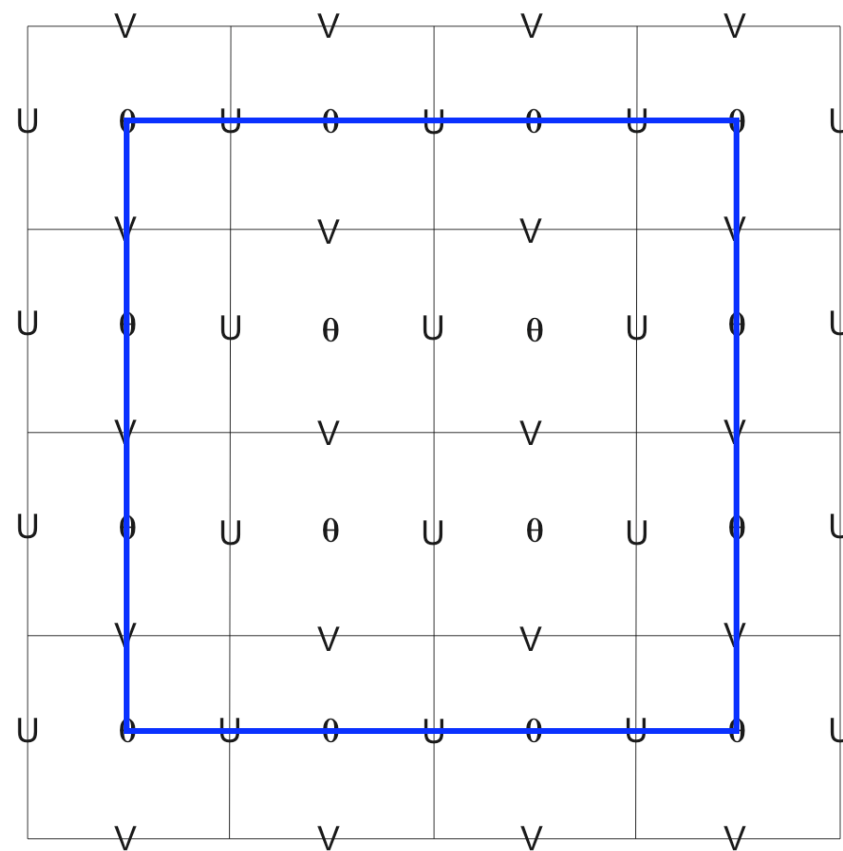
# GrADS conversion - question

---

- **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*

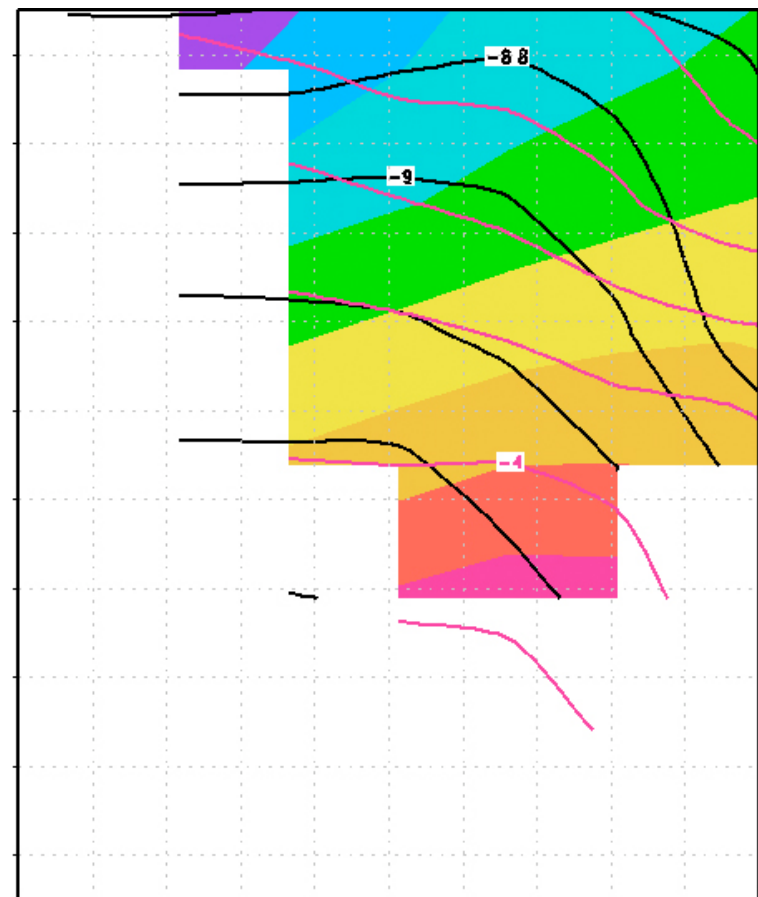
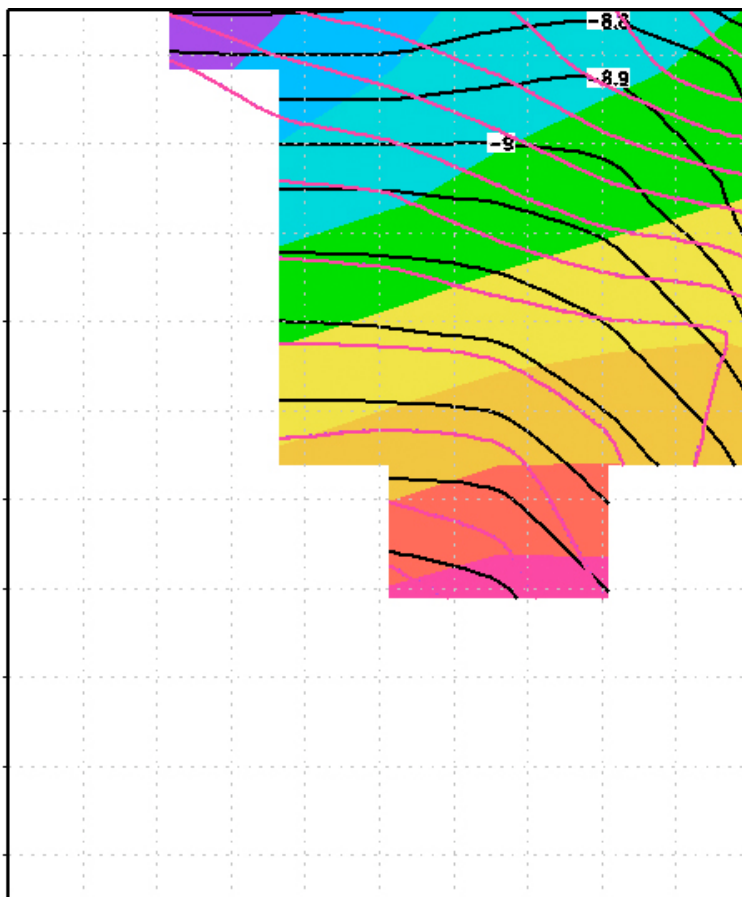
# GrADS conversion - question

