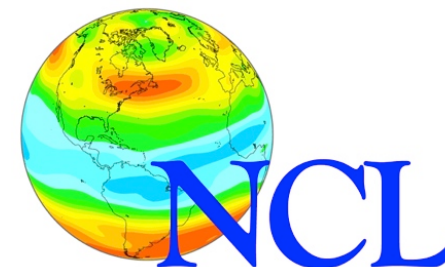




Analyzing and Visualizing WRF-ARW data using NCL and related Python tools



17th Annual WRF User's Workshop

Mary Haley & Bill Ladwig • CISL / TDD / DVAT

July 1, 2016



The National Center for Atmospheric Research is sponsored by the National Science Foundation

Our plan for the next 2 hours

8:30-9:20	NCL / WRF-NCL Intro (with demos) <i>Mary</i>
9:20-9:50	Python tools intro (with Jupyter Notebook demos) <i>Mary</i>
9:50-10:00	Break
10:00-10:30	WRF-Python Intro <i>Bill</i>

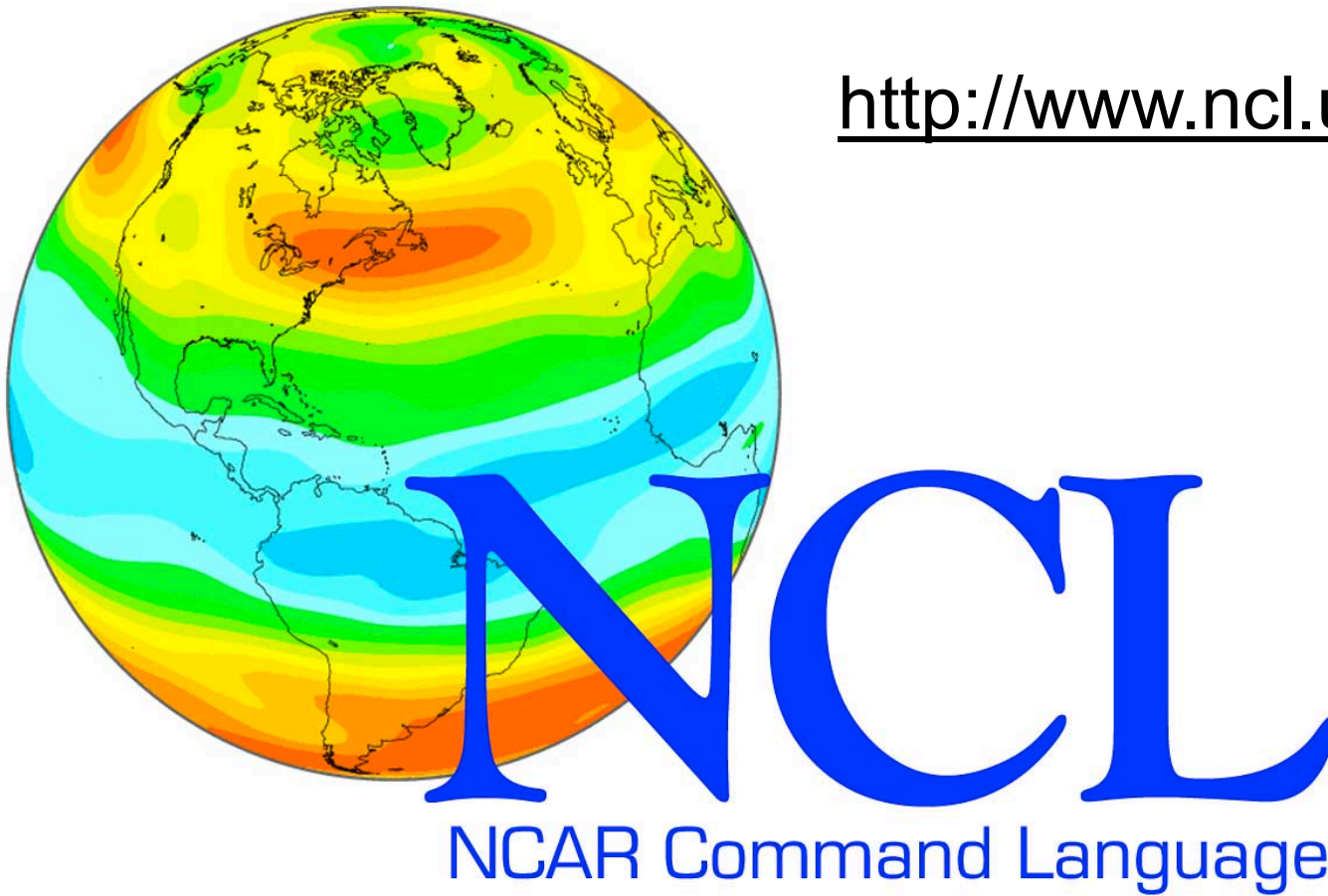
Bookmark this page!

http://www.ncl.ucar.edu/Training/Tutorials/WRF_Users_Workshop/

NCL Home Page -> Support -> Training -> Tutorials

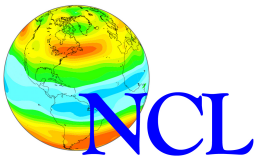
*A scripting language developed at NCAR
and tailored for the analysis and
visualization of geoscientific data*

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



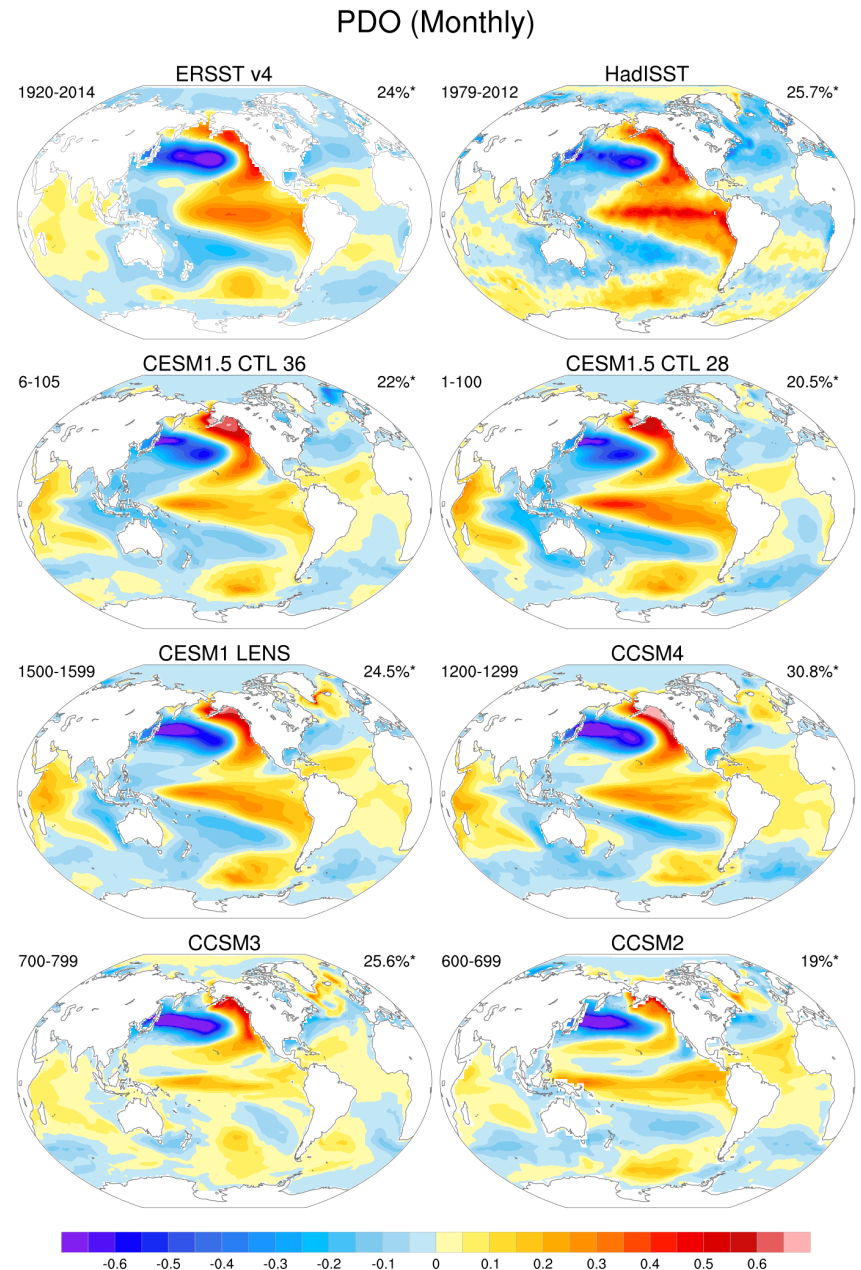
NCL Overview

- A scripting language similar to Matlab, Python, IDL
- Tailored to climate and weather sciences
- Has variable types, if-then-end if, do loops, arithmetic operators, functions, procedures
- Based on NetCDF data model
- F90-like array arithmetic
- Can call your own Fortran or C routines



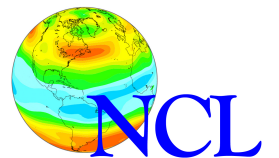
Why use NCL?

- Developed in NCAR/CISL in close collaboration with NCAR scientific staff
- Mature package
- Open source, free
- Extensive website, hundreds of examples
- Well-supported
- Intensive training workshops



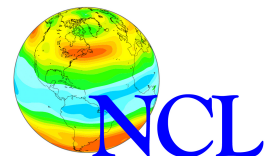
Three key features of NCL

1. Simple, but robust, file input/output
2. Hundreds of data analysis routines
3. Publication-quality graphics are highly customizable



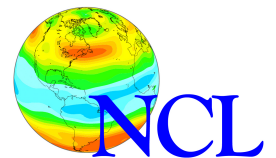
1. File input and output

- Data model based on netCDF model (metadata describes data)
- **One function** reads all supported data formats:
 - NetCDF3, GRIB 1 and 2, HDF4, HDF5, HDF-EOS2, HDF-EOS5, shapefiles, NetCDF4 (groups, compound data, variable length arrays)
 - Writes NetCDF3, NetCDF4, and HDF4
- OPeNDAP-enabled client available
- ASCII, Fortran/C binary (read and write)
- “Never fear a data format”

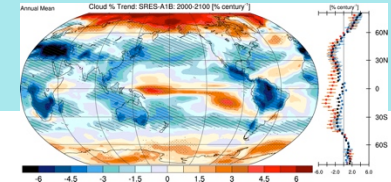
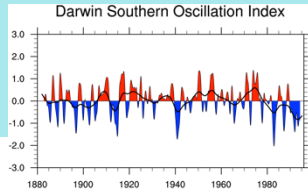


2. Data analysis

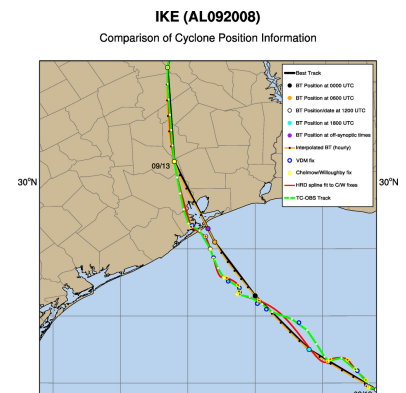
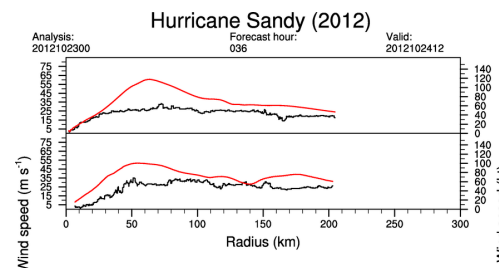
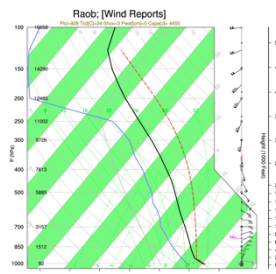
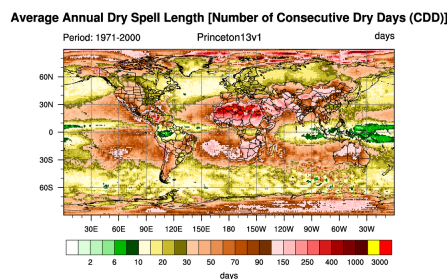
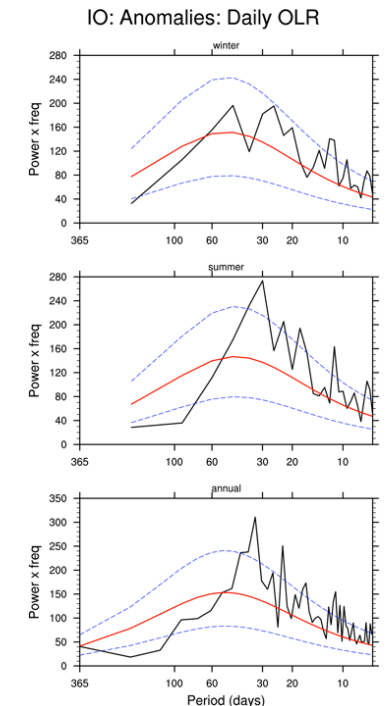
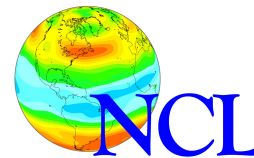
- Array-based math
- Hundreds of functions
 - WRF-ARW specific functions
 - Climatologies
 - Spherical harmonics
 - Interpolation and regridding
 - Crop / Heat stress – new in next release
 - EOFs
- Most automatically handle missing data



3. Visualization



- Publication quality and customizable 2D visualizations
- Contours, XY, vectors, wind barbs, streamlines
- Maps with common map projections
- Handles data on rectilinear, curvilinear, and unstructured grids (MPAS, triangular meshes)
- Specialized scripts for meteograms, skew-T, wind roses, histograms, cross section, panels
- Suite of wrf_xxxx functions: simplifies visualization for WRF-ARW data
- Over 1,400 visualization “options”



Demo: examining file on UNIX command line

ncl_filedump

Similar to “ncdump” but works on NetCDF, GRIB1, GRIB2, HDF4, HDF5, HDF-EOS2, HDF-EOS5, shapefile

#1 rule of data analysis:

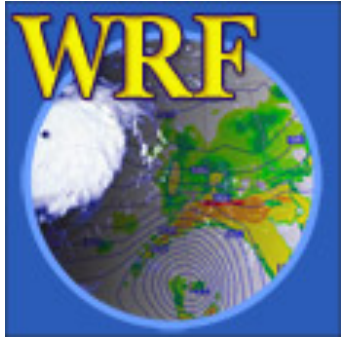
“Look at your data”

Demo: examining file in an NCL script

wrf_demo_examine_file.ncl

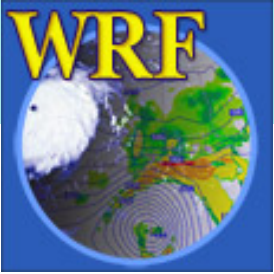
Functions used:

addfile
getvarnames / getvaratts
printVarSummary / printMinMax
typeof / dimsizes / isatt



WRF-NCL

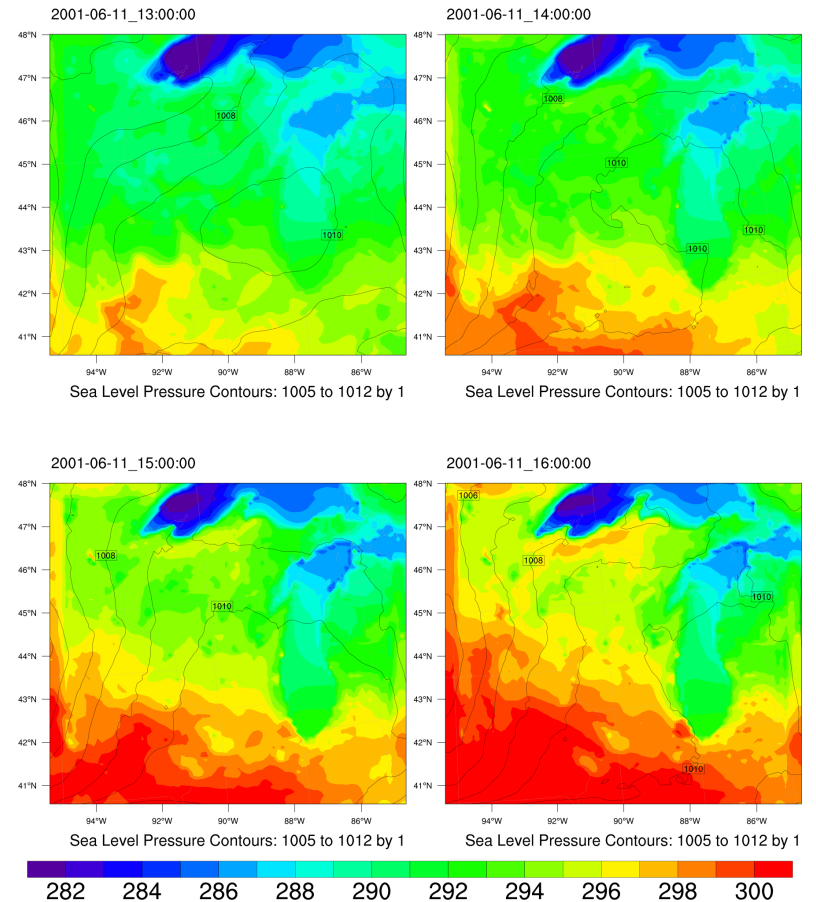
*NCL suite of analysis and
visualization functions tailored
for WRF-ARW model data*



