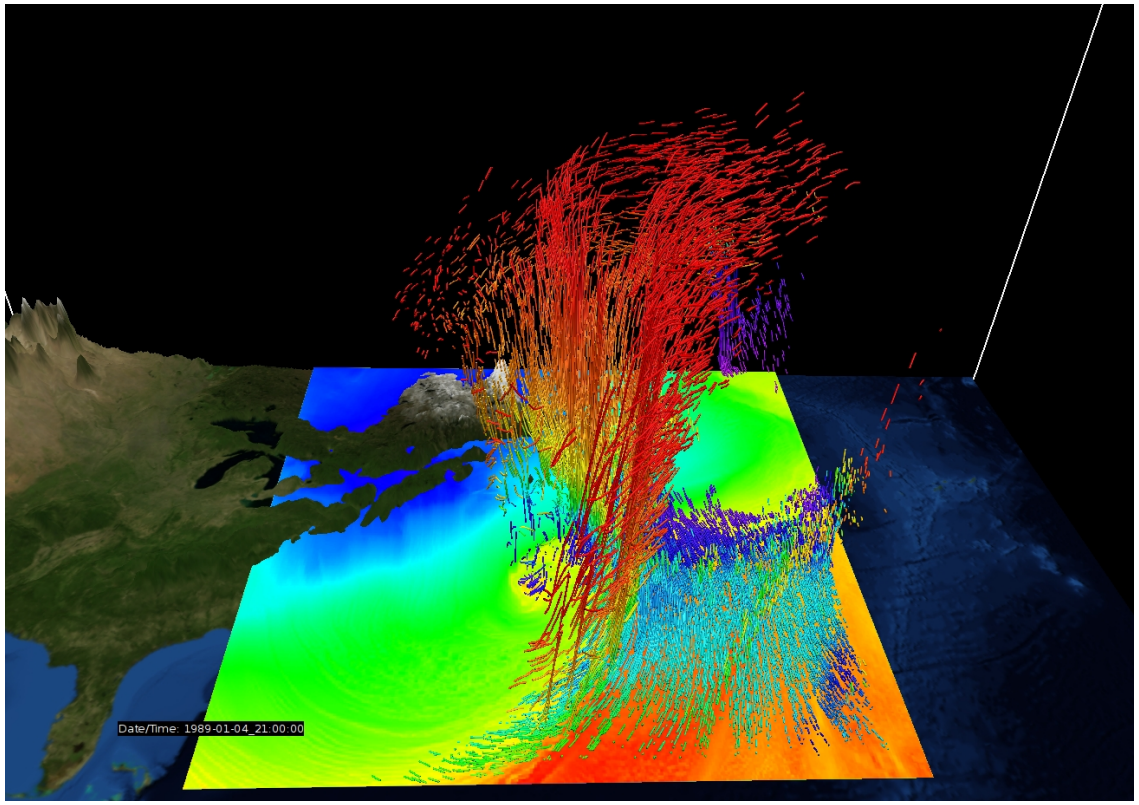
Post-processing Tools



Kelly Werner & Abby Jaye

October 2019

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

NCAR-supported Post-processing Packages

http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/contents.html

Package	Users' Guide Page #	Information
NCL	9-2	Graphical package Supported by NCAR/CISL (forum.mmm.ucar.edu and ncl-talk@ucar.edu)
ARWpost	9-29	Converter (GrADS) (forum.mmm.ucar.edu)
RIP4	9-20	Converter and interface to graphical Package, NCAR graphics (forum.mmm.ucar.edu)
UPP	9-36	Converter (GrADS & GEMPAK) (upp-help@ucar.edu)
VAPOR	9-38	Converter and graphical package Supported by VAPOR (vapor@ucar.edu)
IDV	None – see unidata.ucar.edu	GRIB (from UPP) GEMPAK (from wrf2gem) Vis5d CF compliant data (from wrf_to_cf) Supported by unidata (support@unidata.ucar.edu)
GEMPAK	None - see: unidata.ucar.edu/software/gempak	Data from wrf2gem or UPP Supported by unidata (support@unidata.ucar.edu)

Choosing the Right Tool

- Can it read your data?
- Will you need to pre-process the data first?
- Is it purely a visualization tool, or does it include post-processing?
- Can it handle big datasets?
- Which diagnostic/statistical functions does it have?
- How easy is it to add diagnostics?
- 3D or 2D visualization?
- Can it handle staggered grids?
- How is data below the ground handled?
- Vertical grids?
- How are model time stamps handled?
- Easy to use?
- Cost of package?
- How well supported is it?

Data Handling

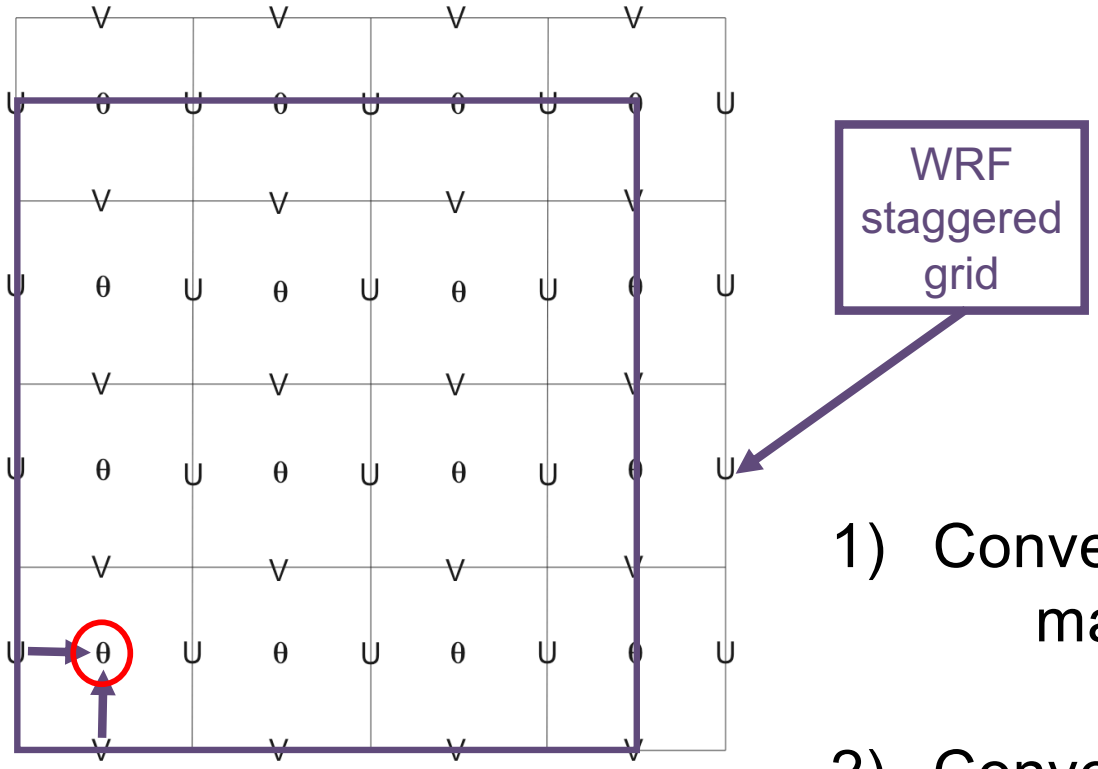
	NCL	RIP4	GrADS	UPP	VAPOR	IDV
netCDF		ripdp	ARWpost	converter	converter	converter
GRIB						
ASCII						
shapefiles						
geogrid & metgrid output						
intermediate file format						
wrfinput data						
Idealized data						
wrfoutput						
big data						

Post-processing

	NCL	RIP4	GrADS	UPP	VAPOR	IDV
Post-processing						
Data output						
3D						
diagnostics	some	a lot	some	some	limited	limited
Add diagnostics	Very easy	easy	easy	Relatively easy	Not as easy	Not as easy
Vertical output Coordinate	Model pressure height	Model pressure height	Model pressure height	pressure	model	model
Extrapolate Below ground						

Model Staggering

Why is a converter necessary if a package can display netCDF files?