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



# staggering

shaded=T ; black=U ; red=V



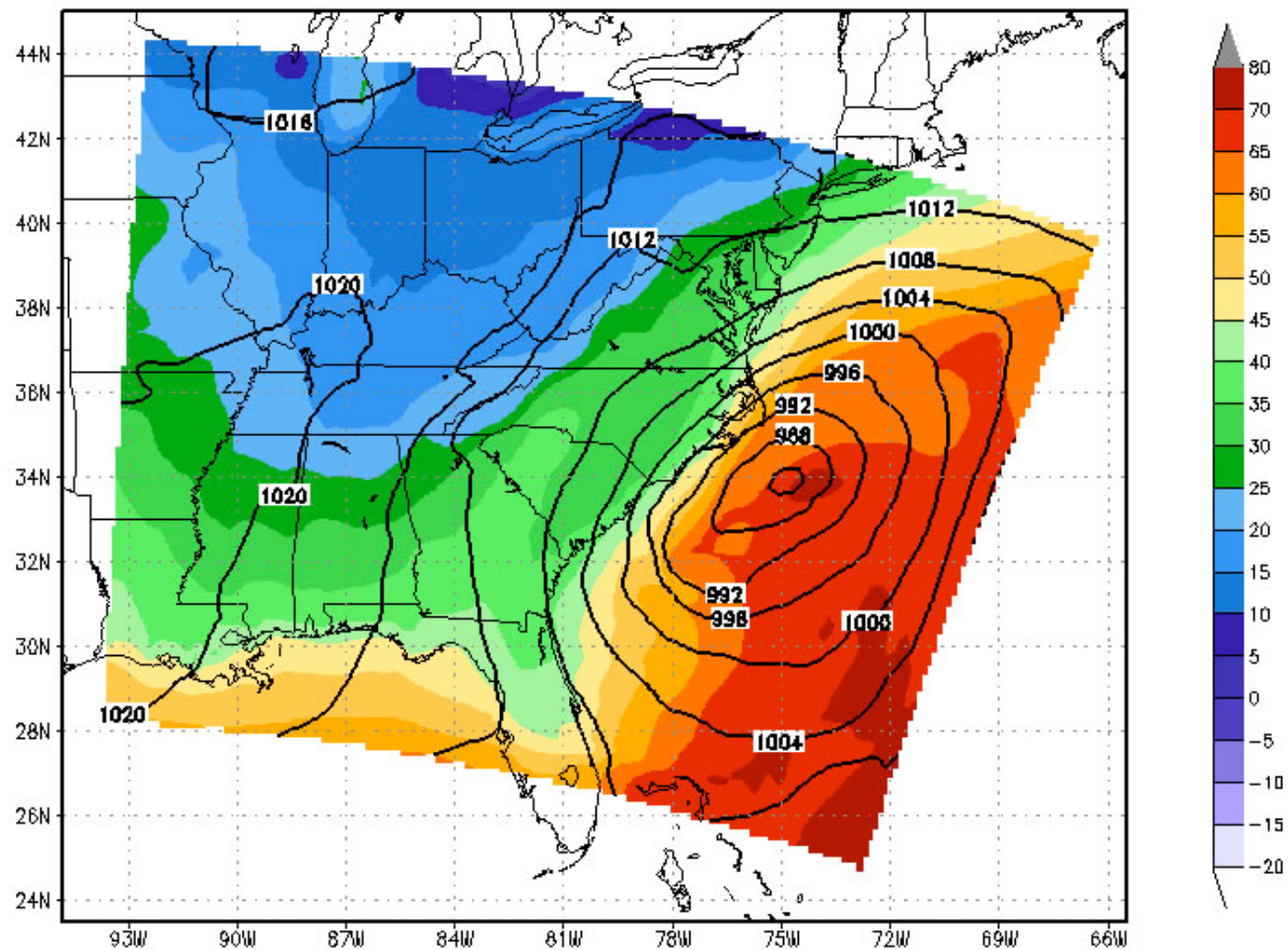