WRF-NCL Overview

- Included with NCL since 2006
- Developed by staff in NCAR/MMM
- Maintained by MMM and CISL
- Functions for calculating basic diagnostics (**wrf_user_getvar**)
- Functions for highly customized visualizations
- Website with lots of analysis and visualization examples

TEMP at 2 M (K)



Full list of WRF-NCL functions

<http://www.ncl.ucar.edu/Document/Functions/wrf.shtml>

NCL functions to be called by user

wrf_user_getvar

wrf_user_ij_to_ll
wrf_user_intrp2d
wrf_user_intrp3d
wrf_user_list_times
wrf_user_ll_to_ij
wrf_user_unstagger
wrf_user_vert_interp

Graphics

wrf_contour

wrf_vector

wrf_overlays
wrf_map
wrf_map_overlays
wrf_map_resources
wrf_map_zoom

WPS intermediate files

wrf_wps_open_int
wrf_wps_rddata_int
wrf_wps_rdhead_int
wrf_wps_read_int
wrf_wps_write_int
wrf_wps_close_int

Computational routines not generally called directly by user

wrf_avo	wrf_pvo	wrf_smooth_2d	wrf_interp_1d
wrf_cape_2d	wrf_rh	wrf_helicity	wrf_interp_2d_xy
wrf_cape_3d	wrf_slp	wrf_updraft_helicity	wrf_interp_3d_z
wrf_dbz	wrf_uvmet	wrf_virtual_temp	wrf_ij_to_ll
wrf_eth	wrf_td	wrf_wetbulb	wrf_ll_to_ij
wrf_omega	wrf_tk		

Main WRF-NCL function: **wrf_user_getvar**

wrf_user_getvar - Get fields from input file and/or calculate diagnostics

```
a = addfile("wrfout_d01_2005-08-28_00:00:00","r")
cttmp = wrf_user_getvar(a,"ctt",0) ; 0 → first time step
t2     = wrf_user_getvar(a,"T2",-1) ; -1 → all time steps
slp    = wrf_user_getvar(a,"slp",1)
```

Sample diagnostics

<i>avo/pvo</i>	Absolute/Potential Vorticity
<i>cape_2d</i>	2D mcape/mcin/lcl/lfc
<i>cape_3d</i>	3D cape/cin
<i>ctt</i>	Cloud Top Temperature [K]
<i>dbz/mdbz</i>	Reflectivity
<i>slp</i>	Sea level pressure
<i>tv</i>	Virtual temperature [K]
<i>tw</i>	Wet bulb temperature [C]
<i>updraft_helicity</i>	Updraft helicity [m-2/s-2]
<i>ua/va/wa</i>	Wind on mass points
<i>uvmet/uvmet10</i>	U/V components of wind rotated to earth coords

Demo: using wrf_user_getvar

Using `wrf_user_getvar` to read data
and / or calculate diagnostics

```
wrf_demo_getvar_simple.ncl
```

```
wrf_demo_getvar_detailed.ncl
```

Sample code

```
f = addfile("wrfout_d01_2008-09-29_16:30:00","r")

;--- Read HGT variable
hgt = wrf_user_getvar(f,"HGT",0)      ; read first time step

printMinMax(hgt,0)

;--- Calculate sea level pressure at all time steps
slp = wrf_user_getvar(f,"slp",-1) ; [Time | 1] x [south_north | 197] x [west_east | 206]

;--- Calculate u and v on mass points; put on same grid
u = wrf_user_getvar(f,"ua",0)
v = wrf_user_getvar(f,"va",0)

printVarSummary(u) ; [bottom_top | 34] x [south_north | 160] x [west_east | 180]
printVarSummary(v) ; the same size as "u"
```

WRF-NCL graphical functions

wrf_contour

wrf_vector

wrf_overlays

wrf_map_overlays

wrf_map

wrf_map_resources

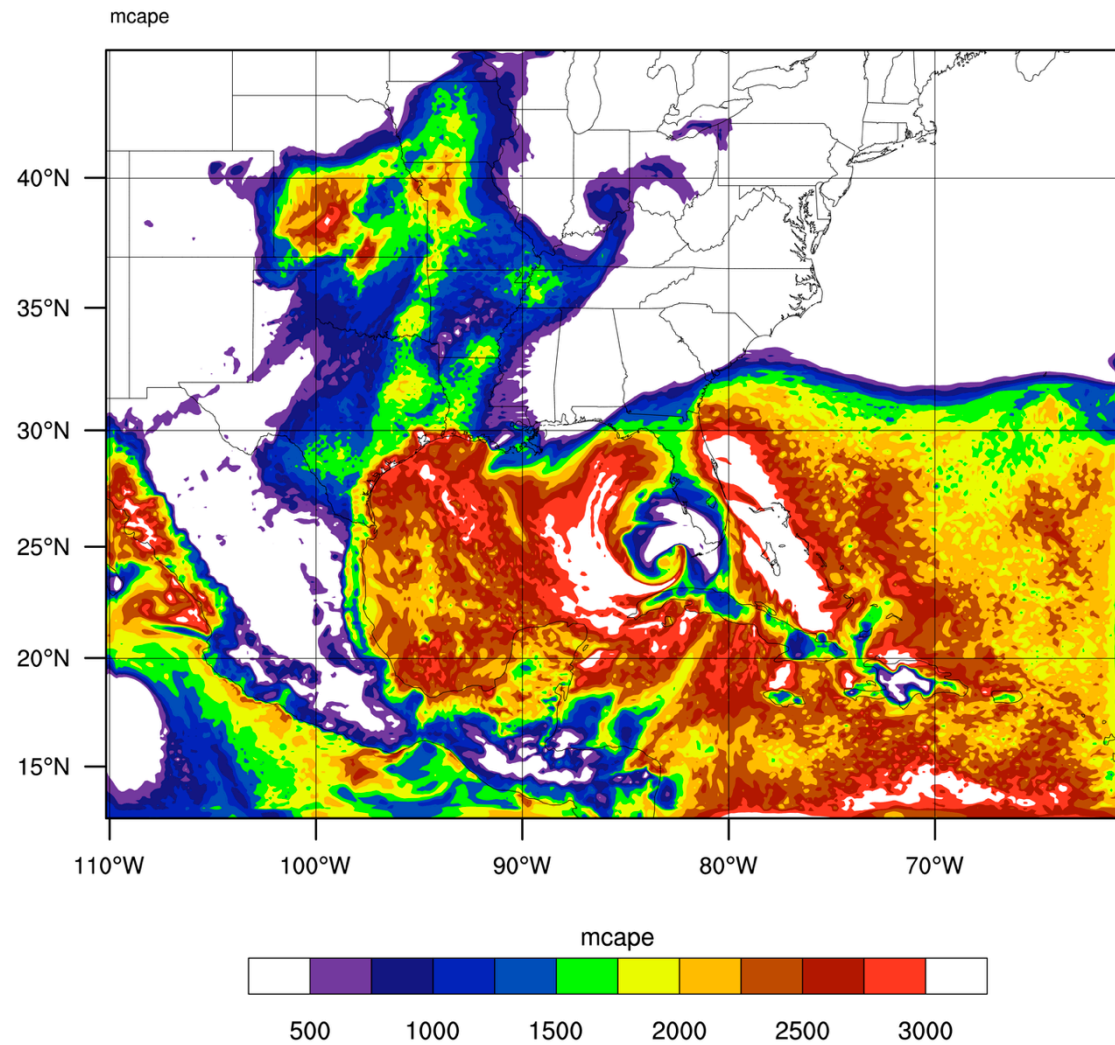
wrf_map_zoom

Sample WRF-NCL visualizations

REAL-TIME WRF

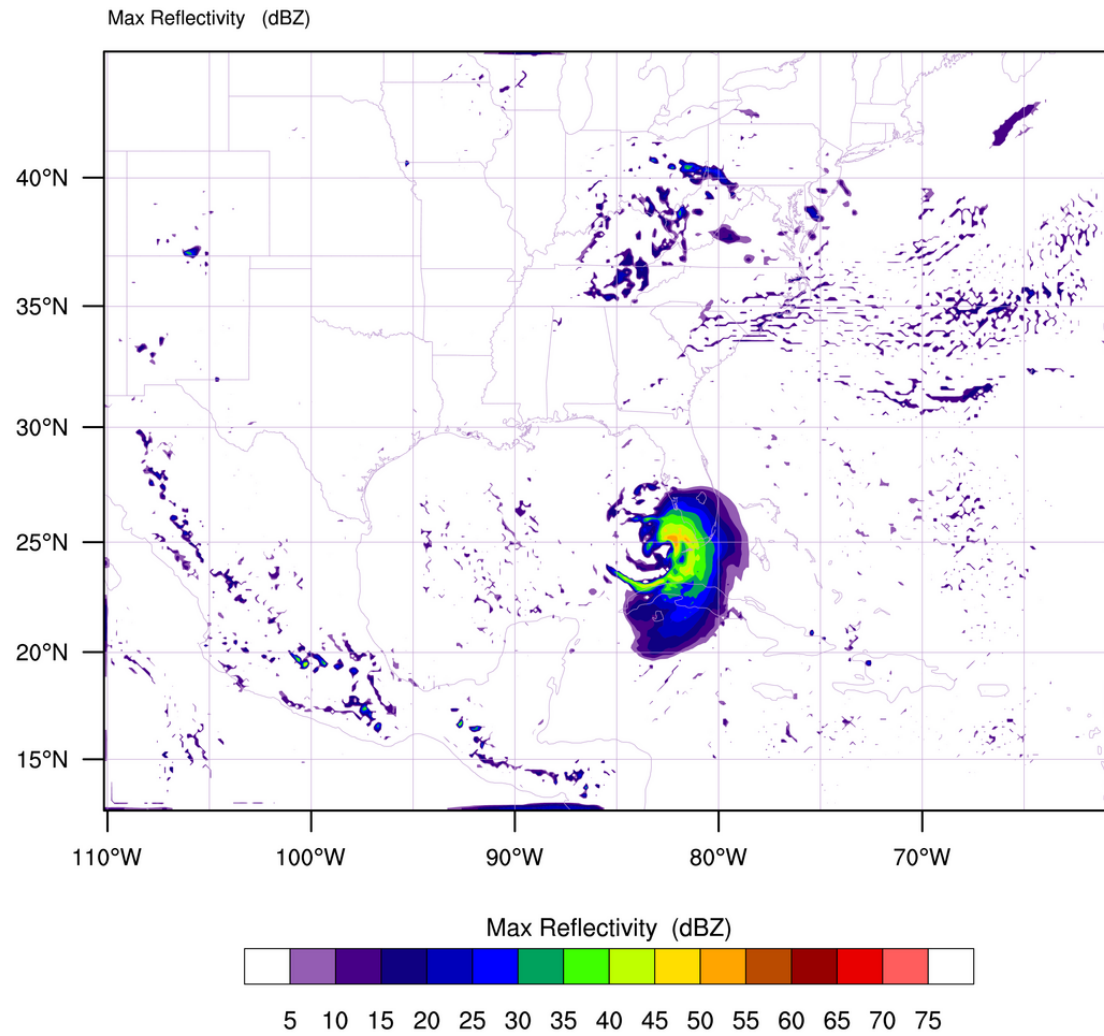
Init: 2005-08-26_00:00:00

Valid: 2005-08-27_00:00:00



REAL-TIME WRF

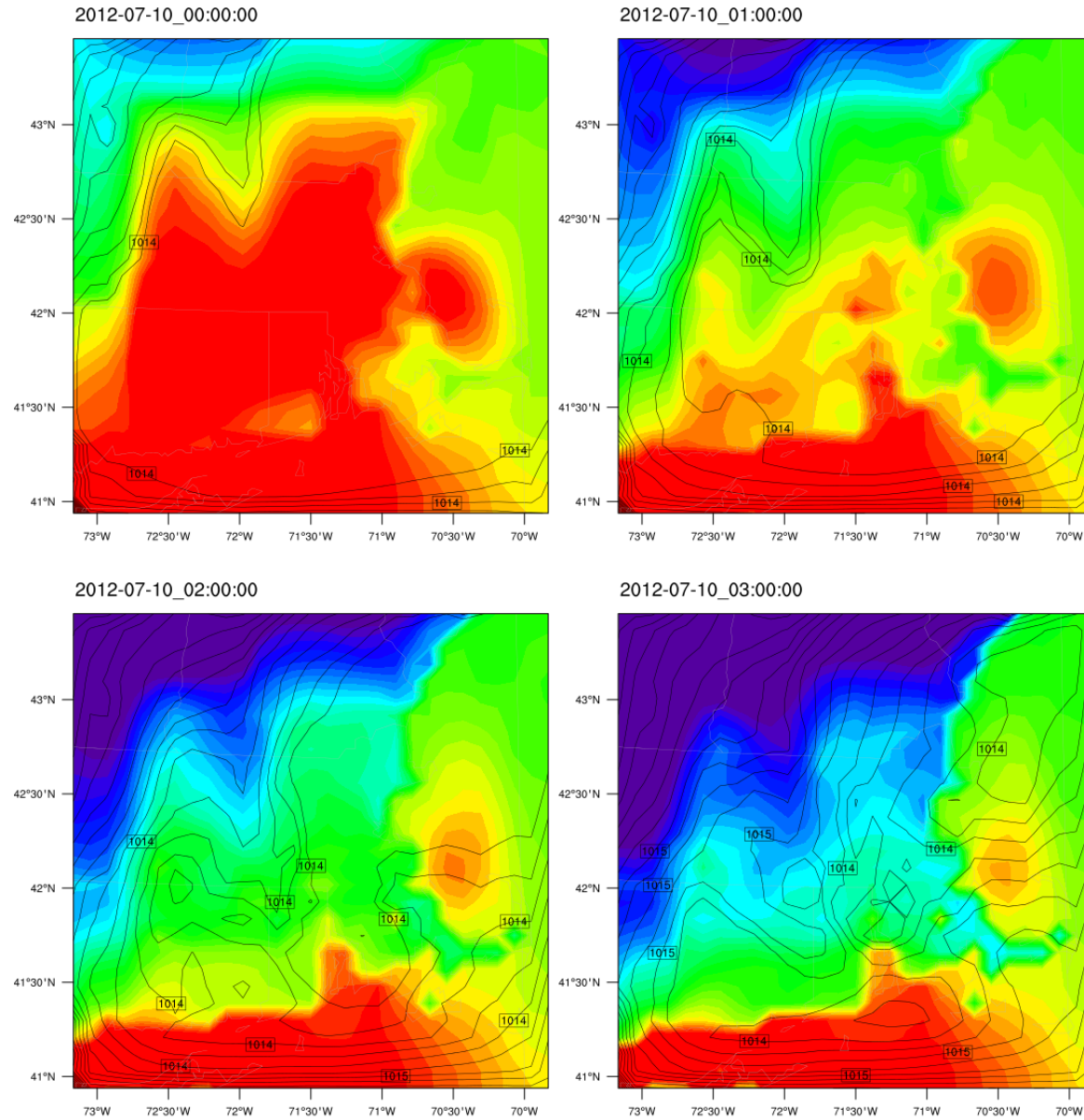
Init: 2005-08-26_00:00:00
Valid: 2005-08-27_00:00:00



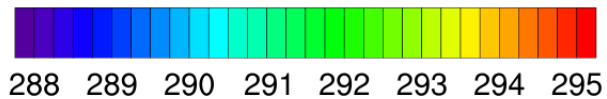
OUTPUT FROM WRF V2.1.2 MODEL
WE = 400 ; SN = 301 ; Levels = 35 ; Dis = 12km ; Phys Opt = 3 ; PBL Opt = 1 ; Cu Opt = 1

TEMP at 2 M (K)