1) Converter co-locates data to mass points

2) Converter translates variables
- e.g., “T” is not really temp. Must add 300 for actual temp (K)

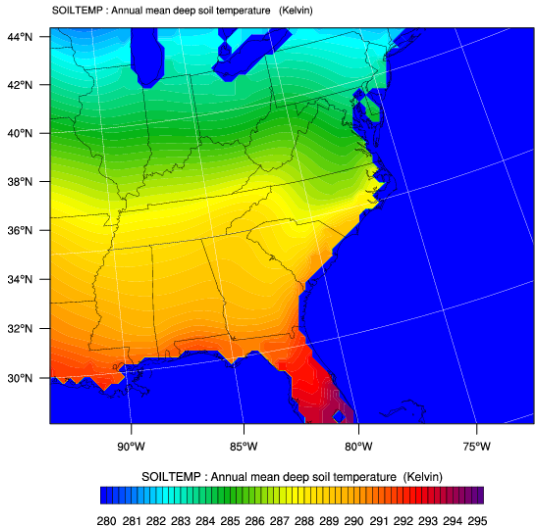

NCL

NCL: General Information

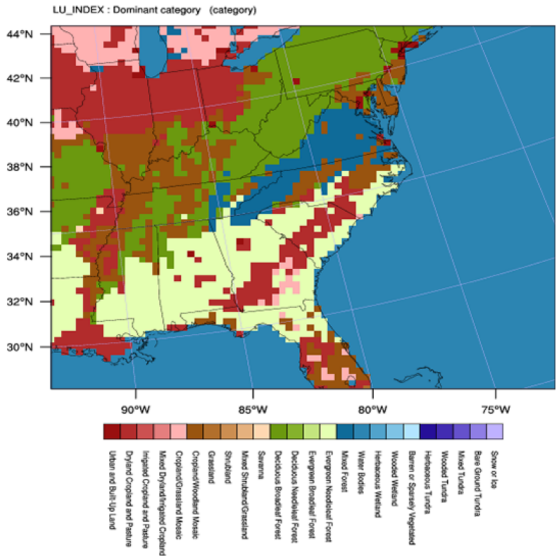
- NCAR Command Language (NCL)
- Website: http://www.ncl.ucar.edu
- Reads WRF-ARW data directly
- Can generate many types of graphical plots
  - Horizontal
  - Cross-section
  - SkewT
  - Meteogram
  - Panel