# staggering

---

- **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



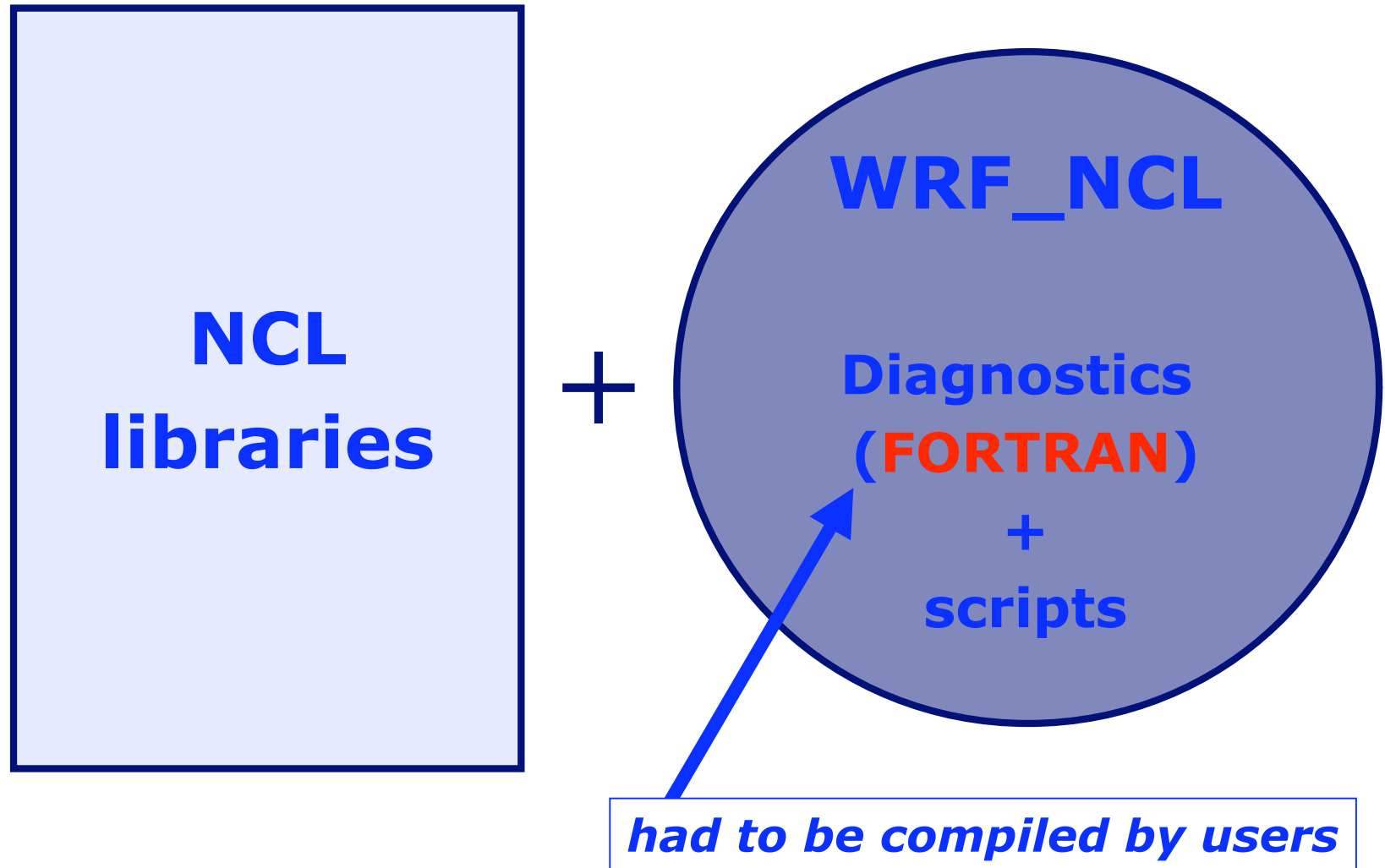