These are
called
“panel” plots

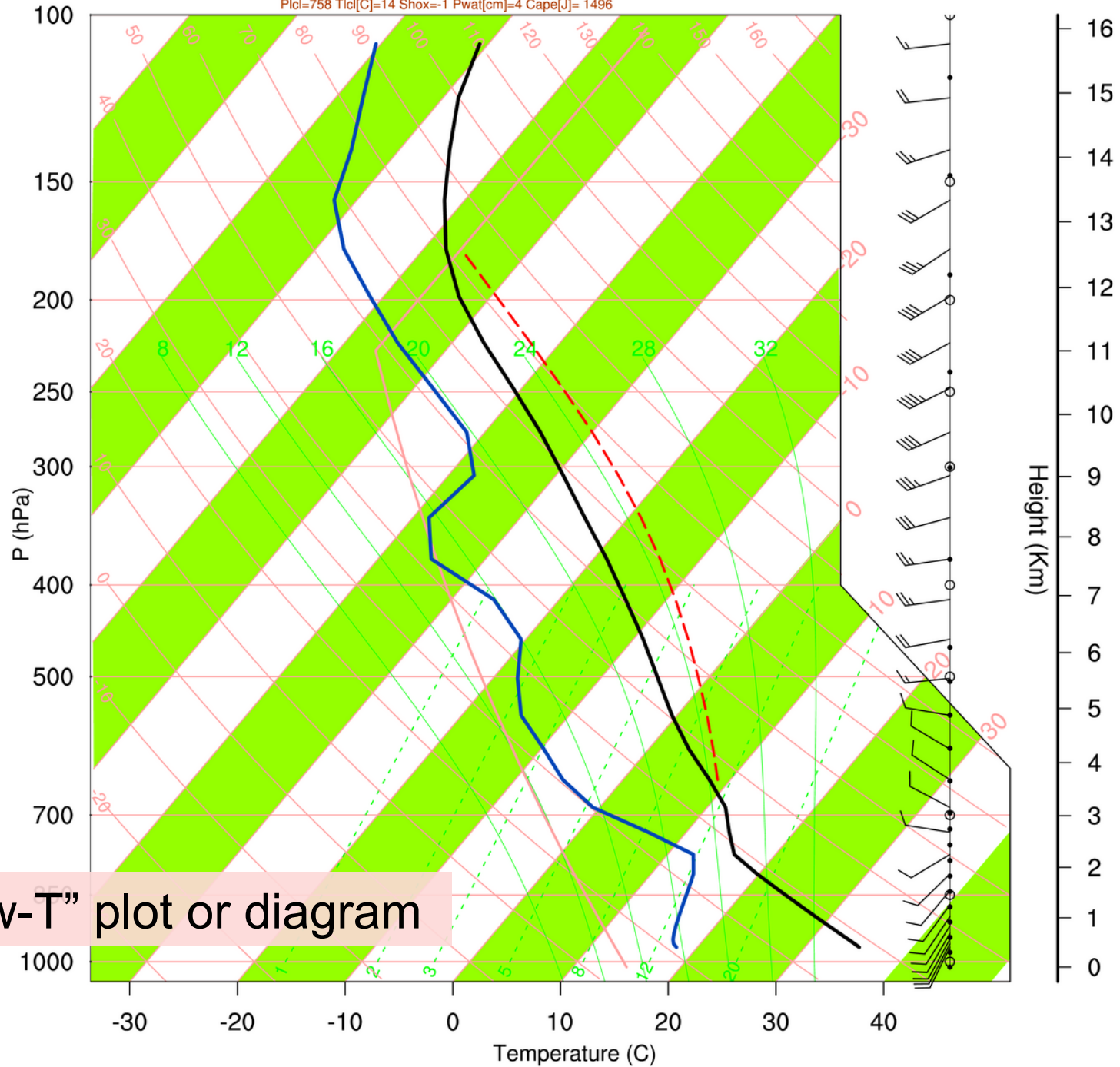


Sea Level Pressure Contours: 1014 to 1016 by .1



Norman (OK,USA) at 2005-08-27_00:00:00

Ptcl=758 Tlcl[C]=14 Shox=-1 Pwat[cm]=4 Cape[J]= 1496



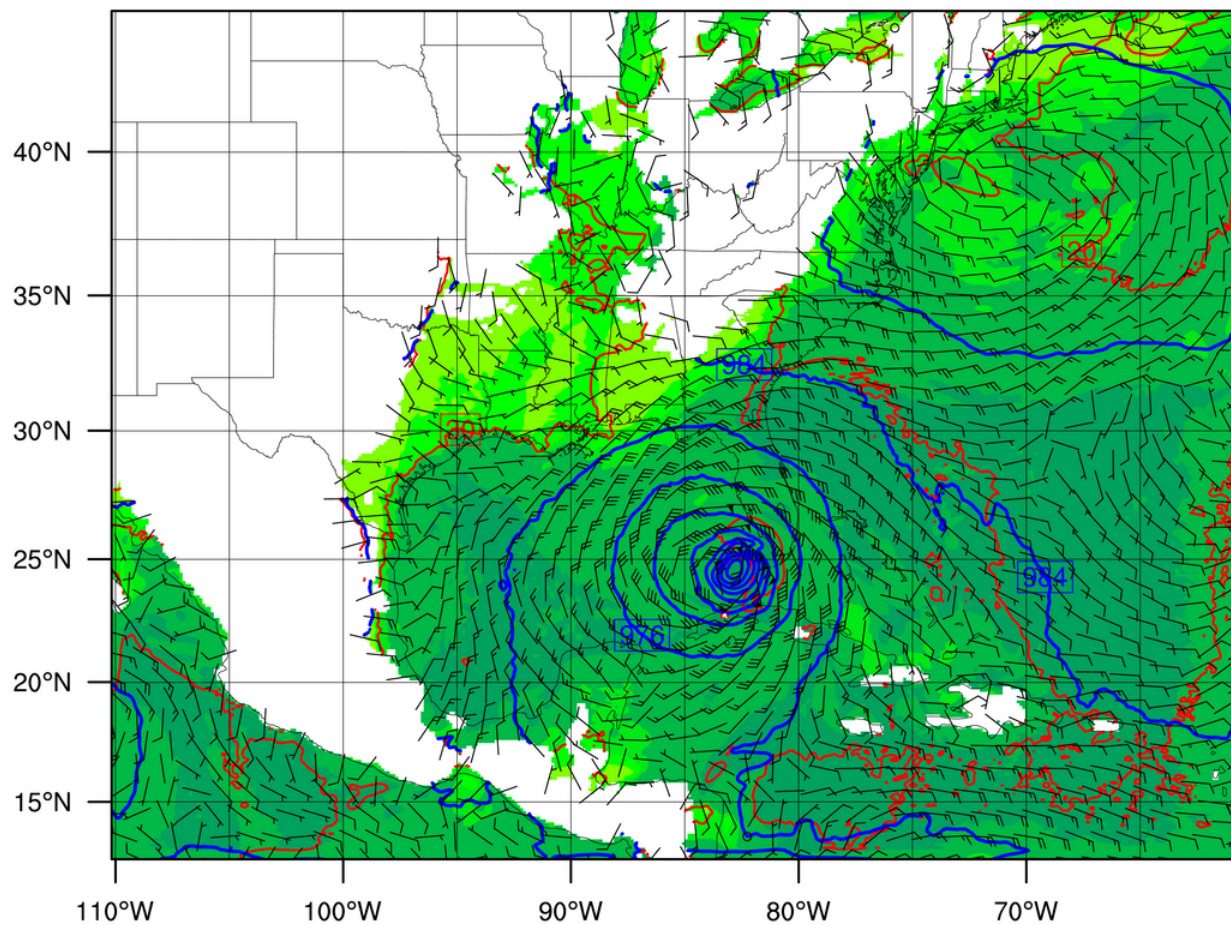
“Skew-T” plot or diagram

REAL-TIME WRF

Init: 2005-08-26_00:00:00

Valid: 2005-08-27_00:00:00

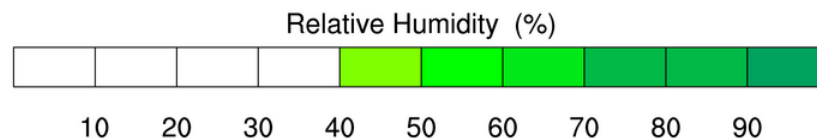
Relative Humidity (%) at 0.25 km
Temperature (C) at 0.25 km
Pressure (hPa) at 0.25 km
Wind (kts) at 0.25 km