NCL: Example Plots

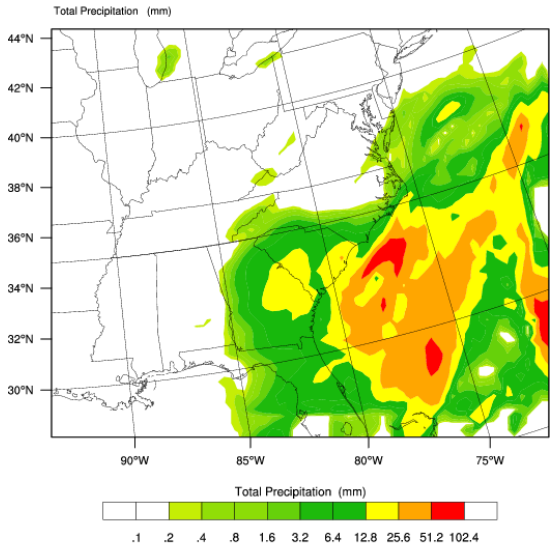
Soil Temperature



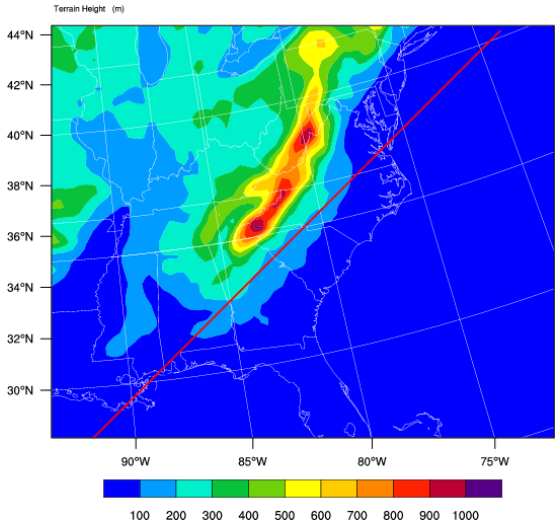
LU-INDEX



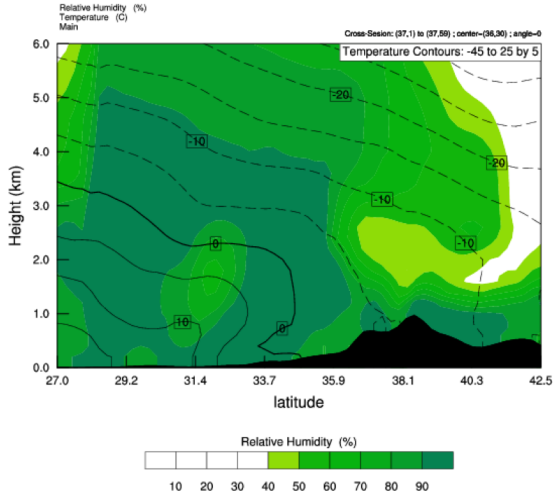
Total Precipitation



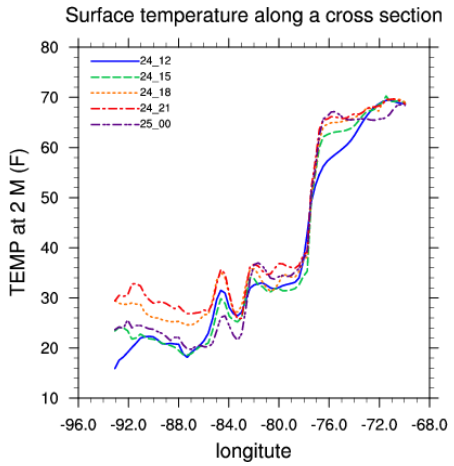
Terrain Height



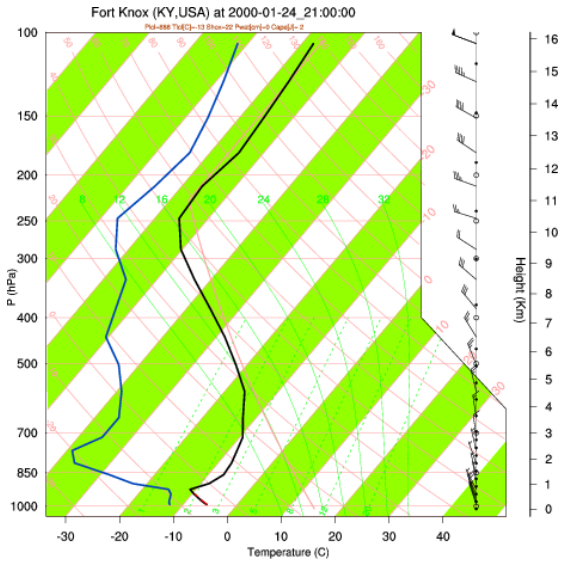
Relative Humidity Vertical X-section



Sfc Temperature Horizontal X-section

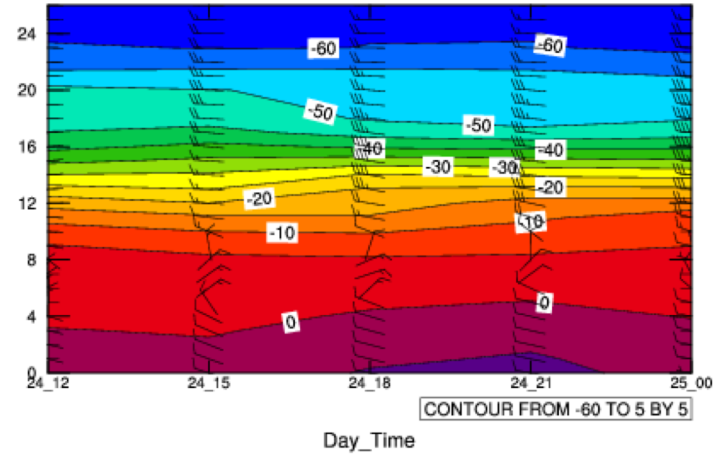


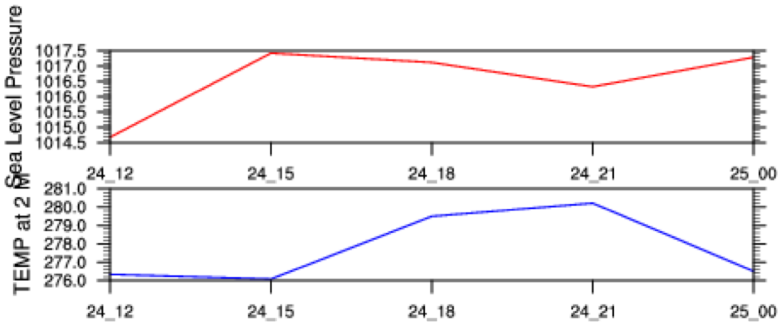
NCL: Example Plots

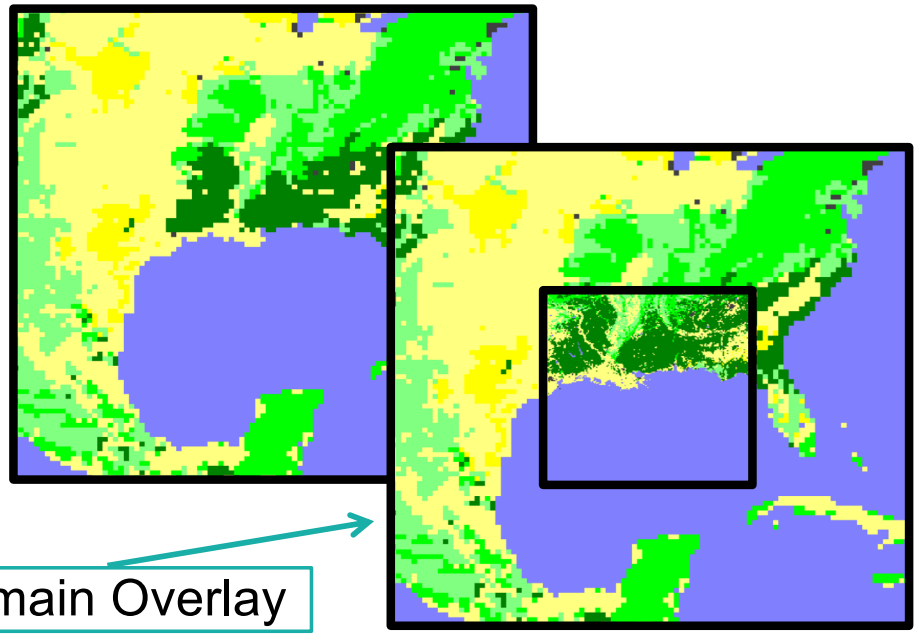


Skew-T Diagram

Time Series

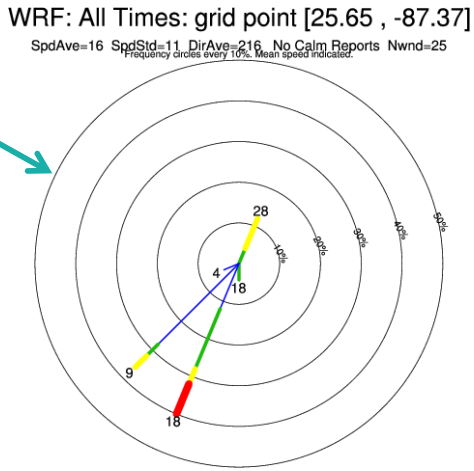






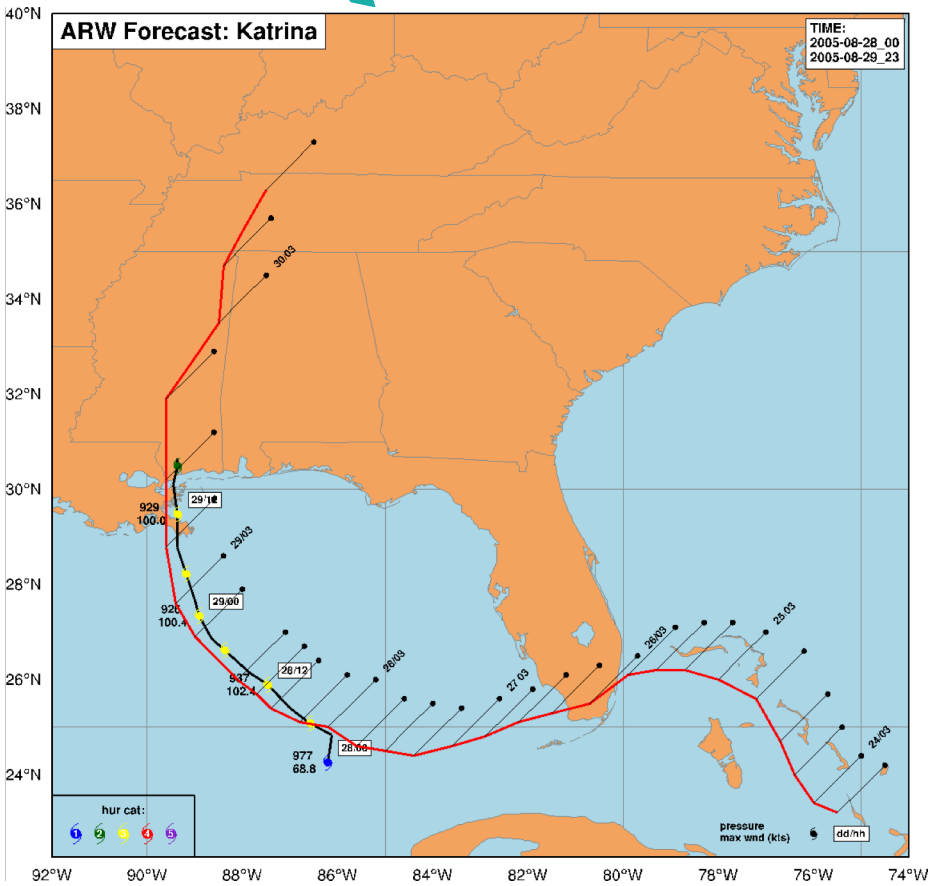
Domain Overlay

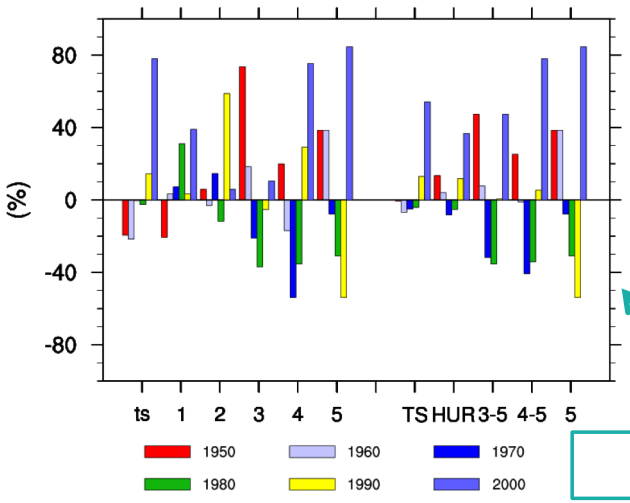
Wind Rose



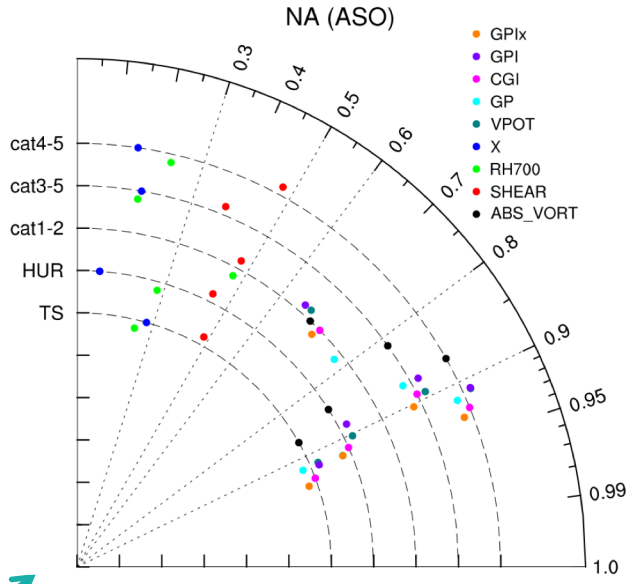
NCL: Example Plots

Storm Track





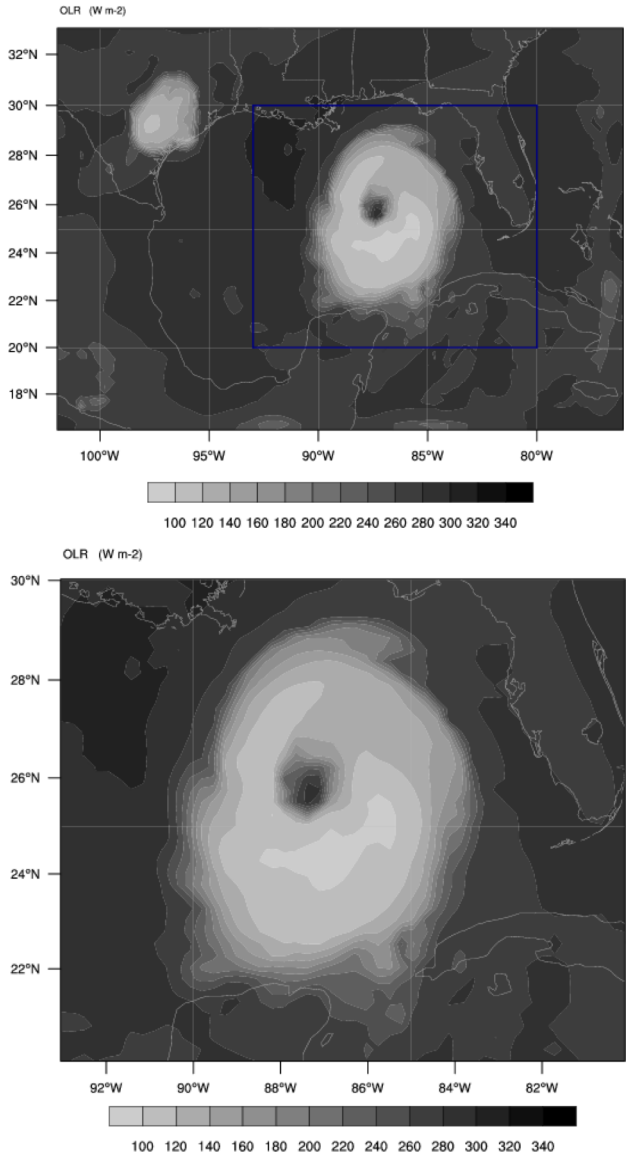
Bar Chart



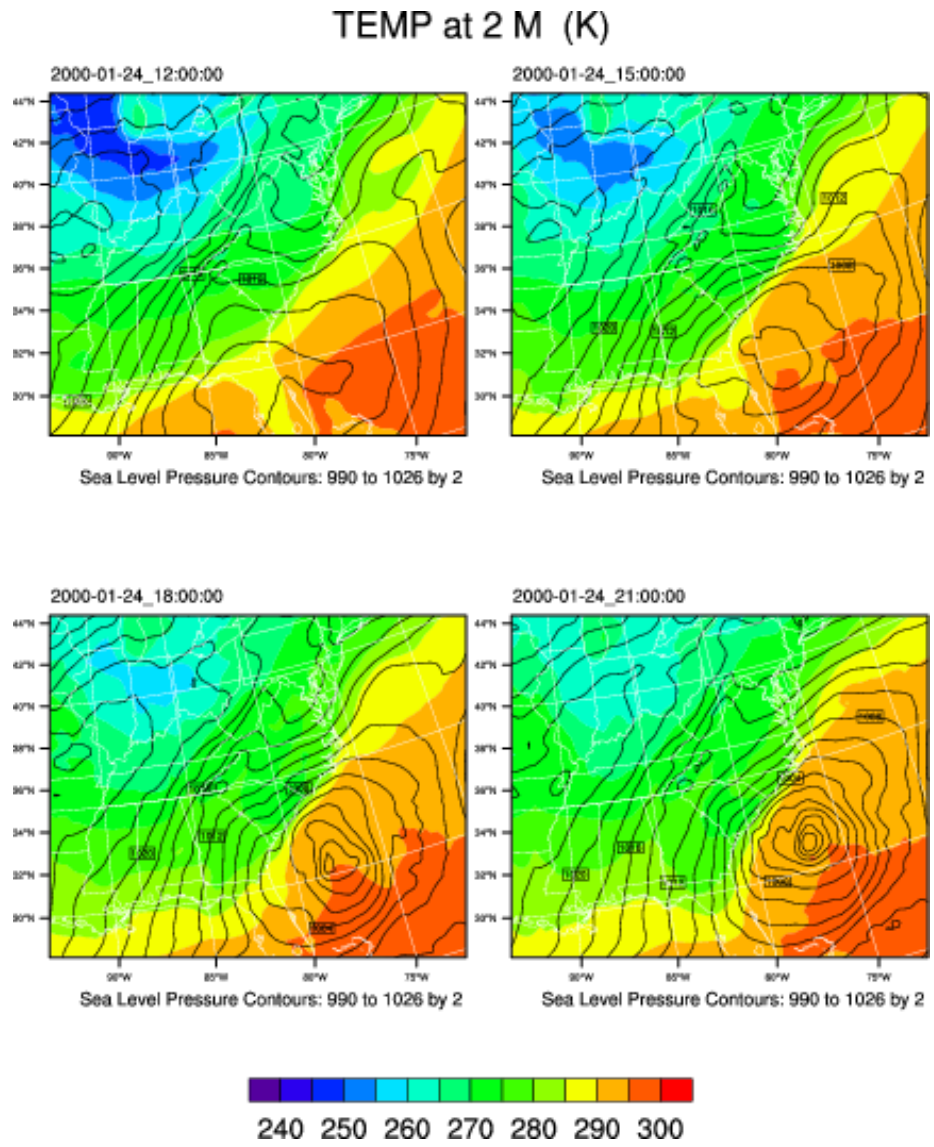
Taylor Diagram

NCL: Example Plot Functions

Zooming



Panel Plots



NCL: Downloading

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

- Fill out short registration form (short waiting period)
- Read and agree to OSI-based license
- Get version 6 or LATER (current: v6.6.2)

****Always download binary code instead of source code****

NCL: Libraries

- Scripts
  - User modifiable – can alter for your own purpose
  - WRFUserARW.ncl – contains most functions, diagnostics, etc. used by the sample script
    - Found in: $NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl
- Diagnostics
  - E.g., sea-level pressure
  - Written in Fortran and attached to code

NCL: Generating Plots

- Set NCARG_ROOT environment variable

  setenv NCARG_ROOT /usr/local/ncl ← **csh example**

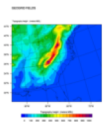
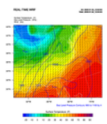
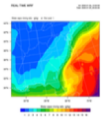
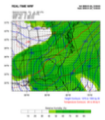
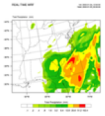
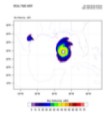
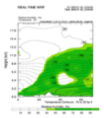
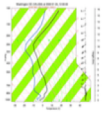
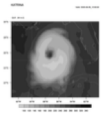
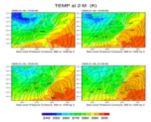
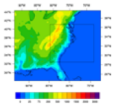
  - Can also set in .cshrc file

- Create a script
  - Start with a sample script (e.g., wrf_real.ncl)
  - Most of the WRF script routines are called from **WRFUserARW.ncl**

- Run NCL script

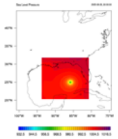
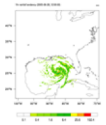
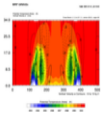
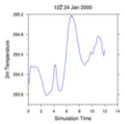
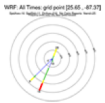
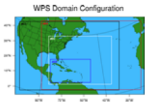
  ncl *your_script.ncl*

NCL: Generating Plots

Basic Plots	Basic Surface Plots	Plots on Model Levels	Plots on Interpolated Levels
 Basic Plot Setup *(This series of examples takes users though same basic steps in generating plotting scripts.)* Get and plot a single field Multiple input files Plot all fields in a file	 Surface 1 Surface 3 Surface 2 Surface with multiple input files	 Clouds Levels from wrfout files Levels from metgrid files	 Height Levels Pressure Levels
Plotting Precipitation	**Diagnostics**	**Cross-section Plots**	**Skew_T Plots**
 Precipitation	 CAPE helicity dBZ updraft_helicity PW Vorticity *(More diagnostics are available, shown are only some newer/special diagnostics)*	 Height - Through a Pivot Point Height - Point A to Point B Pressure Limited Vertical Extent For 2D fields Smooth terrain	 Skew_T
Overlay and Zoom	**Panel Plots**	**Overlay Domains**	
 Overlay Zoom Overlay & Zoom	 **Panel 1** **Panel 2**	 **Overlay 2 domains**	

A good start: WRF Online Tutorial

http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/index.php

Moving Nest	Moving Nest	Global WRF	Idealized cases