# NCL

---

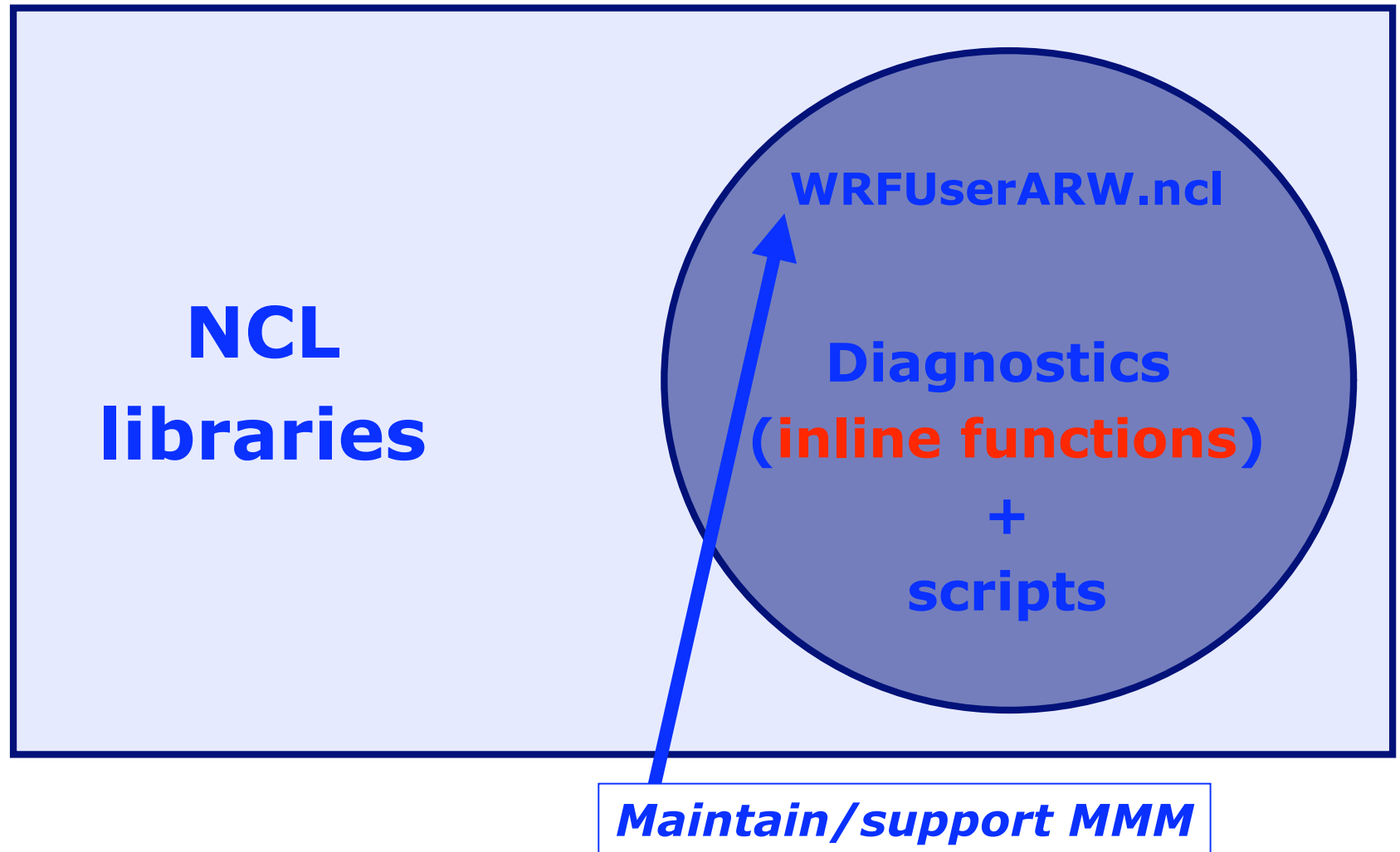
**NCL stands for NCAR Command Language**  
**<http://www.ncl.ucar.edu/>**