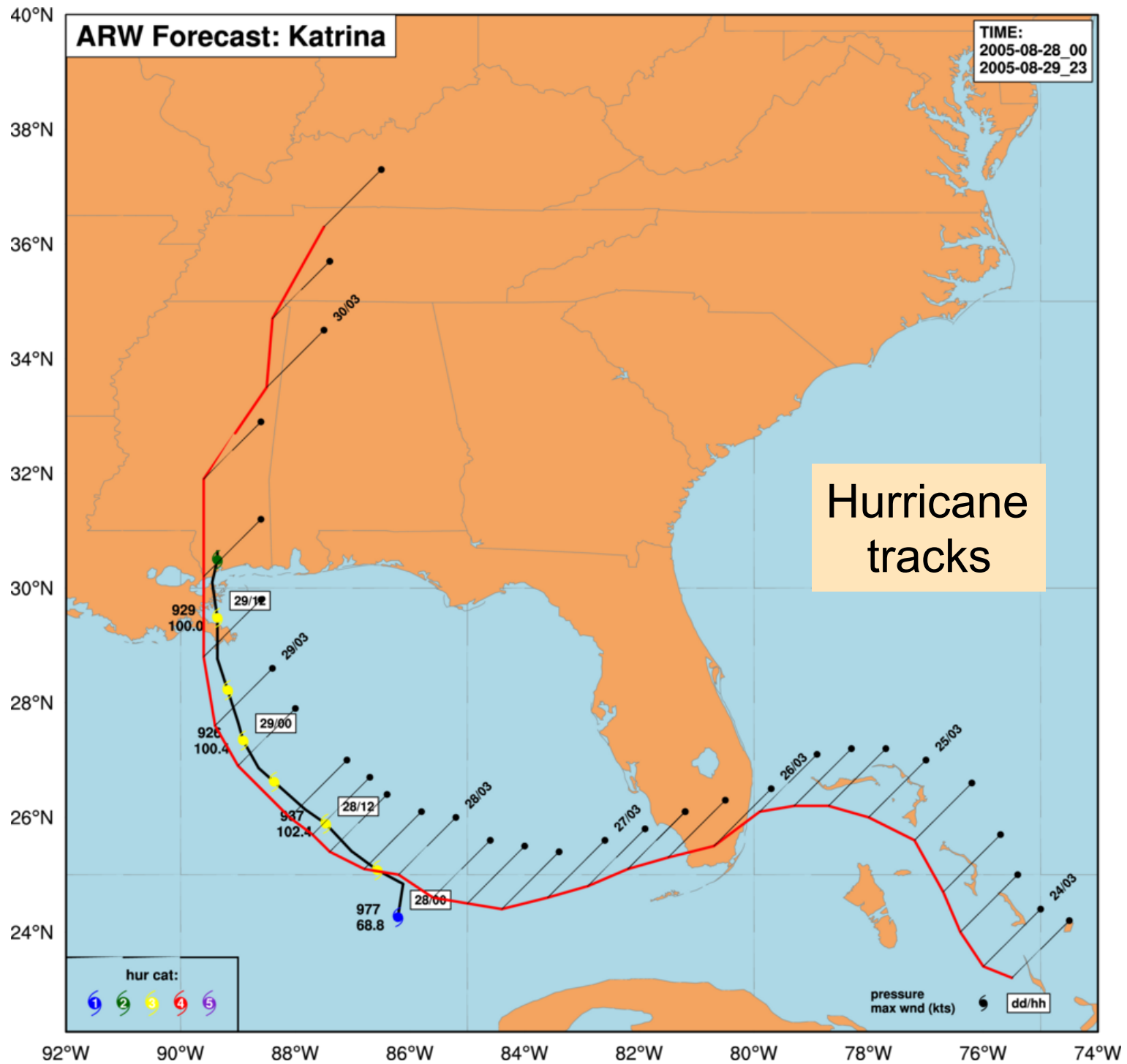


Pressure Contours: 948 to 988 by 4

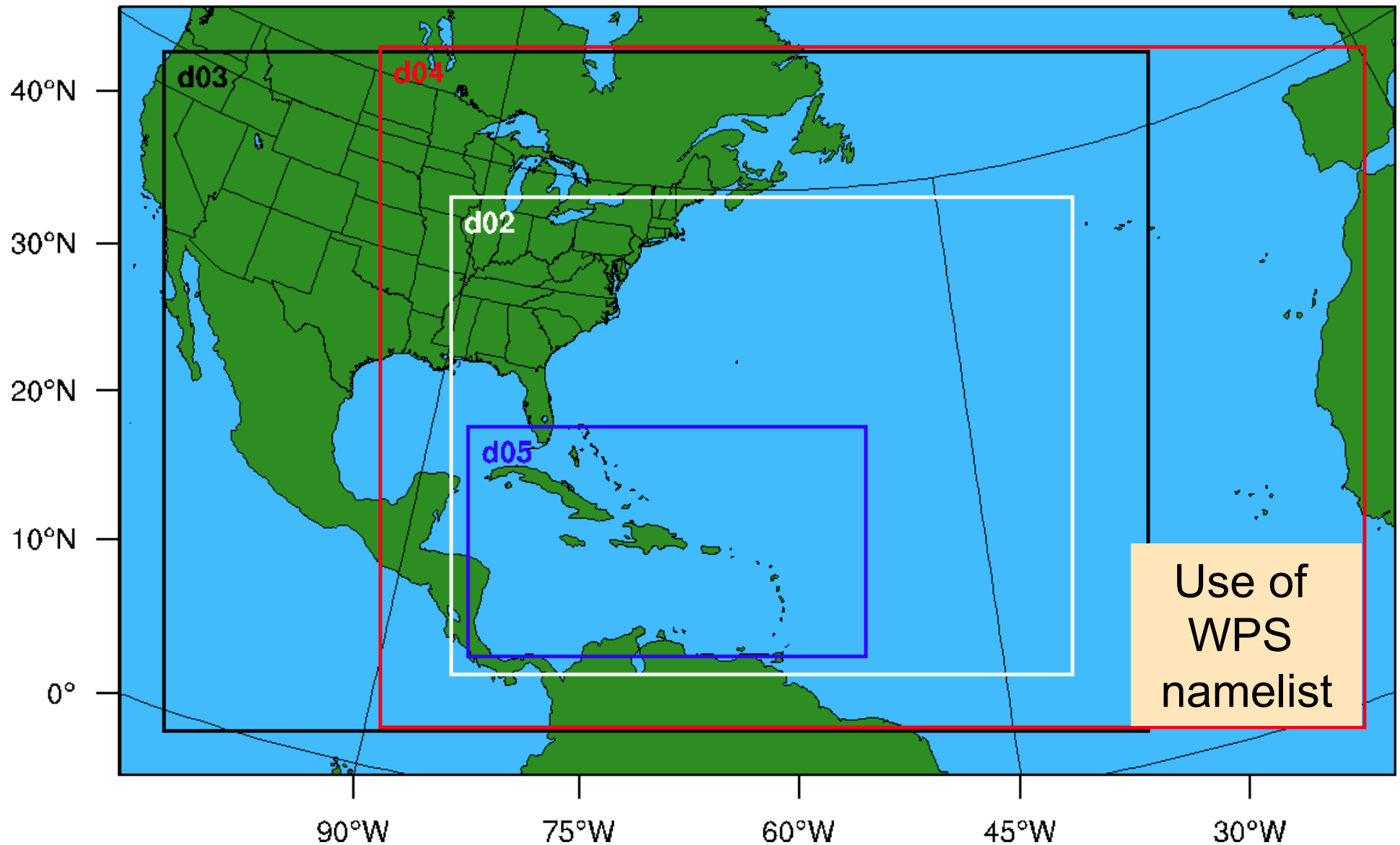
Temperature Contours: 10 to 45 by 5

Wind barbs and contours

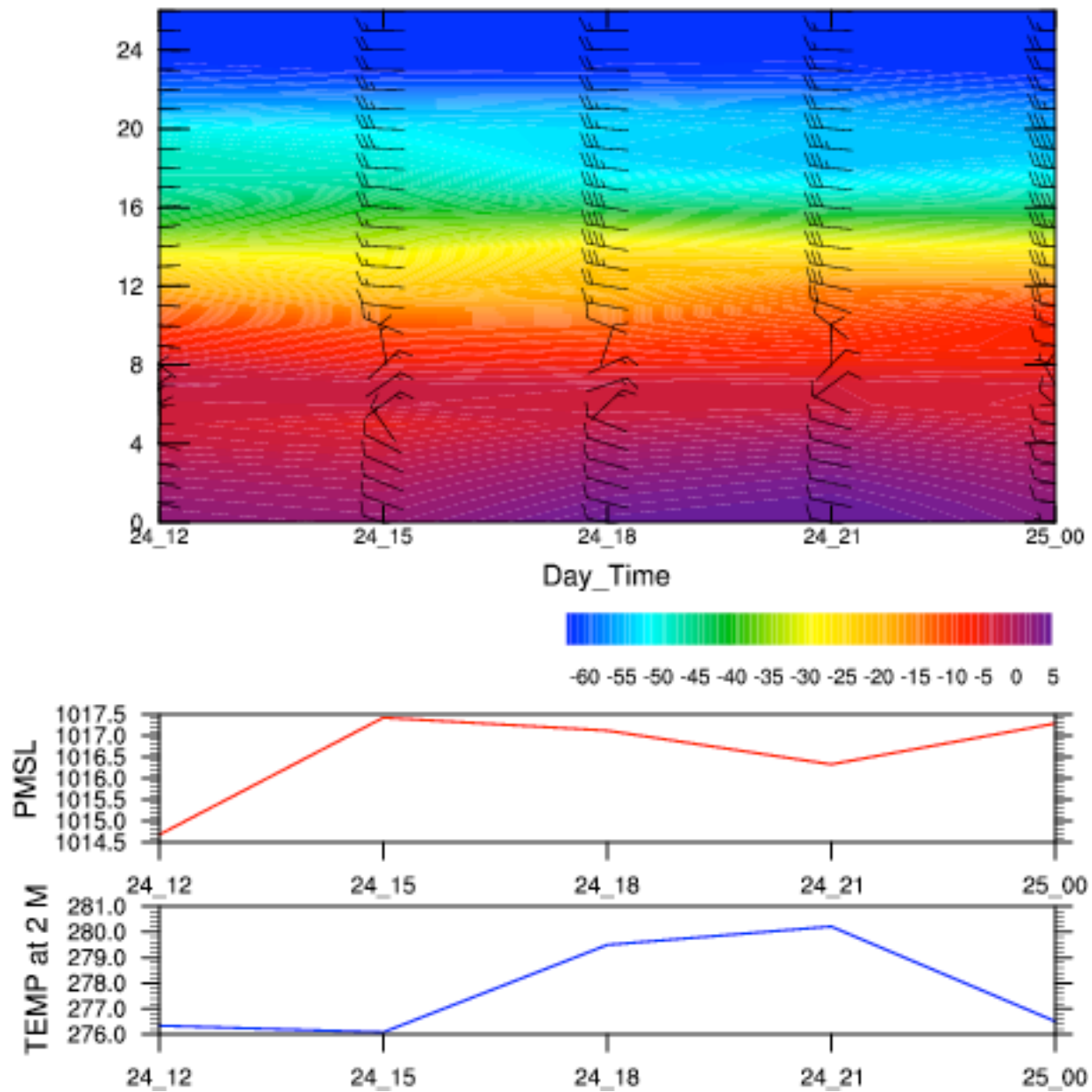




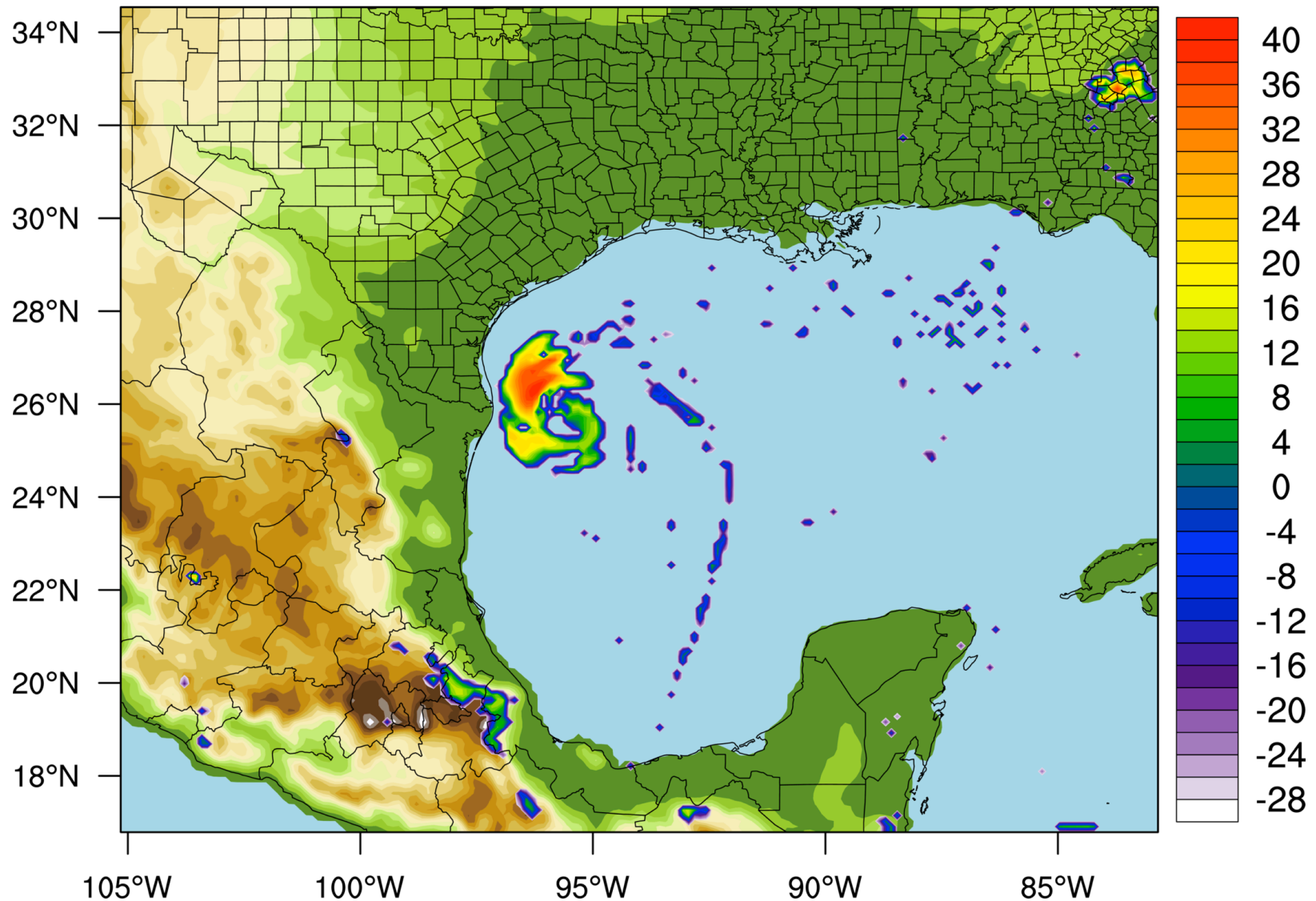
WPS Domain Configuration



Meteogram for lat=32.5 ; lon=-87



Reflectivity (dBZ) at level = 0.996



WRF “DBZ” and “HGT” data
Uses multiple color maps and transparency

Demo: plotting “HGT” variable

Using WRF-NCL functions to create plots

```
wrf_demo_plot_hgt.ncl
```

```
wrf_demo_plot_hgt_custom.ncl
```


wrf_demo_plot_hgt.ncl

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

```
;--- Open file and read a variable  
f = addfile("wrfout_d01_2005-08-28_00:00:00","r")  
hgt = wrf_user_getvar(f,"HGT",0)
```

```
printVarSummary(hgt)    ; LOOK AT YOUR DATA VARIABLE!
```

```
;--- Where to send the graphics  
wks = gsn_open_wks("x11","wrf_hgt")    ; "png", "pdf", "svg"
```

```
;--- Set one plotting resource  
res = True  
res@cnFillOn = True        ; Turn on color fill
```

```
;--- Create a contour plot  
contour = wrf_contour(f,wks,hgt,res)
```

```
;--- Draw the contours over a map  
plot = wrf_map_overlays(f,wks,contour,False,False)
```

Demo: overlaying multiple plots

Creating multiple overlays

```
wrf_demo_plot_overlays.ncl
```

wrf_demo_plot_overlays.ncl

```
f = addfile("wrfout_d01_2005-08-28_00:00:00","r")

;--- Open a file and get several diagnostics
slp = wrf_user_getvar(f, "slp", 0)
t2  = wrf_user_getvar(f, "T2", 0)
u10 = wrf_user_getvar(f, "U10", 0)
v10 = wrf_user_getvar(f, "V10", 0)

wks = gsn_open_wks("x11","wrf_overlays")

;--- Line contours
os      = True
os@cnLineColor      = "NavyBlue"
os@cnLineThicknessF = 2.0
c_slp      = wrf_contour(f,wks,slp,os)

;--- Filled contours
ot      = True
ot@cnFillOn      = True
c_tc      = wrf_contour(f,wks,t2,ot)

;--- Vectors
ov      = True
ov@NumVectors      = 47
vec      = wrf_vector(f,wks,u10,v10,ov)

;--- Overlay vectors, line contours, and filled contours on a map
plot = wrf_map_overlays(f,wks,(/c_tc,c_slp,vec/),False,False)
```

wrf_contour/wrf_vector - special resources

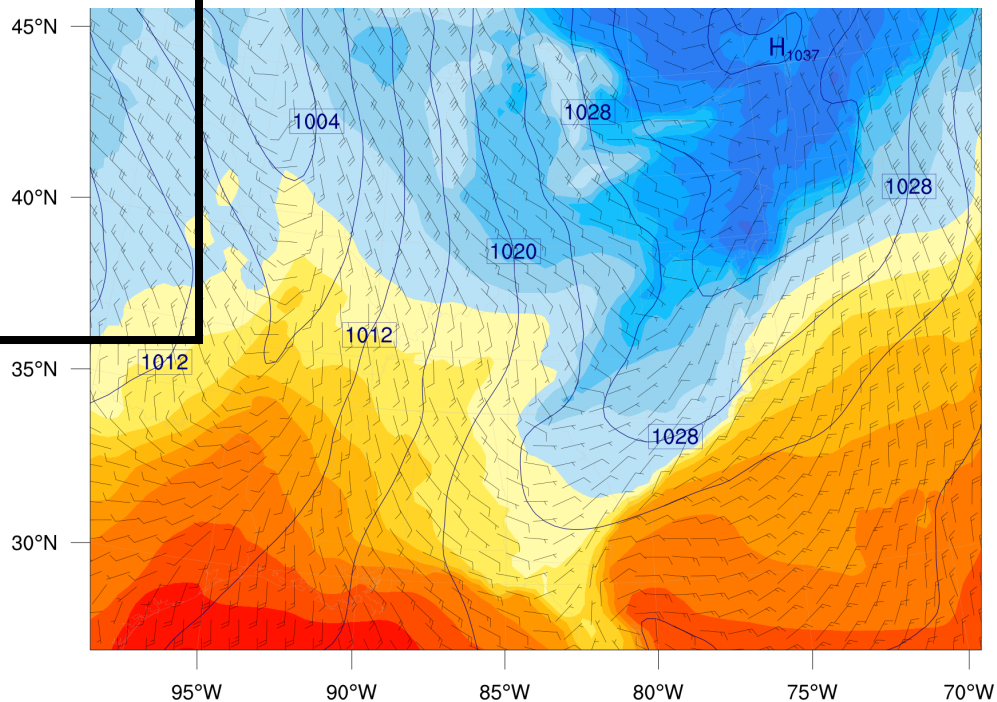
<i>opts@MainTitle</i>	Main title on the plot
<i>opts@MainTitlePos</i>	Main title position (default=left)
<i>opts@NoHeaderFooter</i>	Turn off headers & footers (default=False)
<i>opts@Footer</i>	Add model information as a footer (default=True)
<i>opts@InitTime</i>	Plot initial time on graphic (default=True)
<i>opts@ValidTime</i>	Plot valid time on graphic (default=True)
<i>opts@TimeLabel</i>	Label to use for valid time
<i>opts@TimePos</i>	Time position (default=right)
<i>opts@ContourParameters</i>	Contour parameters
<i>opts@FieldTitle</i>	Overwrite the field title
<i>opts@UnitLabel</i>	Overwrite the field units
<i>opts@PlotLevelID</i>	Add level information to field title
<i>opts@NumVectors</i>	Density of wind vector (<i>wrf_vector</i>) (default=25)

REAL-TIME WRF

Init: 2005-12-14_00:00:00
Valid: 2005-12-14_13:00:00

opts@MainTitle
opts@MainTitlePos

opts@InitTime
opts@ValidTime
opts@TimeLabel
opts@TimePos



*Resources for
wrf_contour &
wrf_vector*

opts@NoHeaderFooter

opts@Footer

OUTPUT FROM WRF V2.1.1 MODEL
WE = 98 ; SN = 84 ; Levels = 37 ; Dis = 30km ; Phys Opt = 2 ; PBL Opt = 1 ; Cu Opt = 1

wrf_map_overlays/wrf_overlays

```
plot = wrf_map_overlays (f, wks, (/contour,vector/), \  
                        pltres, mpres)  
plot = wrf_overlays (f, wks, (/contour,vector/), \  
                     pltres)
```

`pltres@NoTitles`

Turn off all titles

`pltres@CommonTitle`

Common title

`pltres@PlotTitle`

Plot title

`pltres@PanelPlot`

Whether a panel plot is to be drawn

`pltres@FramePlot`

Whether to advance the frame

*Resources for
wrf_overlays and
wrf_map_overlays*

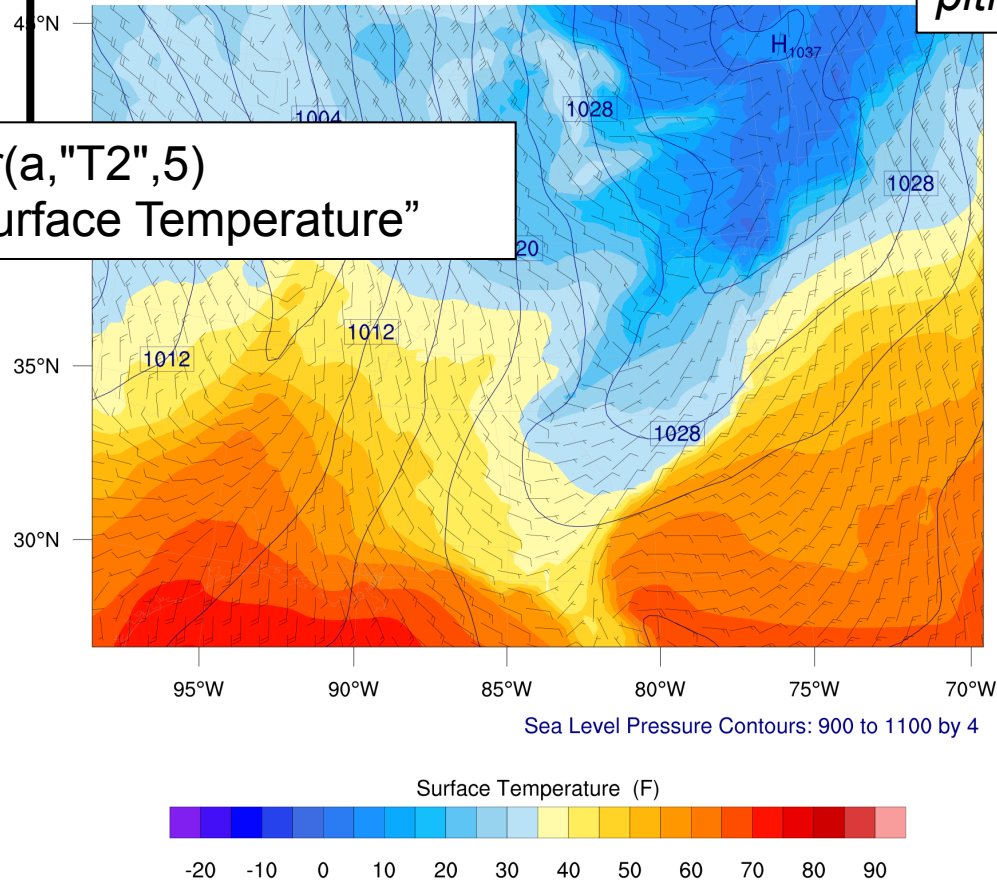
REAL-TIME WRF

Init: 2005-12-14_00:00:00
Valid: 2005-12-14_13:00:00

Surface Temperature (F)
Sea Level Pressure (hPa)
Wind (kts)

*pltres@NoTitles
pltres@CommonTitle*

```
t2 = wrf_user_getvar(a,"T2",5)  
t2@description = "Surface Temperature"
```



OUTPUT FROM WRF V2.1.1 MODEL
WE = 98 ; SN = 84 ; Levels = 37 ; Dis = 30km ; Phys Opt = 2 ; PBL Opt = 1 ; Cu Opt = 1

Where to find NCL scripts

NCL Examples Page:

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

WRF-NCL Examples Page:

<http://www.ncl.ucar.edu/Applications/wrf.shtml>



<http://www.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/>

Visualizing WRF with gsn_csm_xxx scripts

Why?