 Moving Nest *Keeping a fixed background for moving nest domain*	 **Precip Tendencies for a Moving Nest** ***Keeping a fixed background for moving nest domain, and plotting rainfall tendencies in the overlapping regions*** **Domain 1 and 2 Precip Tendencies on a single plot** **(domain 2 is a moving nest)**	 **gWRF_merc**	 **wrf_Grav2x** **wrf_Hill2d** **wrf_Squall_2d_x** **wrf_Squall_2d_y** **wrf_Seabreeze2x** **wrf_BWave** **wrf_QSS**
Time Slice Plots	**Track Cyclone**	**Wind Roses**	**Preview Domain**
 Meteograms WRF Time Series data	 Plot Cyclone Vortex	 **Create a wind rose**	 **Preview** ***Note: This example script makes use to the WPS namelist directly***

Creating a Plot: NCL script

```
load ncl library scripts


begin

  ; Open input file(s)
  ; Open graphical output

  ; Read variables

  ; Set up plot resources & Create plots
  ; Output graphics

end
```

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 input file(s)
  ; Open graphical output

  ; Read variables

  ; Set up plot resources & Create plots
  ; Output graphics

end
```

```
Load not required for NCL version 6.3+
    load "/mydir/myWRFUserARW.ncl"
```

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 input file(s)
  ; Open graphical output

  ; Read variables

  ; Set up plot resources & Create plots
  ; Output graphics

end
```

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

 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")
 ; Open graphical output

 ; Read variables

 ; Set up plot resources & Create plots
 ; Output graphics

end
```

```
Can be "r" = read
       "w" = write
       "c" = create
```

```
> ls wrfout*
  wrfout_d01_2005-10-08_00:00:00

a = addfile("./wrfout_d01_2005-10-
08_00:00:00.nc","r")
```

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

 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")
 wks = gsn_open_wks("X11","my_plot")

 ; Read variables

 ; Set up plot resources & Create plots
 ; Output graphics

end
```

Can output either on the screen (X11), or as pdf, eps, ps, png, cgm

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

 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")
 wks = gsn_open_wks("X11","plt_Surface")

 T2 = wrf_user_getvar(a,"T2",0)

 ; Set up plot resources & Create plots
 ; Output graphics

end
```

```
T2 = wrf_user_getvar(a,"T2",0)
T2 = a->T2(0,:,:)

T2 = wrf_user_getvar(a,"T2",-1)
T2 = a->T2
```

Special WRF NCL Functions

wrf_user_getvar : Function that grabs the fields from the input file

```
ter = wrf_user_getvar(a,"HGT",0)
t2  = wrf_user_getvar(a,"T2",-1)
slp = wrf_user_getvar(a,"slp",1)
```

```
{ter=a->HGT(0,:,:)}
{t2=a->T2}
```

avo/pvo: Absolute/Potential Vorticity
eth: Equivalent Potential Temperature
cape_2d: 2D mcape/mcin/lcl/lfc
cape_3d: 3D cape/cin
ctt: cloud top temperature
dbz/mdbz: Reflectivity (3D and max)
geopt/geopotential: Geopotential
lat/lon: latitude/longitude
helicity/updraft_helicity: Storm Relative Helicity/Updraft helicity
omg: Omega
p/pres/pressure: Pressure
pw: Precipitable Water
rh/rh2: Relative Humidity (3D and 2m)
slp: Sea Level Pressure
times: Time as a string [*(Times: Time as characters)*]
td/td2: Dew Point Temperature (3D and 2m)
ter: terrain
tc/tk: Temperature (C and F)
th/theta: Potential Temperature
tv: Virtual Temperature
twb: Wetbulb Temperature
z/height: Height
ua/va/wa: wind on mass points
uvmet/uvmet10: wind rotated to earth coordinates (3D and 10m)

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

 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")
 wks = gsn_open_wks("X11","plt_Surface")

 T2 = wrf_user_getvar(a,"T2",0)

 ; Set up plot resources & Create plots
 ; Output graphics

end
```