# 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/ncarg*
- ***Recommended to install NCAR Graphics first and then NCL on top of NCAR Graphics (most people don't***  
– *setenv NCARG\_ROOT /usr/local/ncl)*
- ***NCL version 4.3.1***

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>**

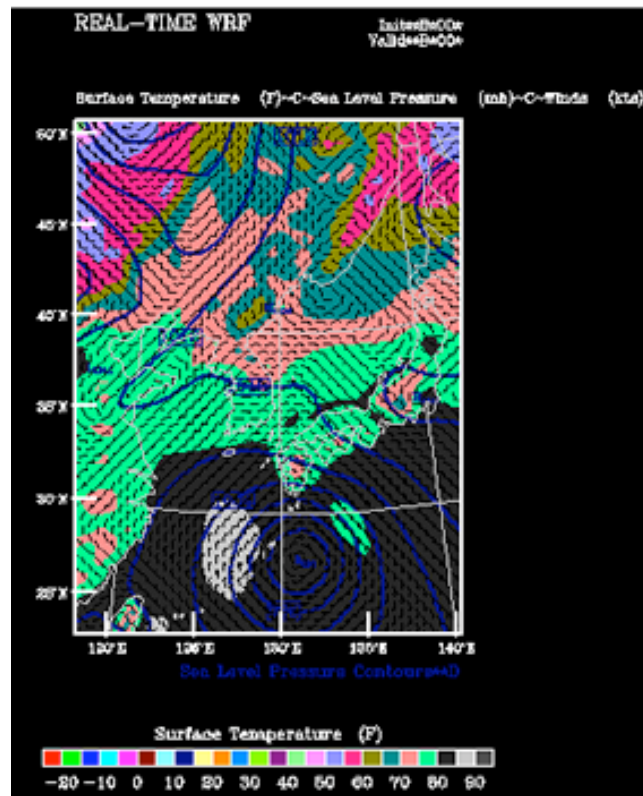
# ~/.hluresfile

---

|                           |                           |
|---------------------------|---------------------------|
| <b>*wkColorMap</b>        | <b>: WhViBlGrYeOrReWh</b> |
| <b>*wkBackgroundColor</b> | <b>: white</b>            |
| <b>*wkForegroundColor</b> | <b>: black</b>            |
| <b>*FuncCode</b>          | <b>: ~</b>                |
| <b>*TextFuncCode</b>      | <b>: ~</b>                |
| <b>*Font</b>              | <b>: helvetica</b>        |
| <b>*wkWidth</b>           | <b>: 900</b>              |
| <b>*wkHeight</b>          | <b>: 900</b>              |