- More fine-tune control of plot options
- Plot WRF data on different map projection

<http://www.ncl.ucar.edu/Applications/wrfgsn.shtml>

Visualizing WRF with gsn_csm_xxx scripts

1. Use “`wrf_map_resources`” to set up map resources
2. Set `tfDoNDCOverlay` resource to True
3. Set `gsnAddCyclic` resource to False
4. Call one of the `gsn_csm_xxx_map` functions:
 - `gsn_csm_contour_map`
 - `gsn_csm_vector_map`
 - `gsn_csm_streamline_map`

Demo: plotting WRF using gsn_csm functions

Original WRF script

```
wrf_demo_plot_tc.ncl
```

Three GSN scripts with different levels of
customization

```
wrf_demo_plot_tc_gsn.ncl
```

```
wrf_demo_plot_tc_gsn_minor_custom.ncl
```

```
wrf_demo_plot_tc_gsn_major_custom.ncl
```

Original WRF script

```
a = addfile("wrfout_d01_2005-08-28_00:00:00","r")
tc = wrf_user_getvar(a,"tc",0)      ; Temperature (C)

wks = gsn_open_wks("x11","wrf_tc")

res                = True
res@cnFillOn       = True

contour = wrf_contour(a,wks,tc(0,:,:),res)
plot     = wrf_map_overlays(a,wks,contour,False,False)
```

Using gsn_csm_contour_map

```
a = addfile("wrfout_d01_2005-08-28_00:00:00","r")
tc = wrf_user_getvar(a,"tc",0)      ; Temperature (C)

wks = gsn_open_wks("x11","wrf_tc_gsn")

res = wrf_map_resources(a,True)

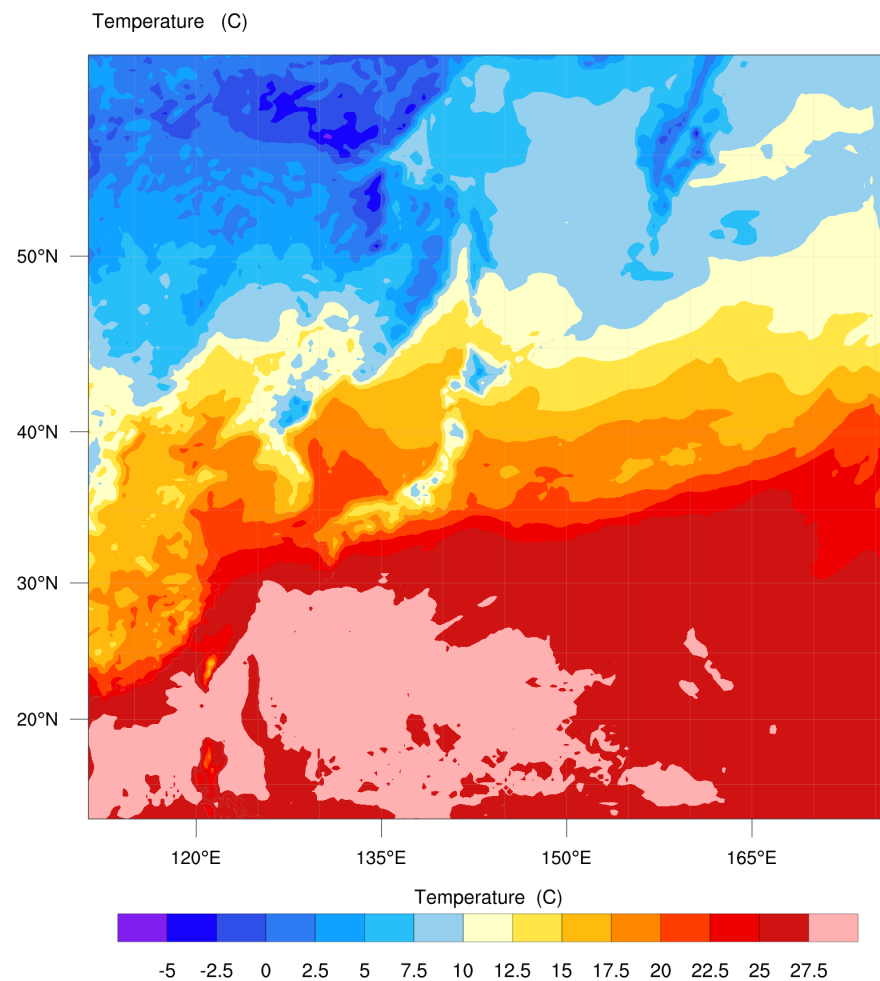
res@cnFillOn      = True
res@cnLinesOn     = False
res@tfDoNDCOverlay = True
res@gsnAddCyclic  = False

plot = gsn_csm_contour_map(wks,tc(0,:,:),res)
```

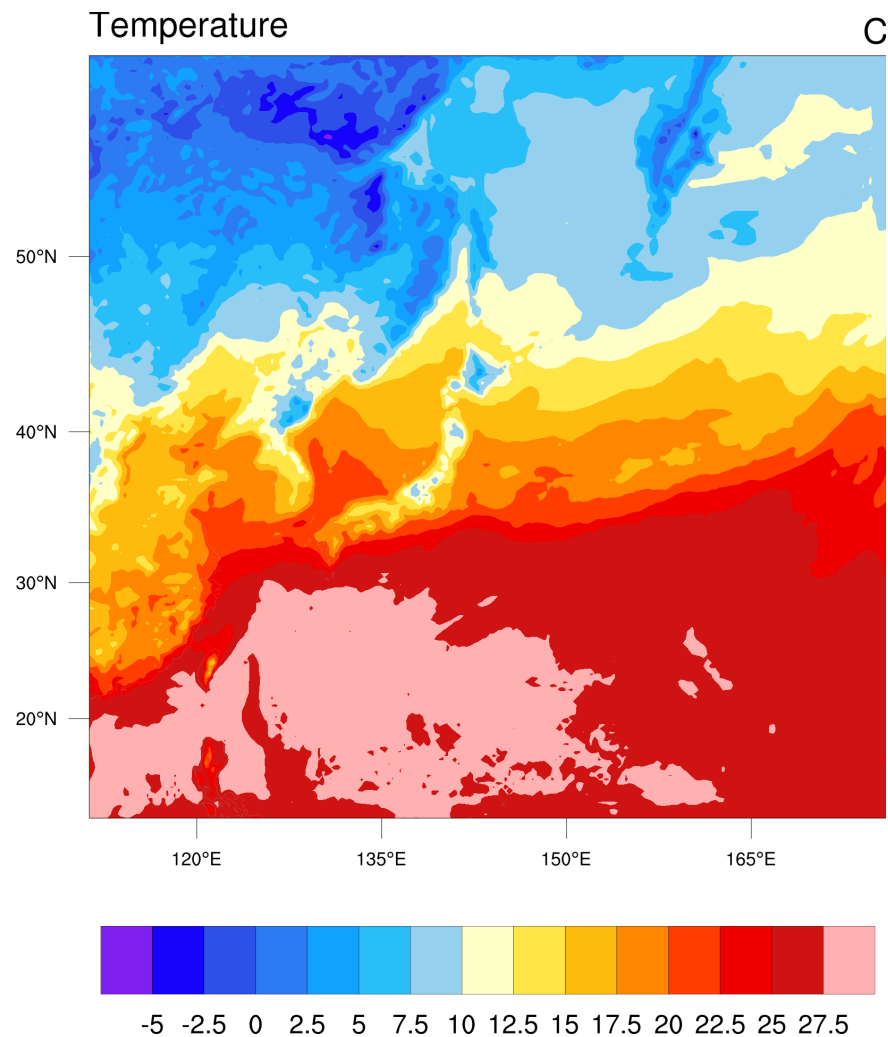
Will produce warnings that you can ignore:

```
warning:start_lat is not a valid resource in wrf_contour at this time
warning:start_lon is not a valid resource in wrf_contour at this time
warning:end_lat is not a valid resource in wrf_contour at this time
warning:end_lon is not a valid resource in wrf_contour at this time
warning:mpNestTime is not a valid resource in map at this time
```

Init: 2008-09-28_00:00:00



Original WRF script



gsn_csm script

OUTPUT FROM WRF V3.1.1 MODEL
WE = 207 ; SN = 198 ; Levels = 33 ; Dis = 27km ; Phys Opt = 2 ; PBL Opt = 1 ; Cu Opt = 1

gsn_csm_contour_map – minor customization

```
a = addfile("wrfout_d01_2005-08-28_00:00:00","r")
tc = wrf_user_getvar(a,"tc",0)      ; Temperature (C)

wks = gsn_open_wks("x11","wrf_tc_gsn")

res = wrf_map_resources(a,True)

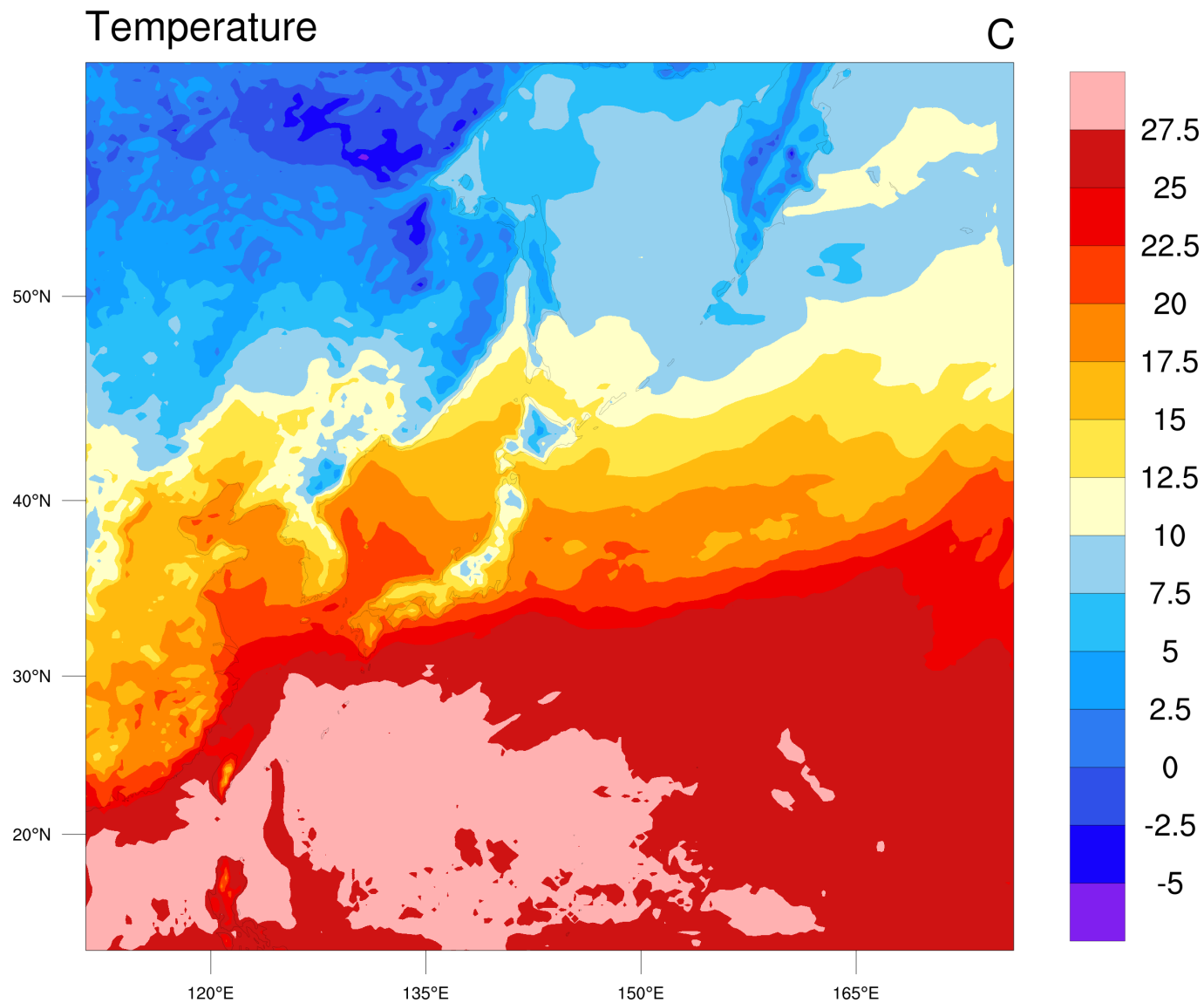
res@cnFillOn      = True
res@cnLinesOn     = False

res@tfDoNDCOverlay = True
res@gsnAddCyclic   = False

res@lbOrientation  = "Vertical"    ; vertical labelbar

contour = gsn_csm_contour_map(wks,tc(0,:,:),res)
```

Minor customization – vertical labelbar

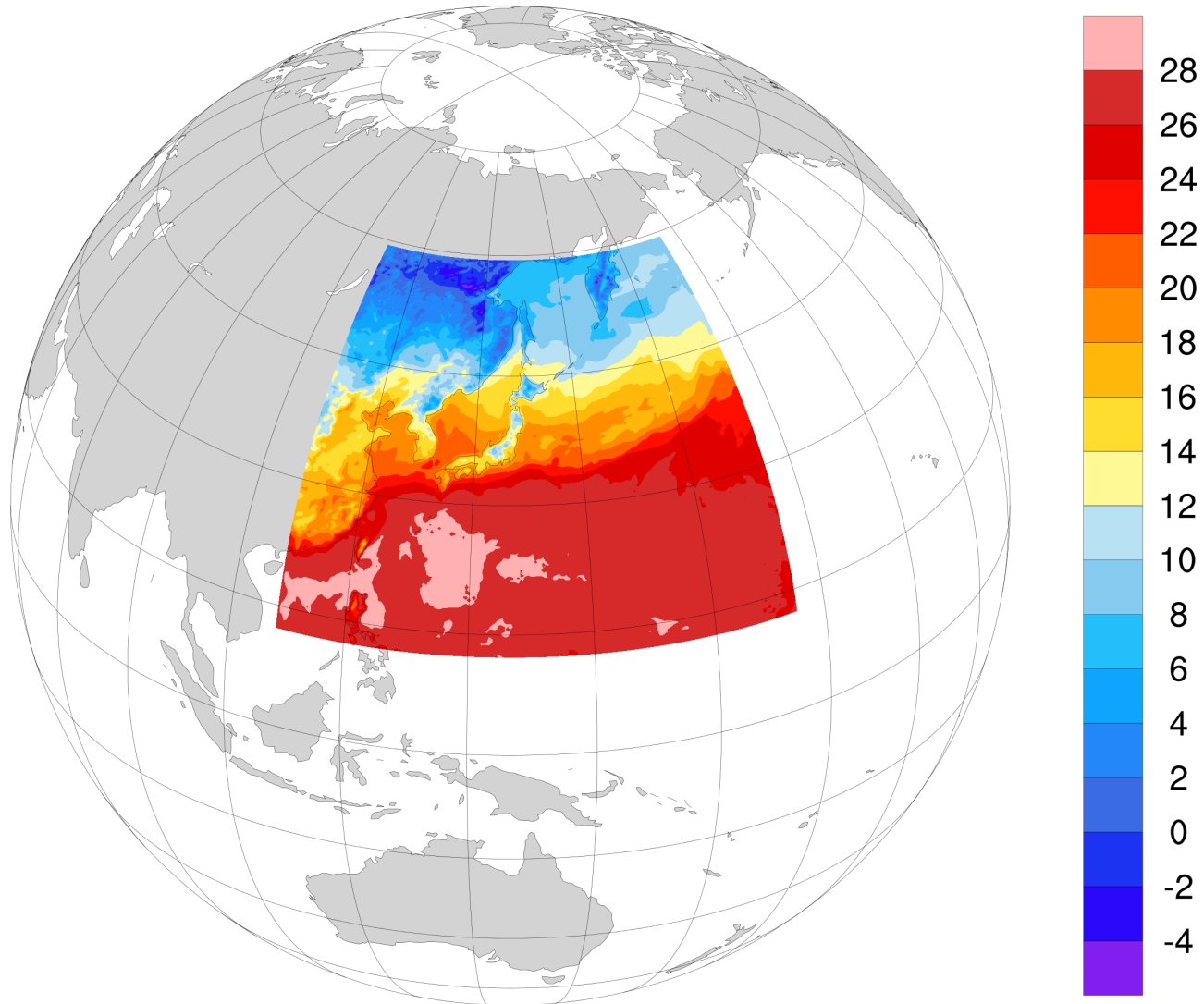


Major customization

wrfout_d01_2008-09-29_16:30:00

Temperature

degC

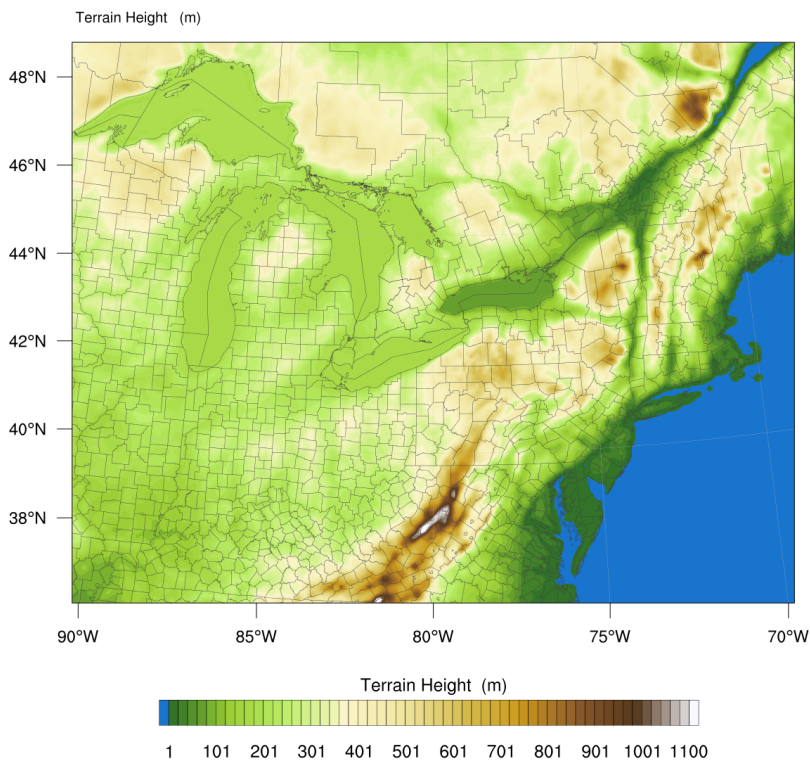


Adding extra features to WRF-NCL graphics

- Set **PanelPlot** to True to “keep” graphics
- Use **gsn_add_text**/ **gsn_add_polyline** / **gsn_add_polymarker** to add text, lines, etc.
- Use **gsn_add_shapefile_polylines** to add shapefile outlines
- Use **gsn_coordinates** to add lines or markers at WRF grid locations