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

 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")
 wks = gsn_open_wks("X11","plt_Surface")

 T2 = wrf_user_getvar(a,"T2",0)

 pltres = True
 mpres = True
 opts = True
 opts@cnFillOn = True
 ; Output graphics

end
```

pltres: Plotting resources – like overlays
mpres: Map resources – like map resolution and zooming option

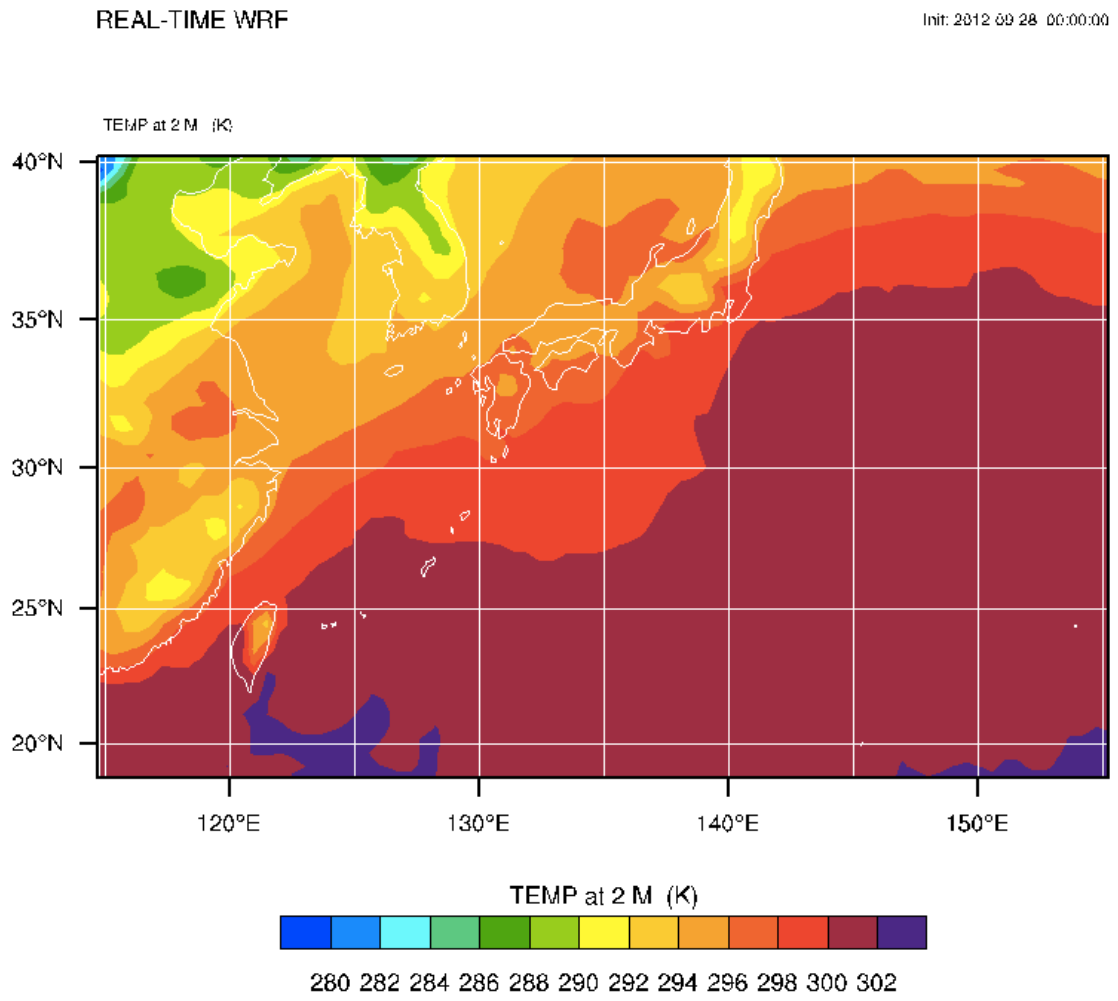
opts: Resources associated with each individual plot

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

 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")
 wks = gsn_open_wks("X11","plt_Surface")

 T2 = wrf_user_getvar(a,"T2",0)

 pltres = True
 mpres = True
 opts = True
 opts@cnFillOn = True
 contour_t2 = wrf_contour(a,wks,T2,opts)
 plot= wrf_map_overlays(a,wks,(/contour_t2/),pltres,mpres)

end
```

Creating

```
load "$NCARG_ROOT/lib/                                    "
load "$NCARG_ROOT/lib/                                    cl"

begin

 a = addfile("./wrfout
 wks = gsn_open_wks("X

 T2 = wrf_user_getvar(

 pltres = True
 mpres = True
 opts = True
 opts@cnFillOn = True
 contour_t2 = wrf_contour(a,wks,T2,opts)
 plot= wrf_map_overlays(a,wks,(/contour_t2/),pltres,mpres)

end
```

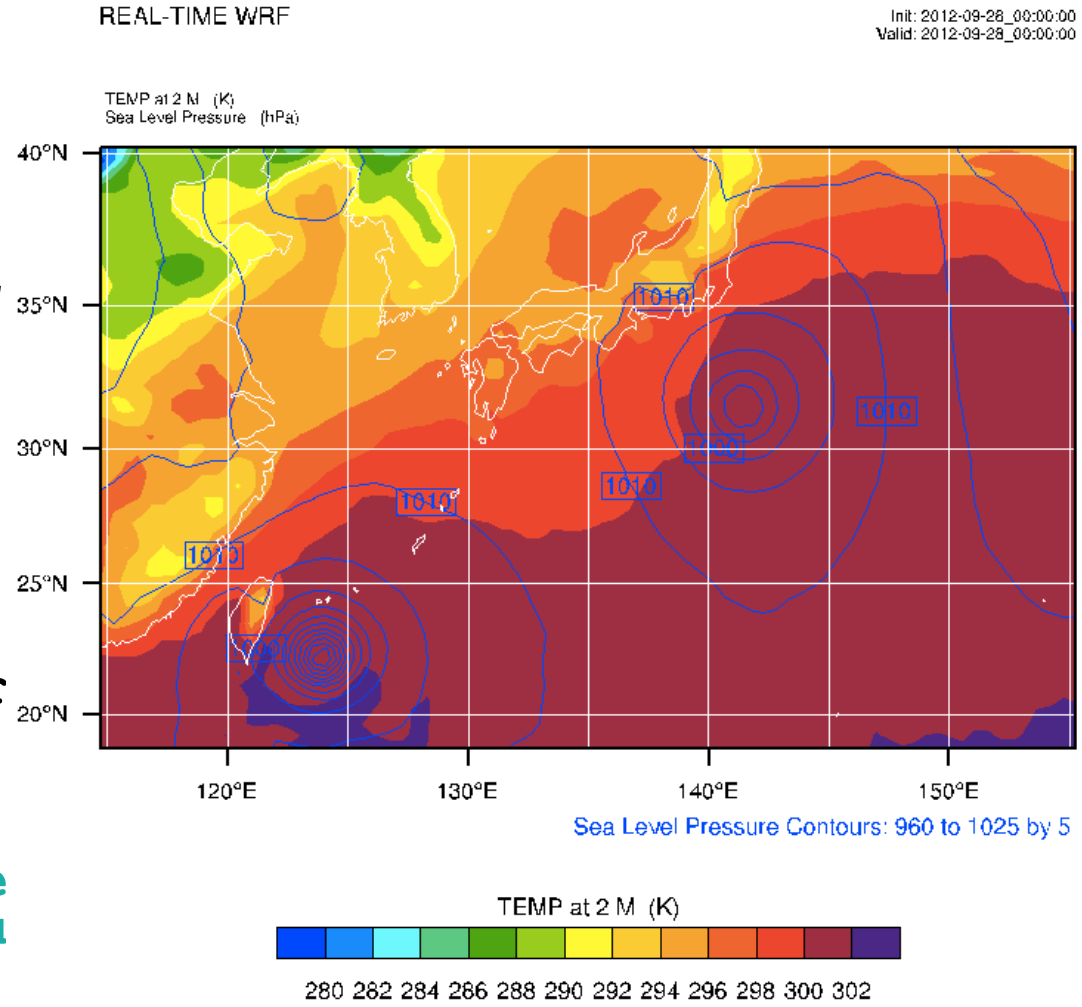


Creating a Plot: NCL script

```
  T2 = wrf_user_getvar(a,"T2",0)
  slp = wrf_user_getvar(a,"slp",0)

  pltres = True
  mpres = True

  opts = True
  opts@cnFillOn = True
  contour_t2 = wrf_contour(a,wks,T2,opts)
  delete(opts)

  opts = True
  opts@cnLineColor = "Blue"
  contour_slp = wrf_contour(a,wks,slp,opts)
  delete(opts)

  plot = wrf_map_overlays(a,wks,(/contour_t2,contour_slp/),
  pltres,mpres)


end
```