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

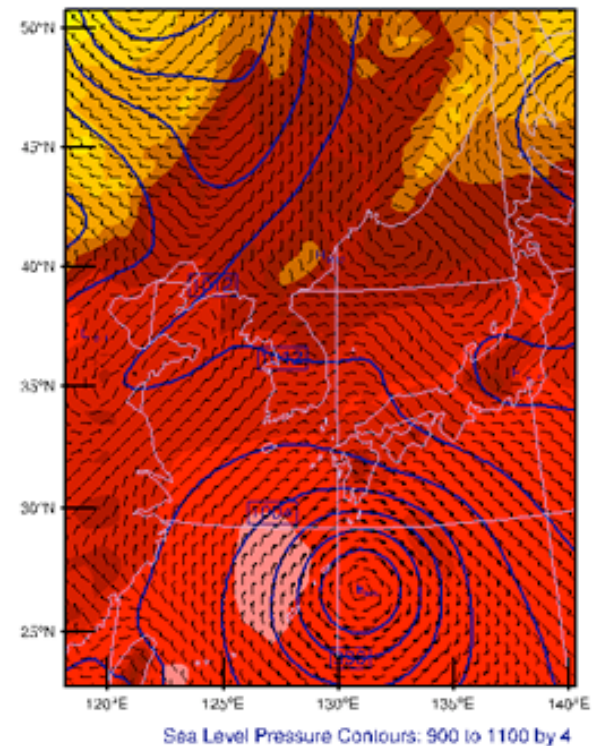
# ~/.hluresfile



## REAL-TIME WRF

Initial: 2002-08-29 00:00:00  
Valid: 2002-08-29 00:00:00

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



# Generate PLOTS

---

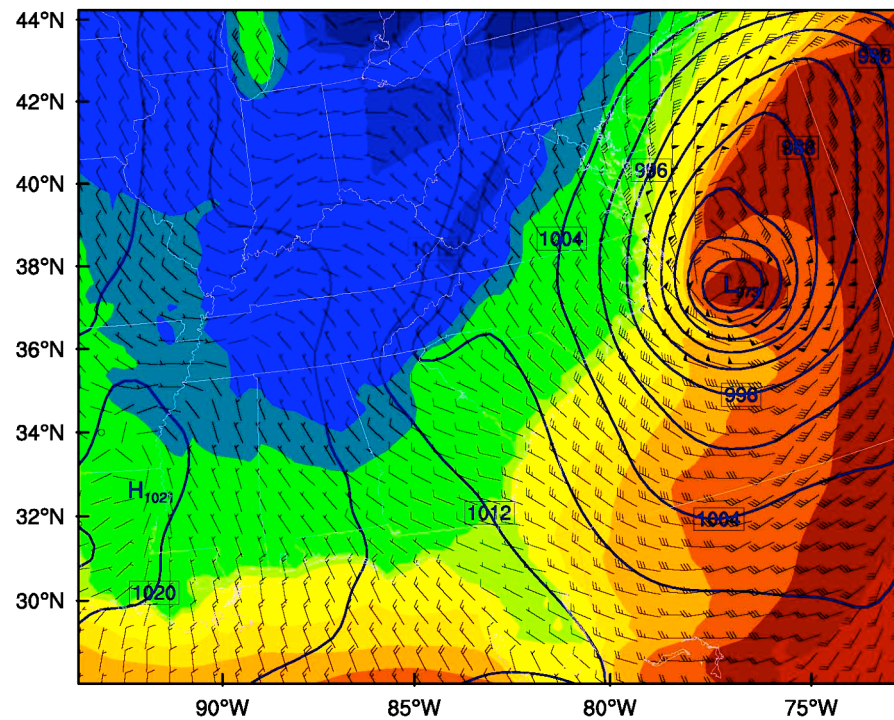
- **OnLine Tutorial:**

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

**Contain sample scripts**

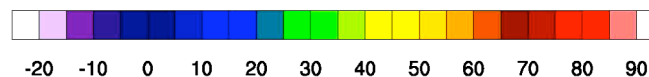
# creating a plot

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



Sea Level Pressure Contours: 900 to 1100 by 4

Surface Temperature (F)



# 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)**

**Not part of file name, but  
needs to be added inside  
the ncl script**

# creating a plot : NCL script

```
; Read variables
```

Function from: WRFUserARW.ncl  
Function will call NCL diagnostics  
if needed

```
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"
```

# 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/ncarg*
- *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
```