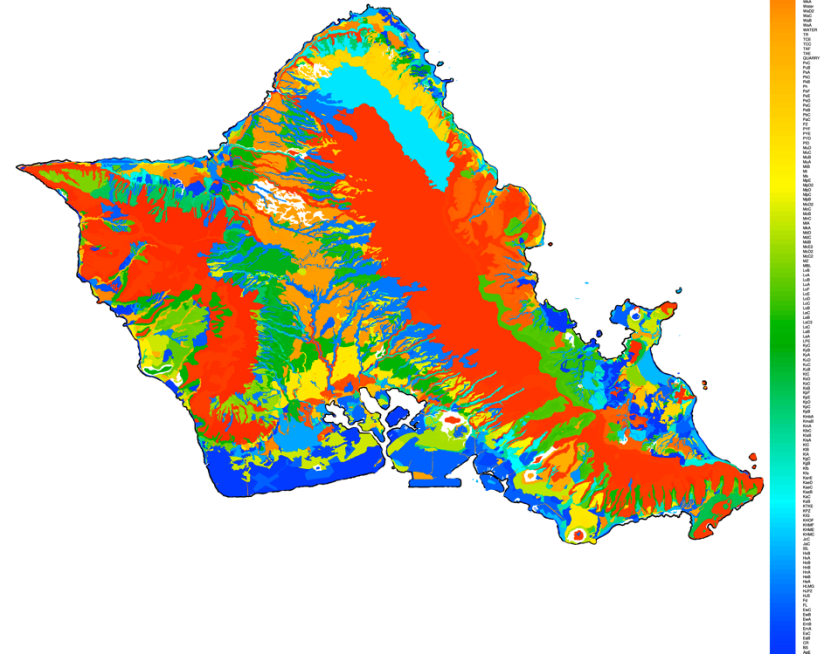
NCL has support for
shapefiles, allowing you
to use the numerous free
shapefiles for adding your
own map outlines

Init: 2002-07-01_00:00:00

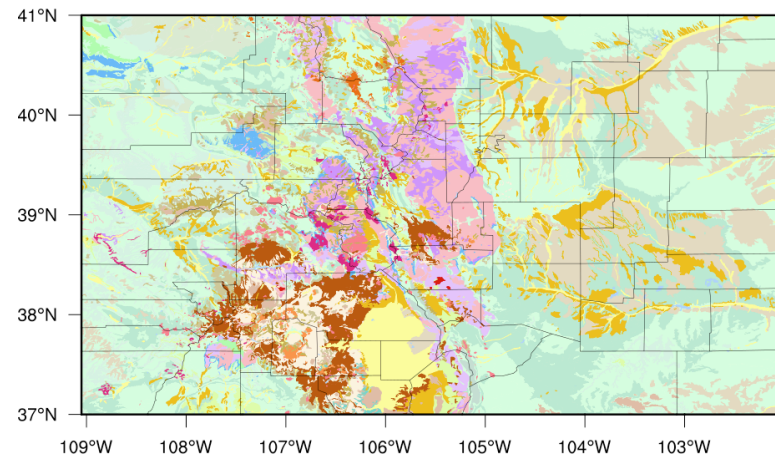


OUTPUT FROM WRF V3.3 MODEL
WE = 225 ; SN = 175 ; Levels = 28 ; Dis = 8km ; Phys Opt = 6 ; PBL Opt = 1 ; Cu Opt = 1

O'ahu, Hawai'i (soil)

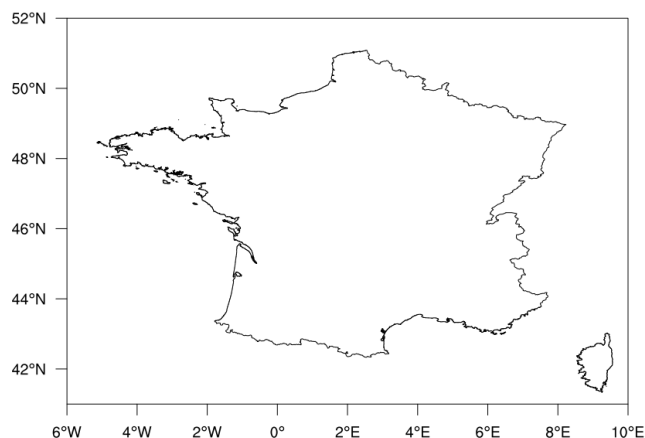


Geologic units and structural features in Colorado

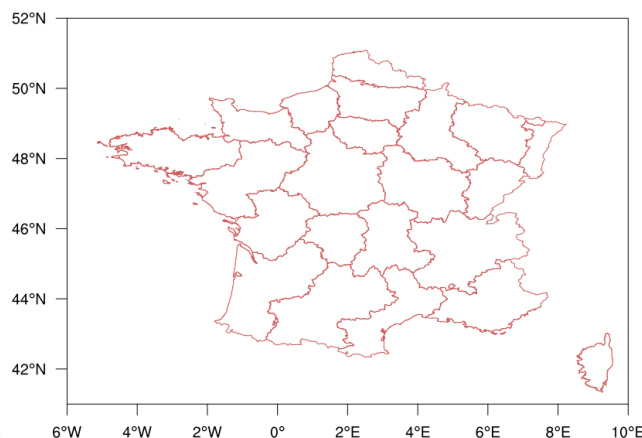


Shapefiles give you detailed geographical outlines

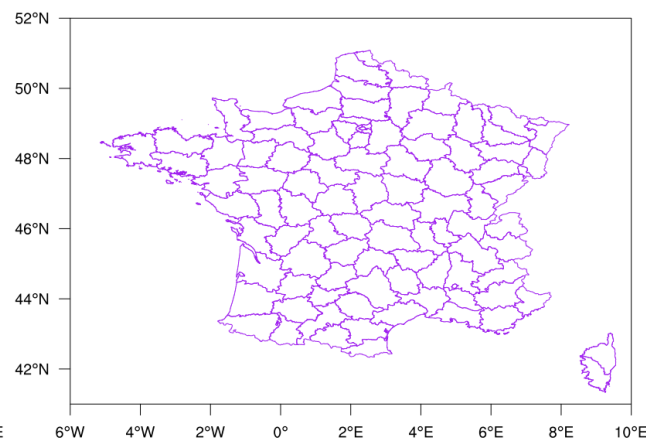
FRA_adm/FRA_adm0.shp



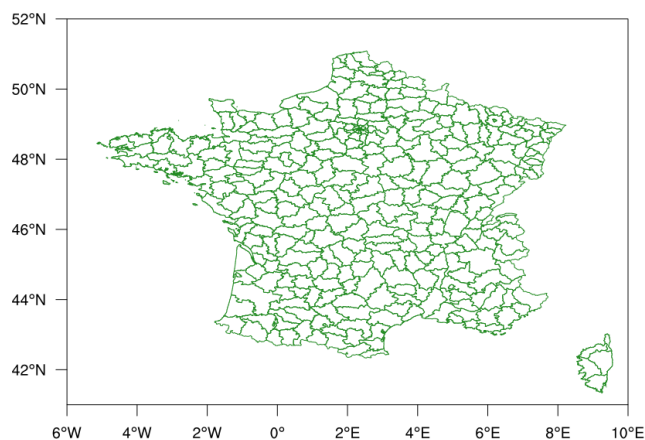
FRA_adm/FRA_adm1.shp



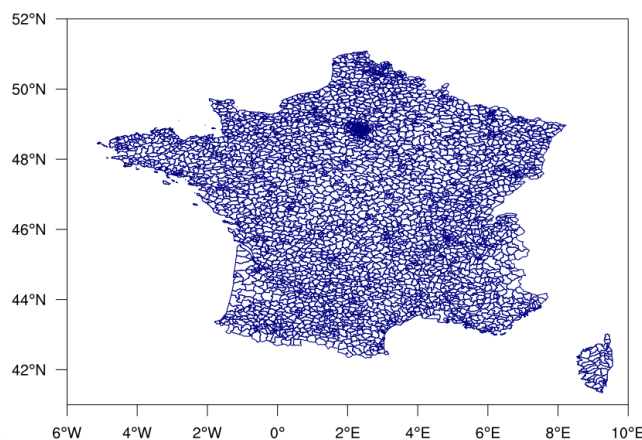
FRA_adm/FRA_adm2.shp



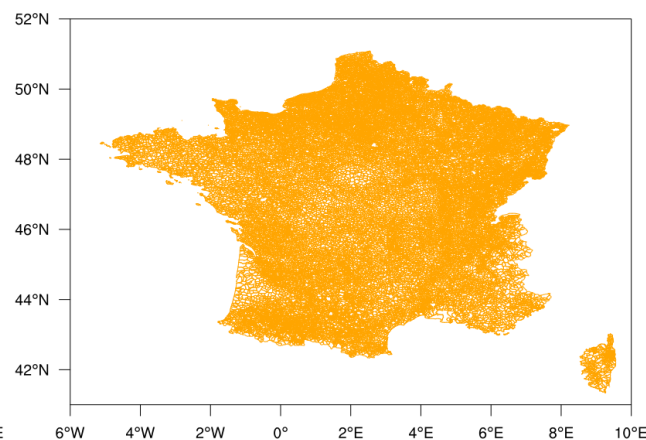
FRA_adm/FRA_adm3.shp



FRA_adm/FRA_adm4.shp



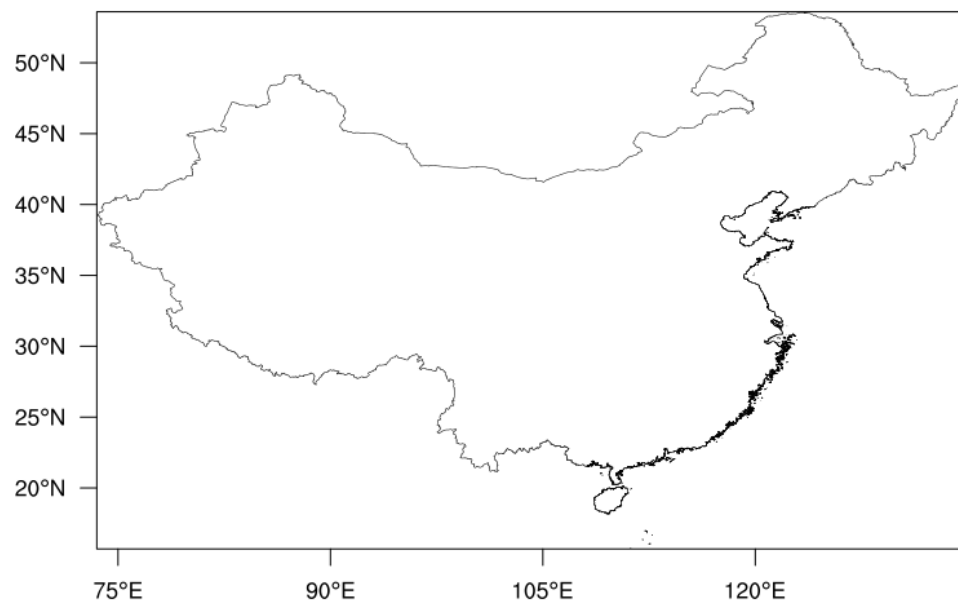
FRA_adm/FRA_adm5.shp



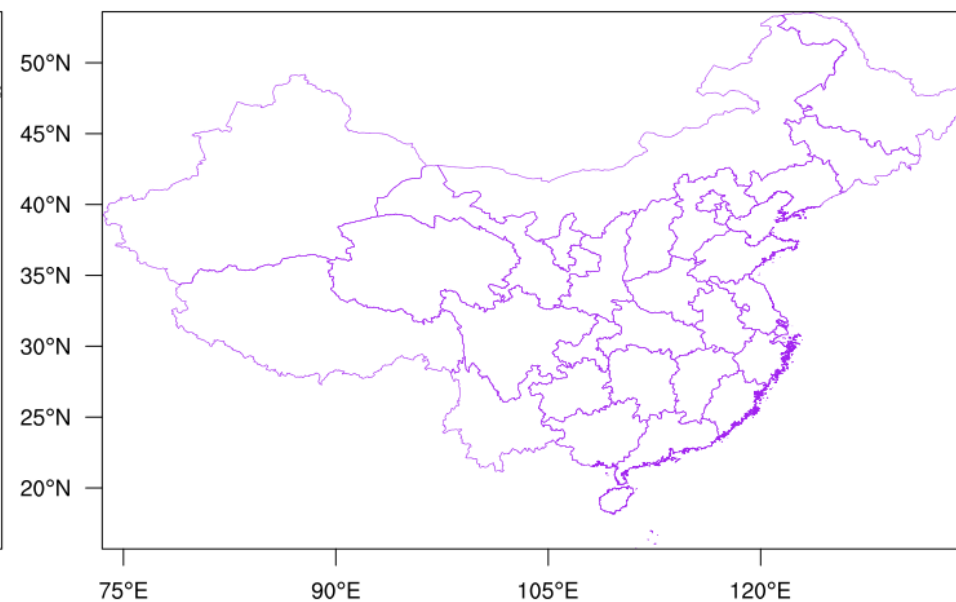
Global Administrative Areas database (<http://www.gadm.org>) offers consistent administrative boundaries at many levels. The level 0 database (nations) is good to use for global or mesoscale results, level 1 is the first level of sub-national administration (typically states/provinces and territories) while level 2 offers the second level of administration and is potentially useful for high-resolution plots.