Creating a

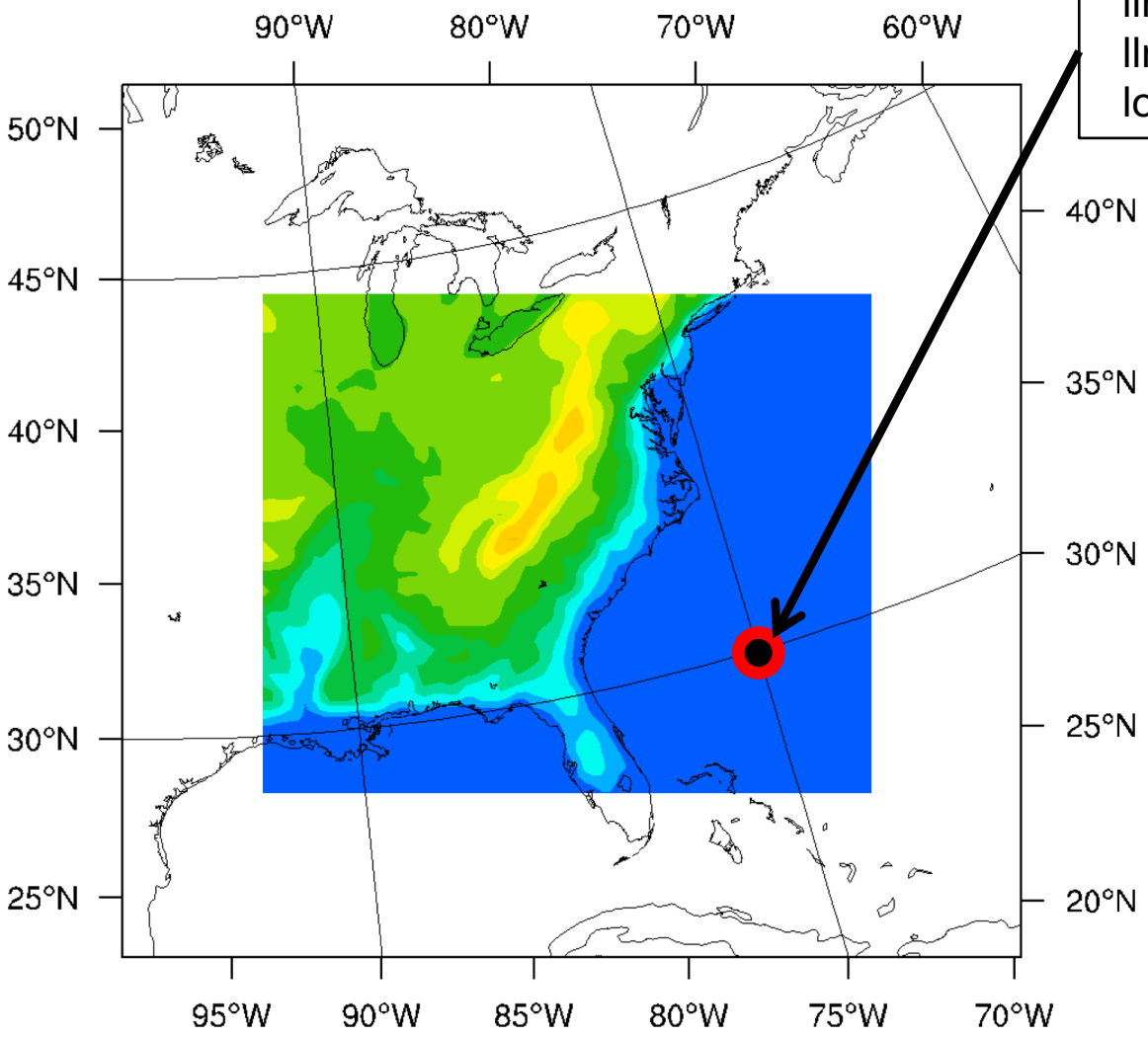
```
T2 = wrf_user_getvar(a,"
slp = wrf_user_getvar(a,

pltres = True
mpres = True

opts = True
opts@cnFillOn = True
contour_t2 = wrf_contour
delete(opts)

opts = True
opts@cnLineColor = "Blue
contour_slp = wrf_contou
delete(opts)

plot = wrf_map_overlays(a,wks,(/contour_t2,contour_slp/),
pltres,mpres)

end
```



NCL: Special WRF Functions

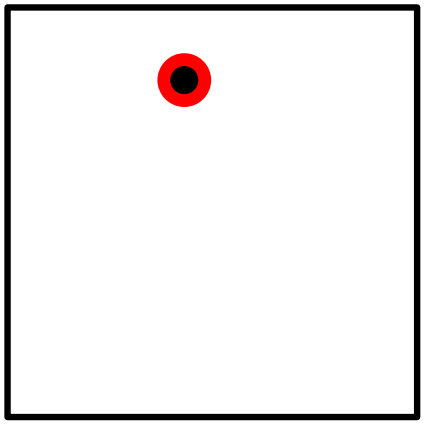
- **wrf_user_getvar**
  - *Get native and diagnostic variables*
- **wrf_contour / wrf_vector**
  - *Create line/shaded & vector plots*
- **wrf_map_overlays / wrf_overlays**
  - *Overlay plots created with wrf_contour and wrf_vector*
- **wrf_wps_read_int / wrf_wps_write_int**
  - *For reading/writing in intermediate format*

NCL: wrf_user_ll_to_ij

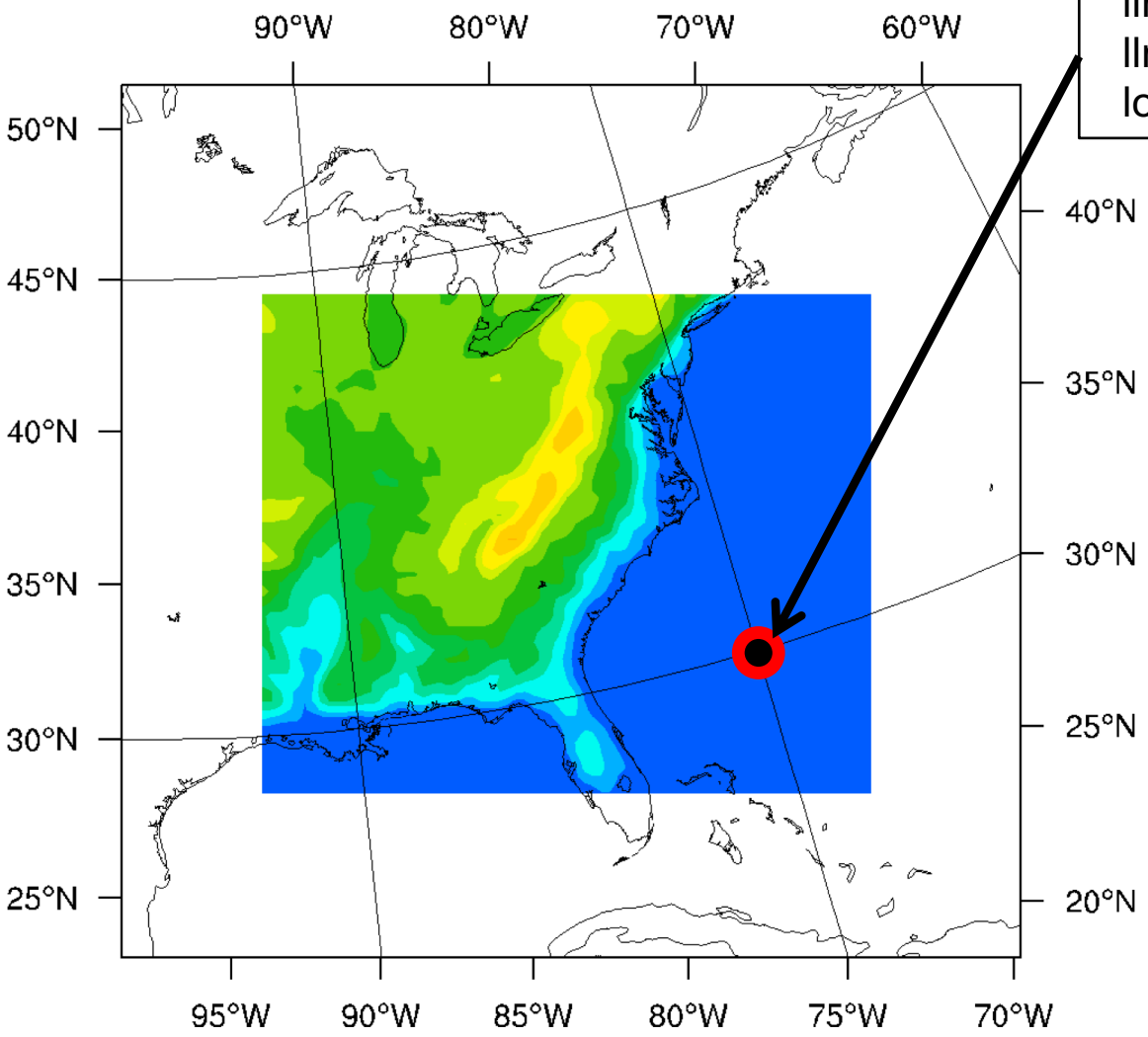



```
llres = True
llres@returnInt = False
locij = wrf_user_ll_to_ij(a, -75.0, 30.0, llres)
```