China shapefiles from gadm.org/country

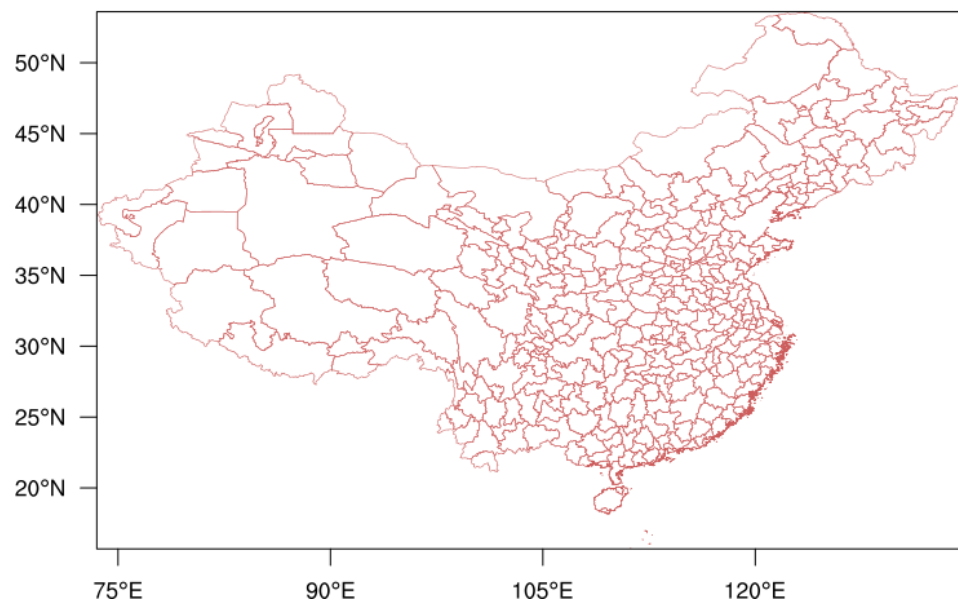
CHN_adm0.shp



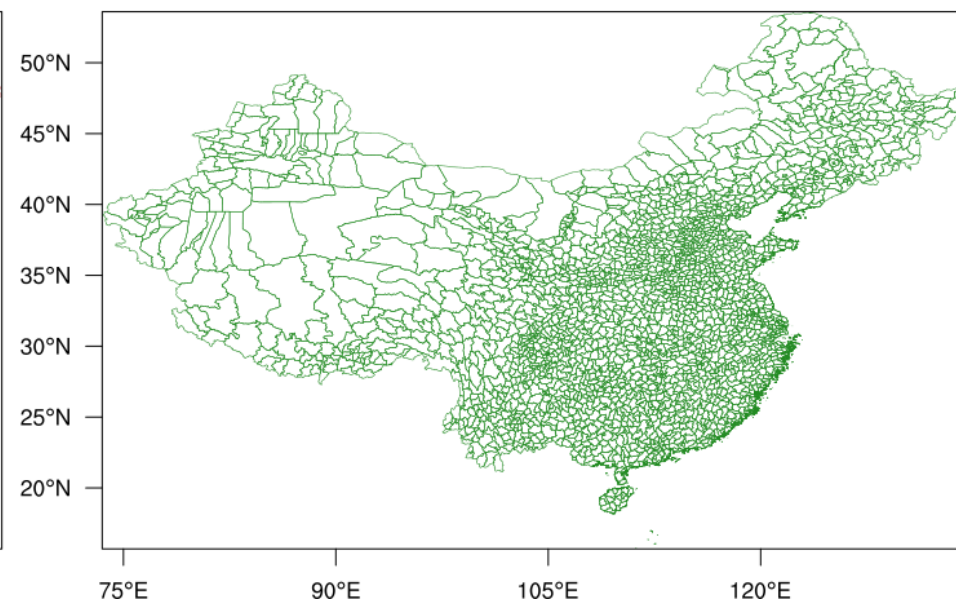
CHN_adm1.shp



CHN_adm2.shp

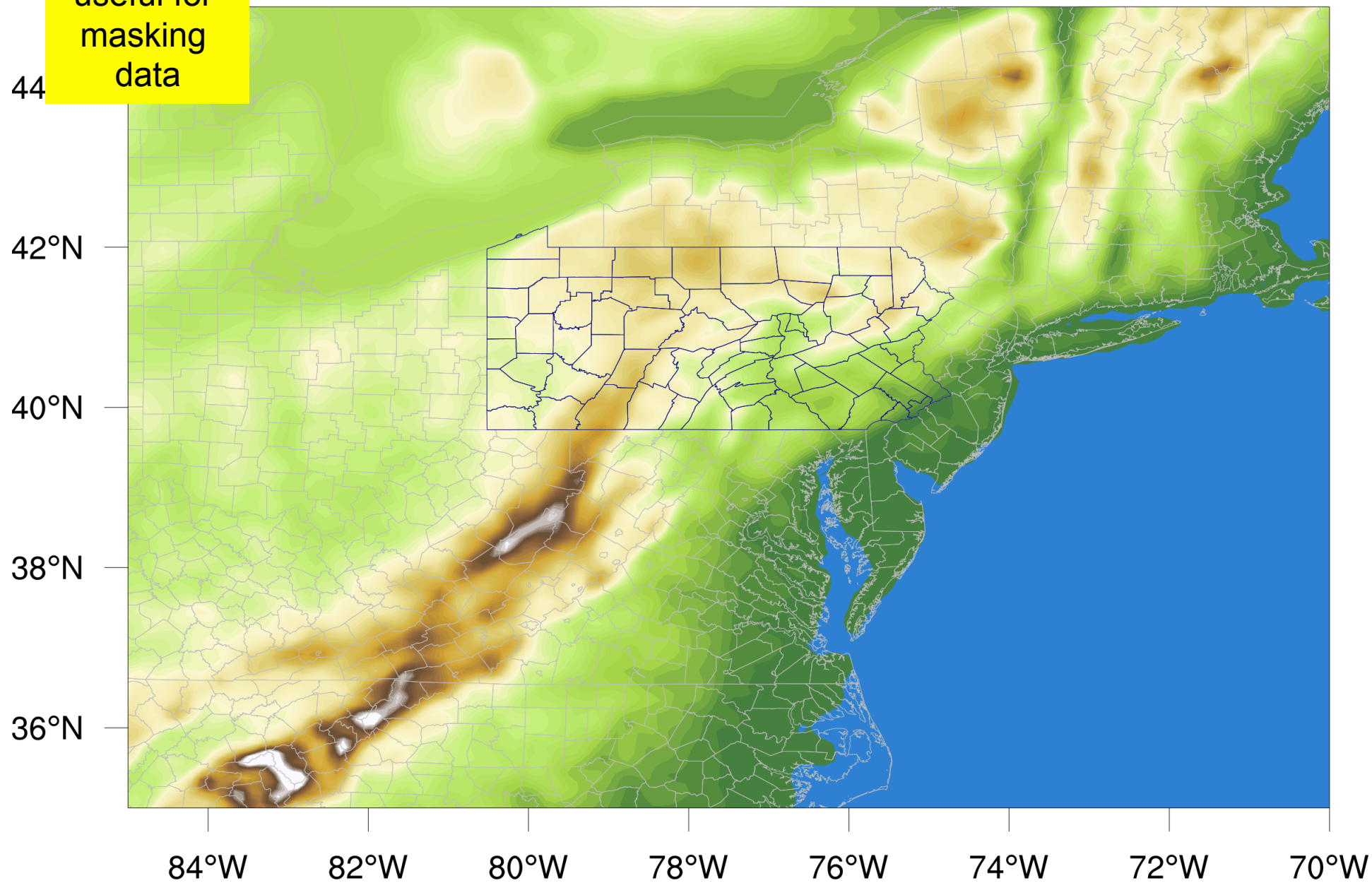


CHN_adm3.shp

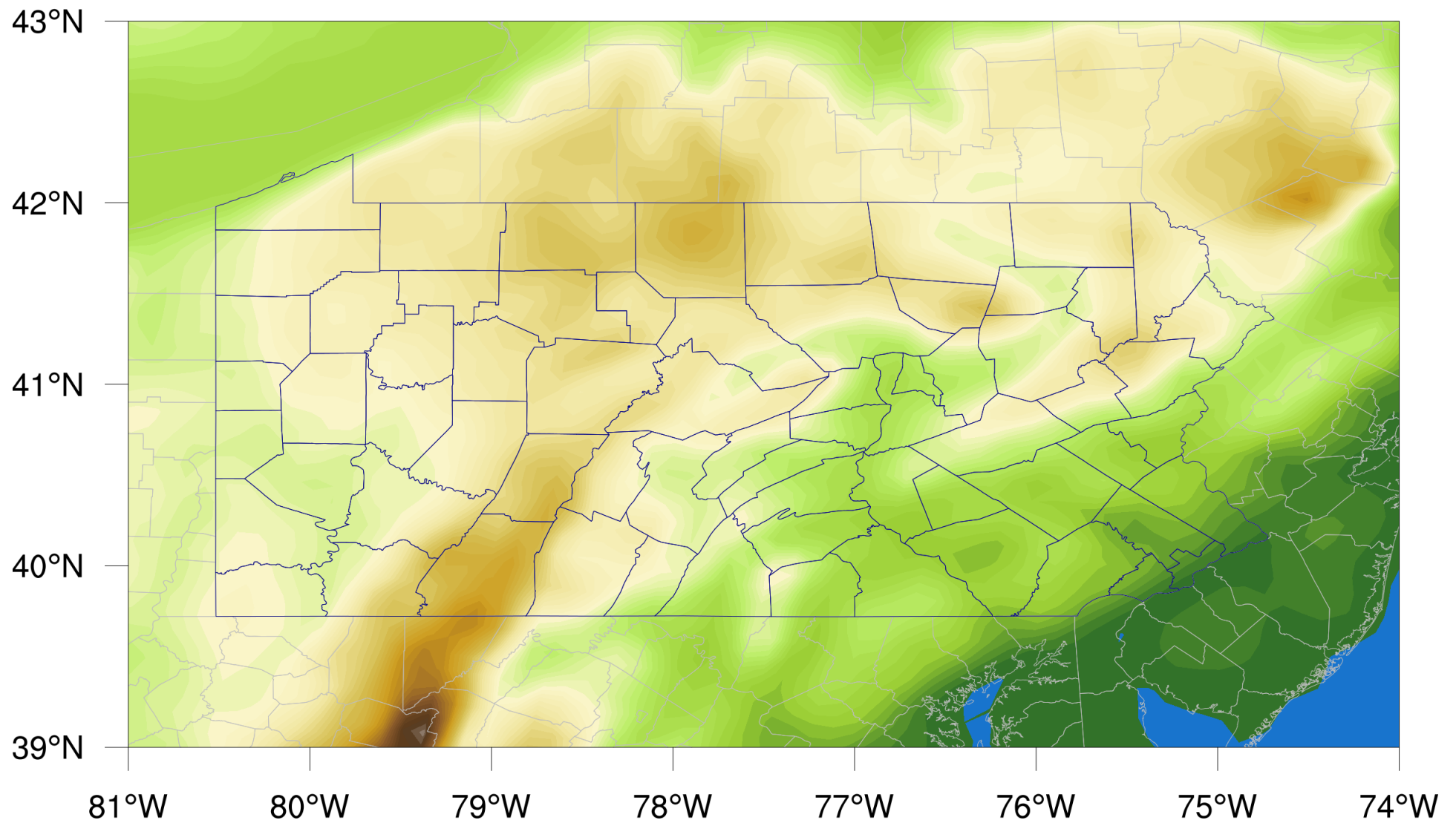


Shapefiles
useful for
masking
data

Original data

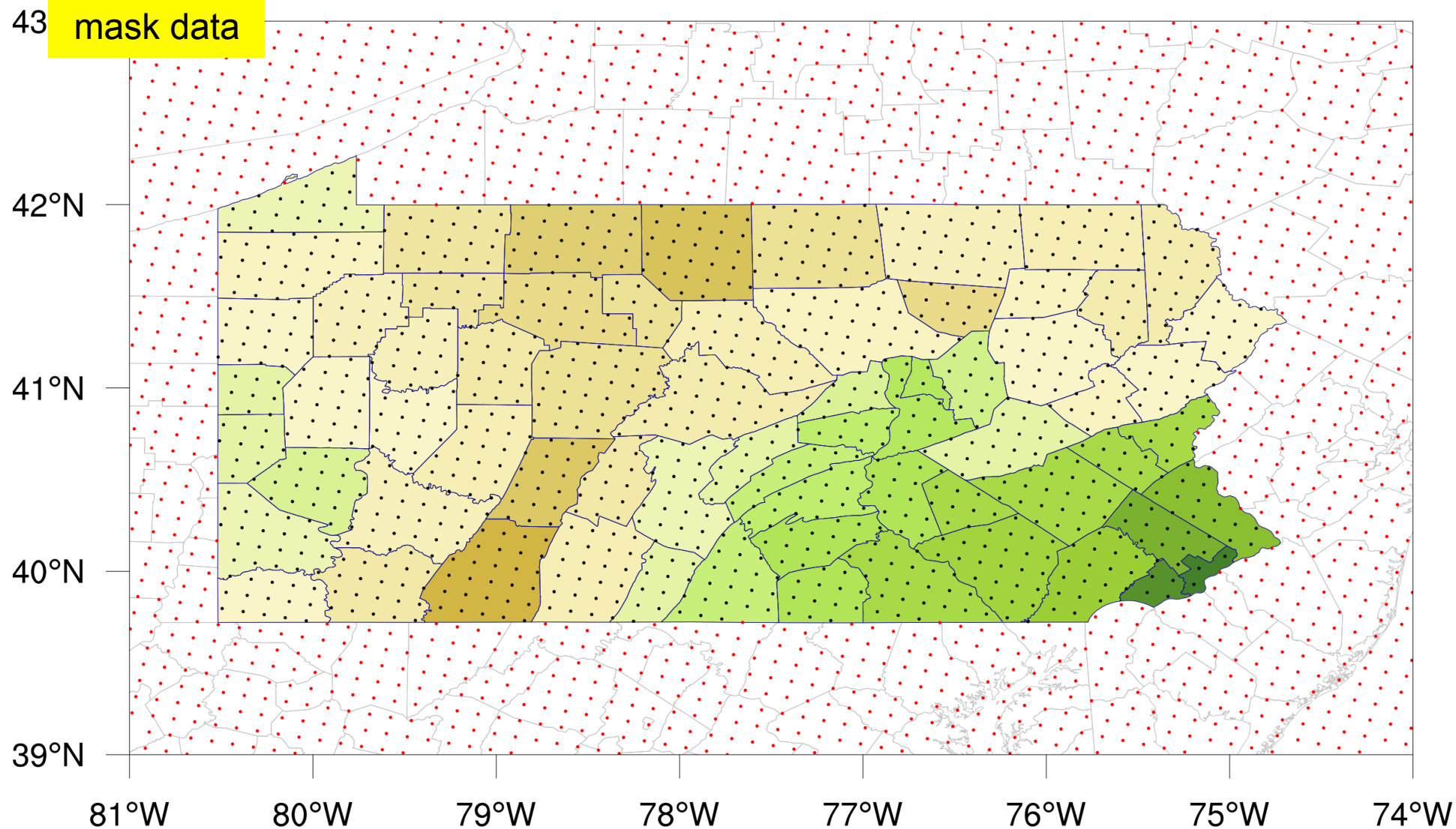


Original data

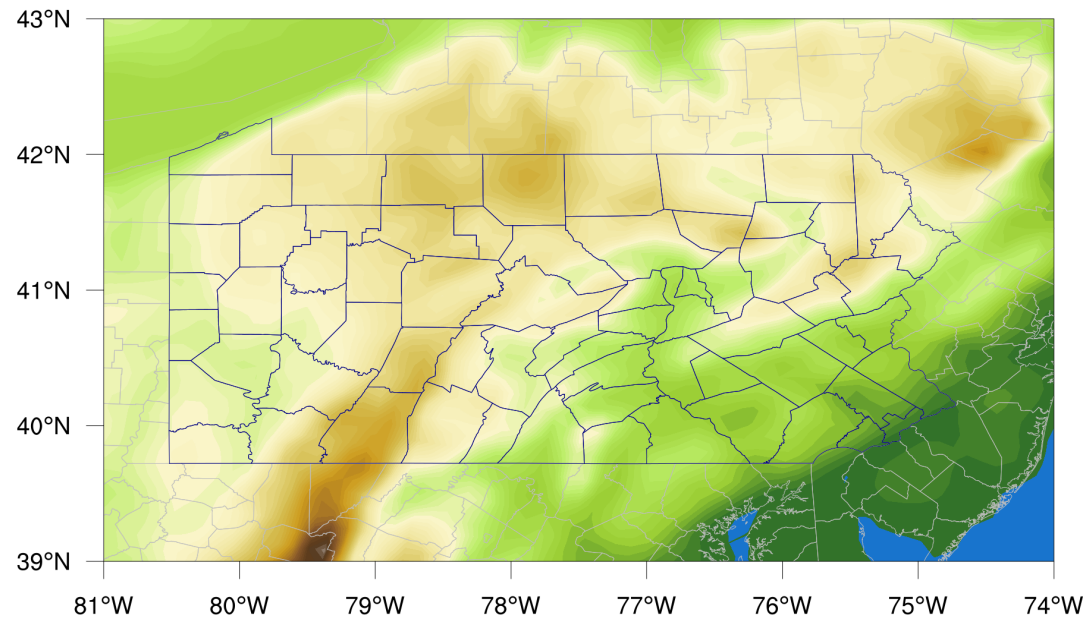


Using
shapefile
outlines to
mask data

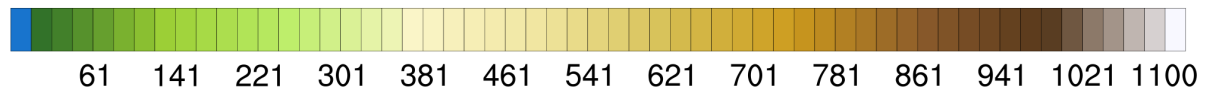
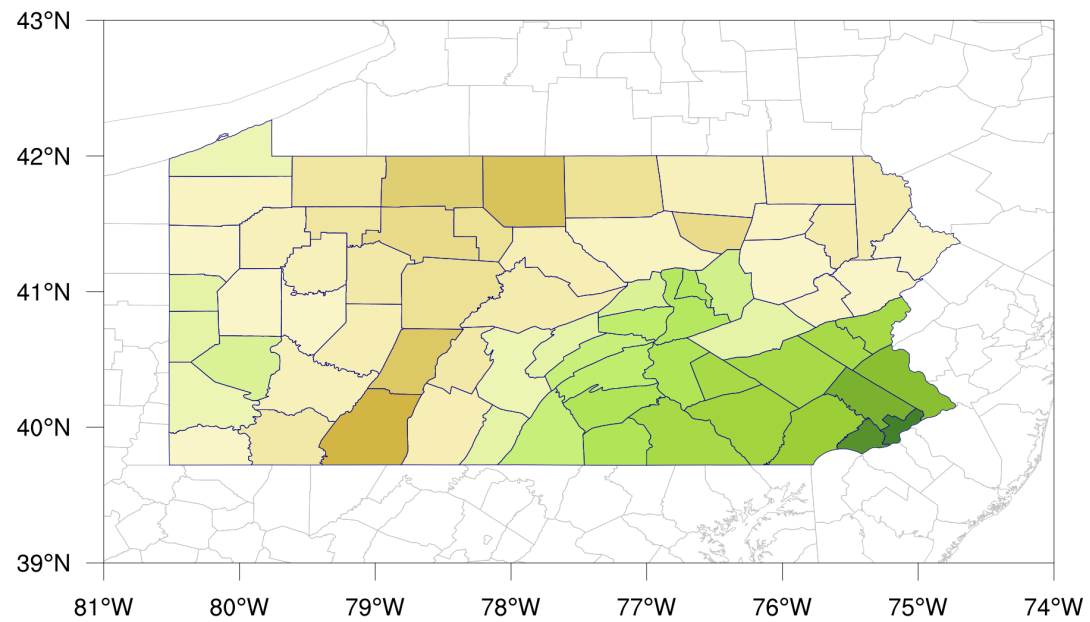
Data averaged over counties



Original data



Data averaged over counties



Demo: adding shapefile outlines

Shapefiles downloaded from

<http://www.gadm.org/country/>

```
wrf_demo_plot_slp_shapefiles.ncl
```

wrf_demo_plot_slp_shapefiles.ncl

```
f = addfile("wrfout_d01_2008-09-29_16:30:00","r")
slp = wrf_user_getvar(f,"slp",0)

wks = gsn_open_wks("x11","wrf_slp_shp")

res = True
res@cnFillOn = True ; Turn on color fill

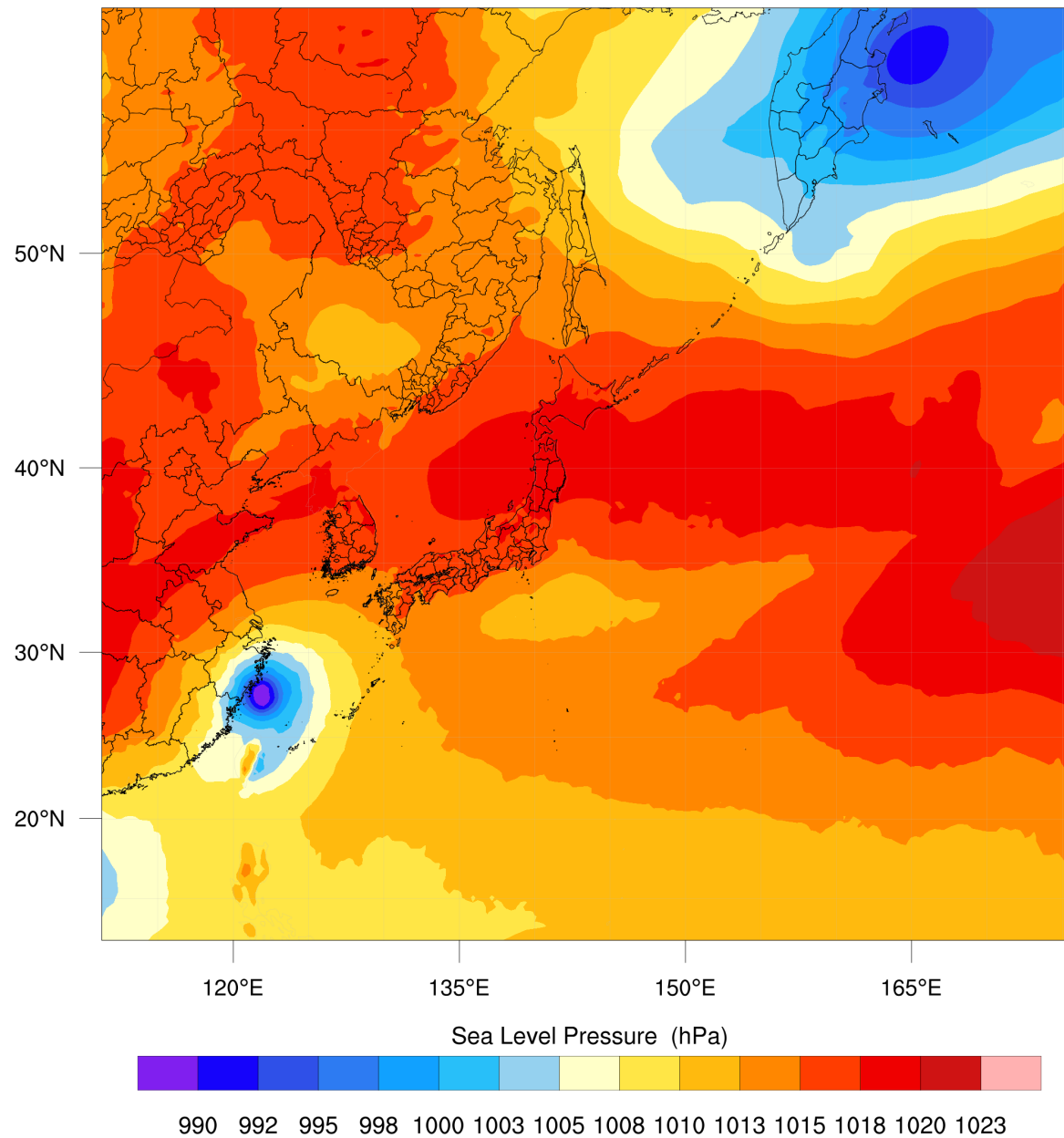
res@NoHeaderFooter = True ; Turn off top/bottom titles
contour = wrf_contour(f,wks,slp,res)

;--- Create contours over a map (nothing is drawn because PanelPlot is True)
pltres = True
pltres@PanelPlot = True ; Tells wrf_map_overlays to not draw plot
pltres@NoTitles = True ; Turn off titles
plot = wrf_map_overlays(f,wks,contour,pltres,False)

;--- Add outlines from Asian countries
lnres = True
lnres@gsLineColor = "Gray25"
jpn_id = gsn_add_shapefile_polylines(wks,plot,"JPN_adm/JPN_adm1.shp",lnres)
chn_id = gsn_add_shapefile_polylines(wks,plot,"CHN_adm/CHN_adm1.shp",lnres)
kor_id = gsn_add_shapefile_polylines(wks,plot,"KOR_adm/KOR_adm1.shp",lnres)
rus_id = gsn_add_shapefile_polylines(wks,plot,"RUS_adm/RUS_adm2.shp",lnres)

draw(plot) ; Now draw the plot
frame(wks) ; and advance the frame
```

China, Japan, Korea, Russia shapefile outlines added



NCL debugging tips

- Start with an existing script, if possible
- Use editor enhancements for coloring of syntax, functions, etc
- Use indentation (even though not needed)
- Use “ncl_filedump” to look at file quickly
- Use `printVarSummary`, `printMinMax`, `print` to examine variables

NCL debugging tips (cont'd)

- Read documentation for functions
- For graphics, make sure spelling the resource name correctly
- Read errors and warnings carefully 😊

Customizing your NCL graphics environment

~/.hluresfile

Download “**.hluresfile**” file, put in home directory

- Change default color map
- Change size of X11 window
- Change size of PNG images
- **WRF-NCL users:** use to change the default color map

<http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml>

Sample .hluresfile

```
! Set the desired color map
! The default is ncl_default
*wkColorMap : BlAqGrYeOrReVi200
```

```
! Make default X11 window larger
! (the default is 512x512)
```

```
*windowWorkstationClass*wkWidth   : 1000
*windowWorkstationClass*wkHeight  : 1000
```

```
! Make default PNG window larger
! (the default is 1024x1024)
```

```
*imageWorkstationClass*wkWidth    : 2500
*imageWorkstationClass*wkHeight   : 2500
```

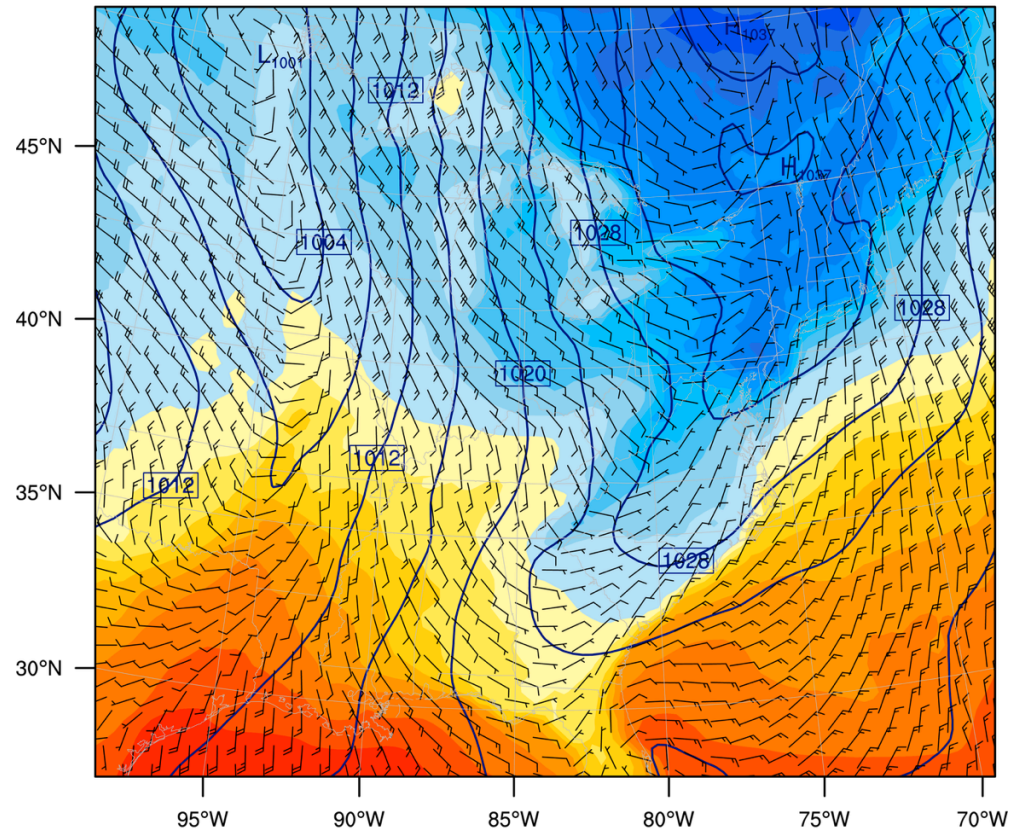

NCL default color table

REAL-TIME WRF

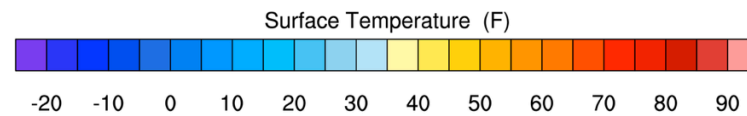
Init: 2005-12-14_00:00:00

Valid: 2005-12-14_13:00:00

Surface Temperature (F)
Sea Level Pressure (hPa)
Wind (kts)



Sea Level Pressure Contours: 900 to 1100 by 4



OUTPUT FROM WRF V2.1.1 MODEL

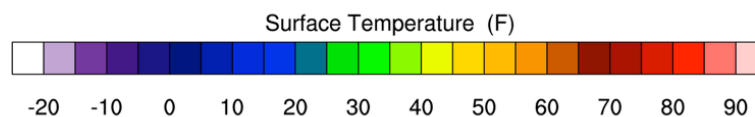
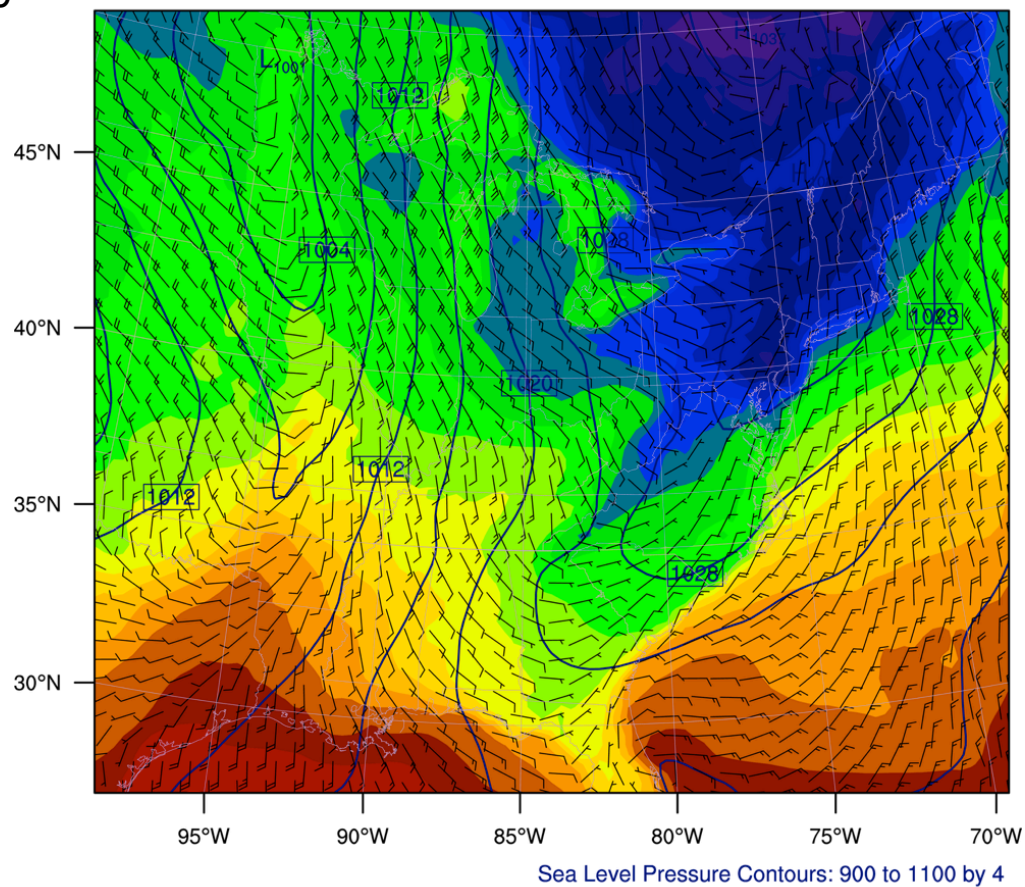
WE = 98 ; SN = 84 ; Levels = 37 ; Dis = 30km ; Phys Opt = 2 ; PBL Opt = 1 ; Cu Opt = 1

Changed default
color map to
BIAqGrYeOrReVi200

REAL-TIME WRF

Init: 2005-12-14_00:00:00
Valid: 2005-12-14_13:00:00

Surface Temperature (F)
Sea Level Pressure (hPa)
Wind (kts)



OUTPUT FROM WRF V2.1.1 MODEL
WE = 98 ; SN = 84 ; Levels = 37 ; Dis = 30km ; Phys Opt = 2 ; PBL Opt = 1 ; Cu Opt = 1

Python tools

- **PyNIO** – reads same data formats as NCL
- **PyNGL** – based on NCL graphics
- **WRF-Python** – New interface, still being developed
- Jupyter Notebook – tool that combines documentation and live code (Python, R) in one environment
- netcdf4-python – reads NetCDF3/4
- matplotlib / basemap – defacto 2D graphics package for Python

Why Python?

- Mainstream, open source language
- Shares similarities with NCL
- Rapidly growing scientific community
- Widely used across many disciplines
- Demand for high-quality 2D visualizations
- Demand for specialized analysis functions
- Capitalize on the work of others
- More collaboration opportunities
- Contribution to Python community

Demos: Various Python tools

Jupyter Notebooks are provided on the workshop page if you want to try them.

Questions?

Useful links can be found on the workshop tutorial page:

http://www.ncl.ucar.edu/Training/Tutorials/WRF_Users_Workshop/

Mary Haley
haley@ucar.edu