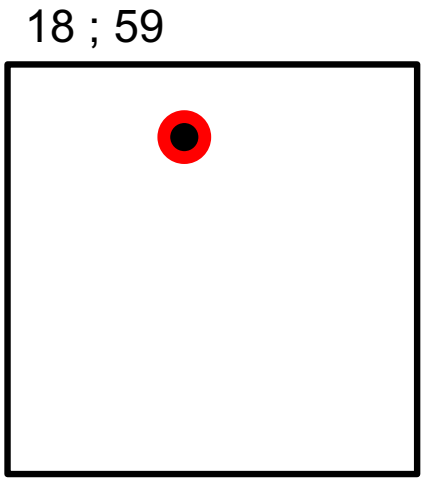
17.95 ; 59.48



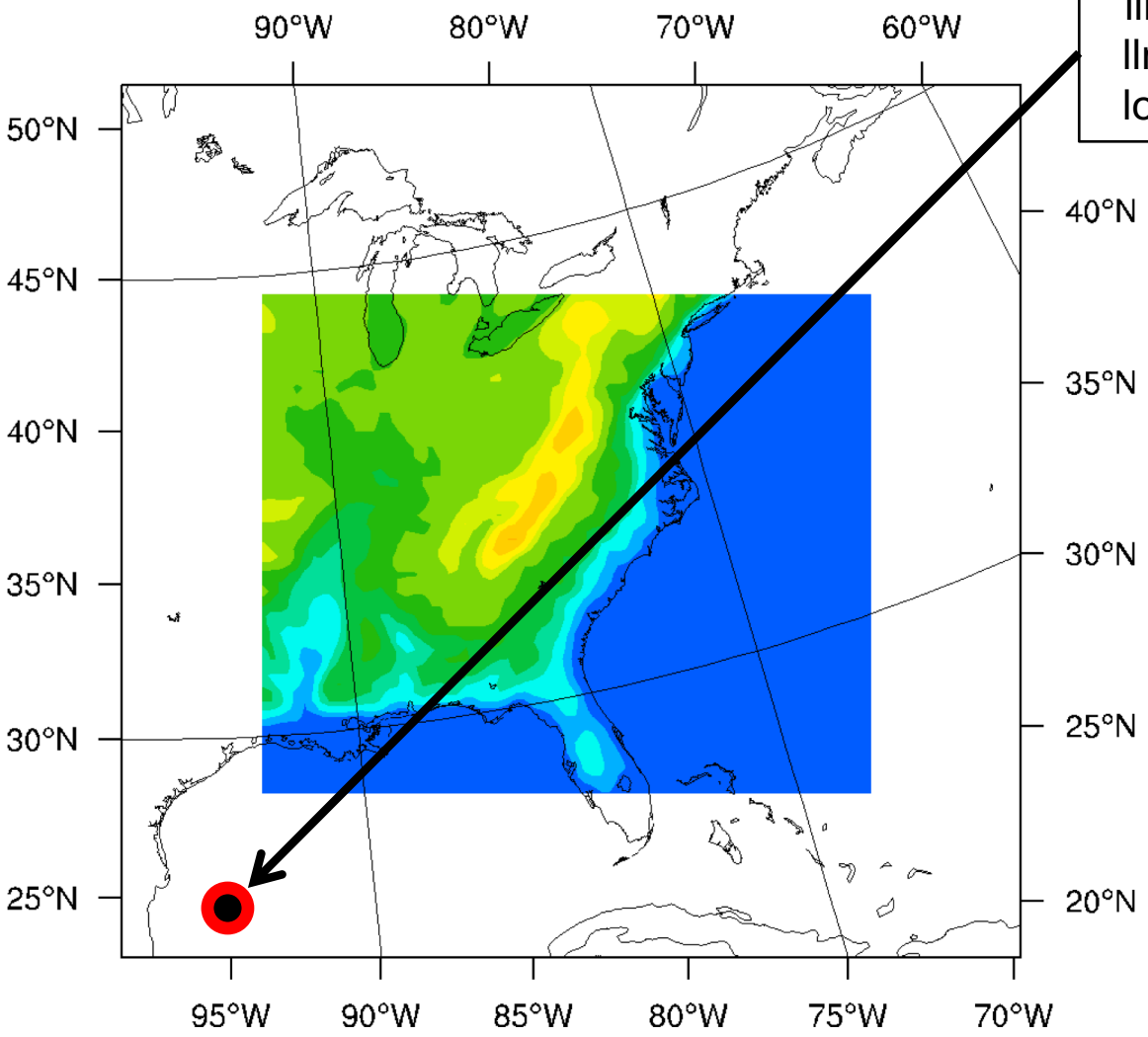
NCL: wrf_user_ll_to_ij




```
llres = True
llres@returnInt = True
locij = wrf_user_ll_to_ij(a, -75.0, 30.0, llres)
```



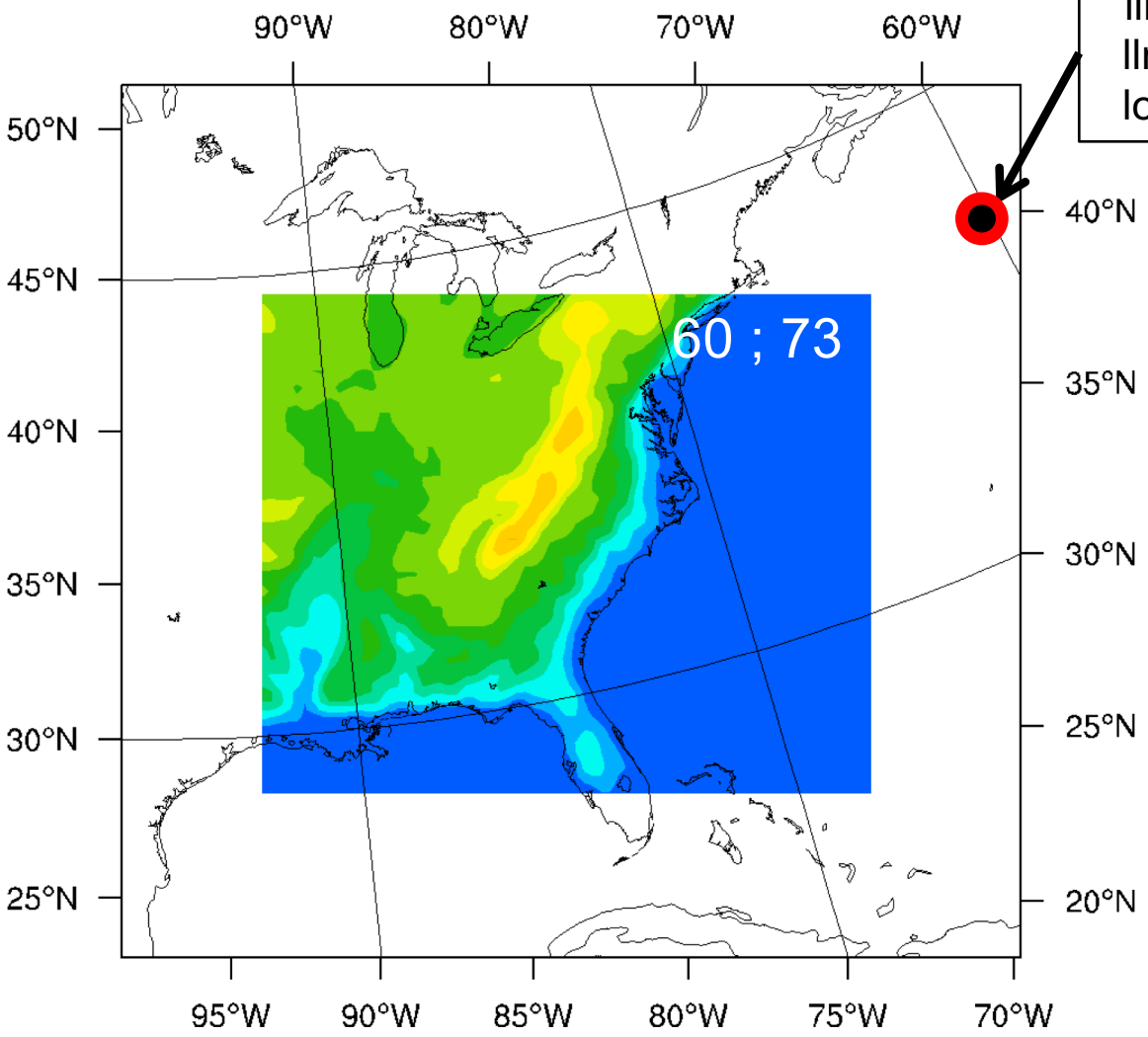
NCL: wrf_user_ll_to_ij




```
llres = True
llres@returnInt = True
locij = wrf_user_ll_to_ij(a, -95.0, 25.0, llres)
```

-10 ; -2

NCL: wrf_user_ll_to_ij




```
llres = True
llres@returnInt = True
locij = wrf_user_ll_to_ij(a, -95.0, 25.0, llres)
```

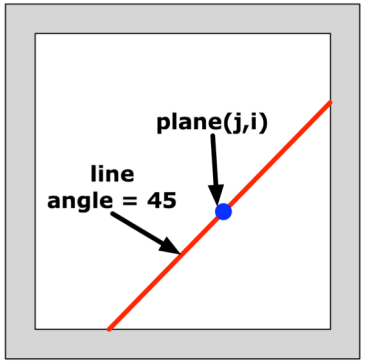
68; 87

Can do this in reverse

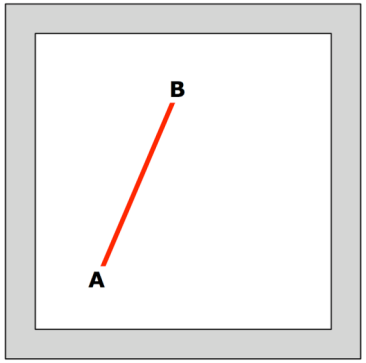
wrf_user_ij_to_ll

NCL: wrf_user_intrp3d/ wrf_user_intrp2d

- *Interpolates horizontally to a given pressure/height*
- *3d data only – interpolates vertically along a given line*

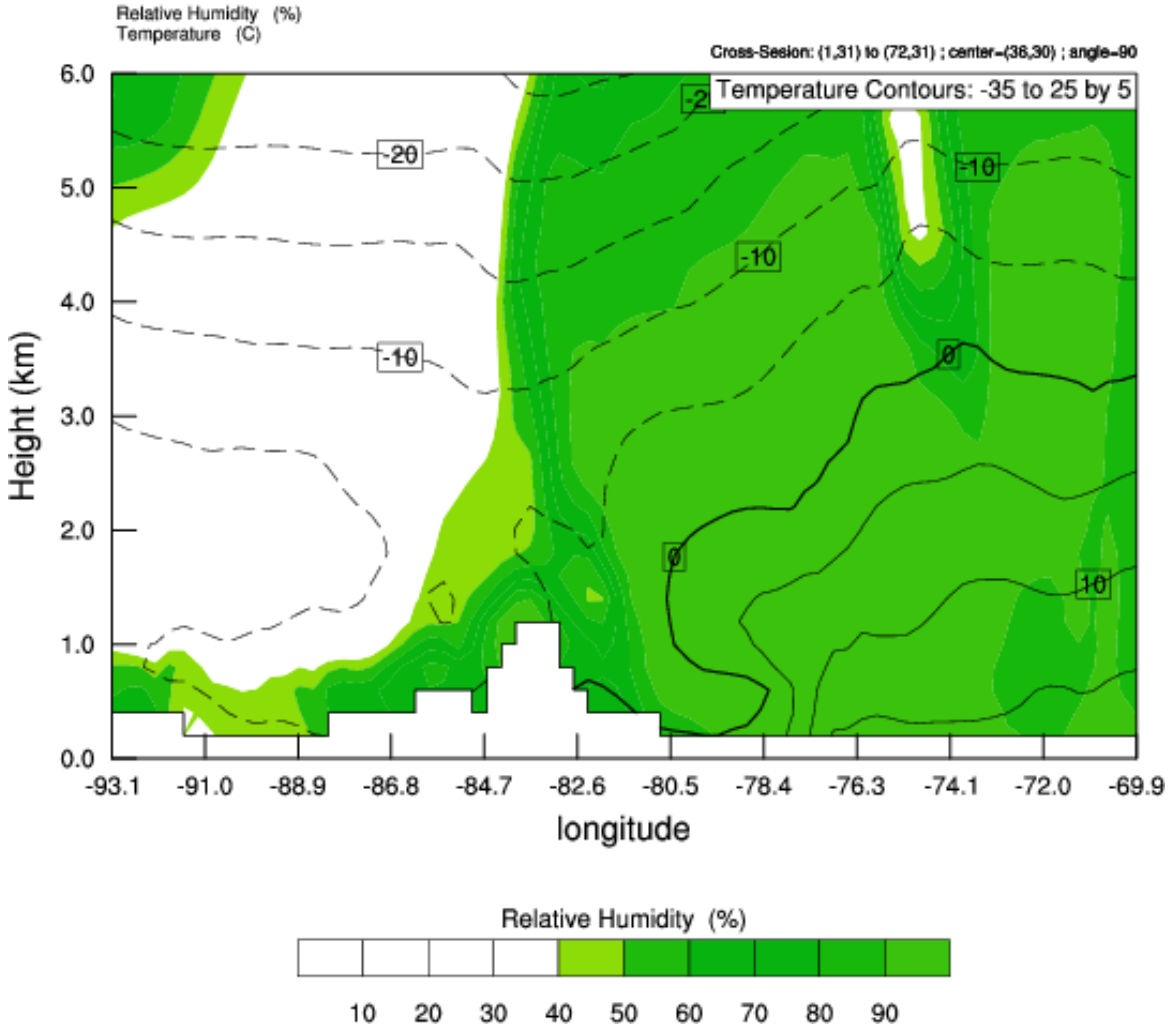


If we give a point and angle, the interpolation will be Done along the line.



Can also specify 2 points: interpolation will be done Along the line, between the points

NCL: wrf_user_intrp3d/ wrf_user_intrp2d



NCL: wrf_user_vert_interp

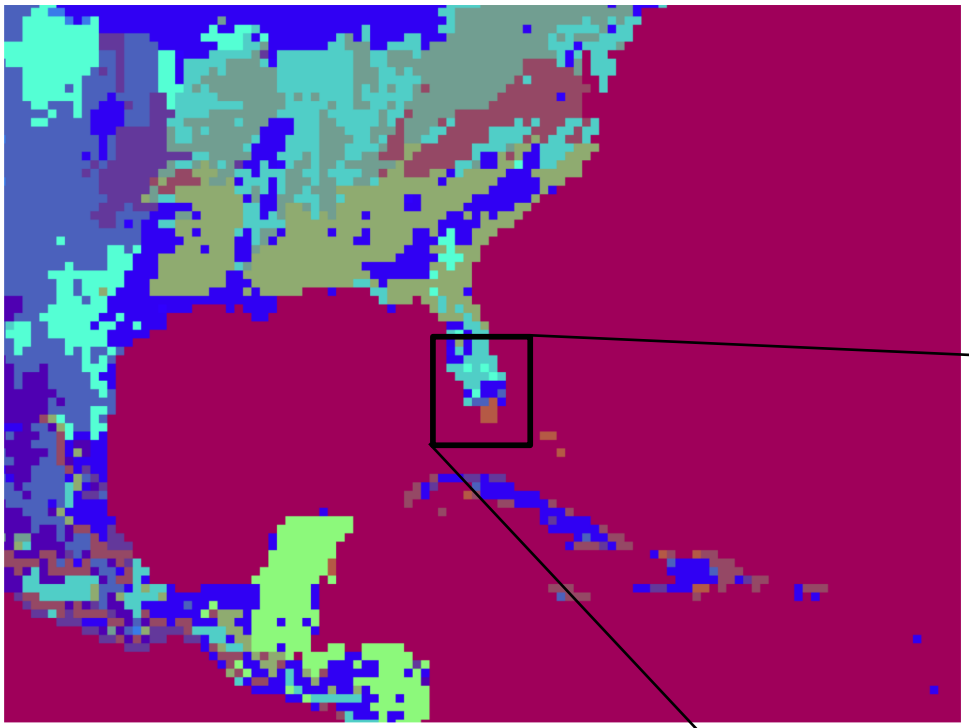
- Interpolates to:
  - “pressure”,”pres” – pressure [hPa]
  - “ght_msl” – grid point height msl [km]
  - “ght_agl” – grid point height agl [km]
  - “theta” – potential temperature [K]
  - “theta-e” – equivalent potential temperature [K]

- Extrapolates below the ground
  - Resource: opts@extrapolate

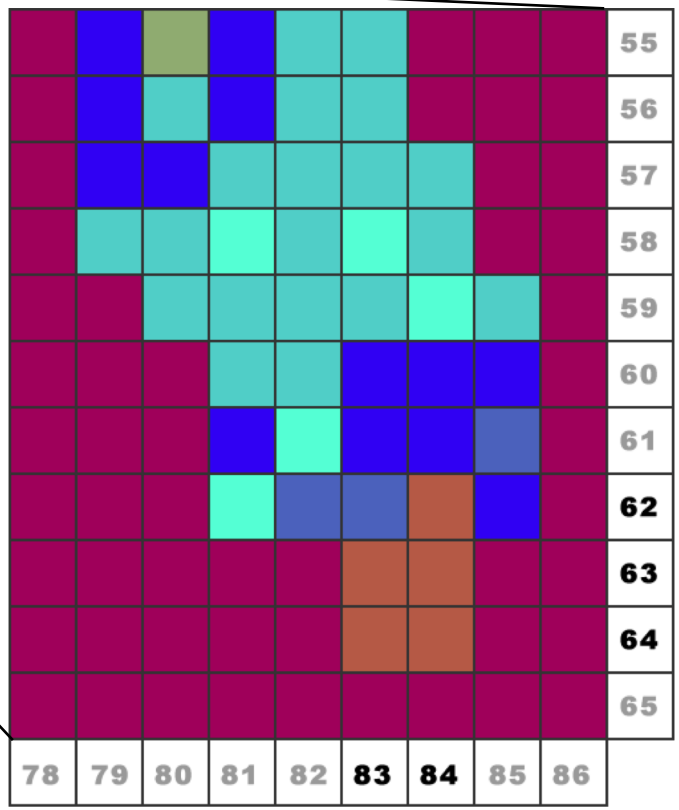
    E.g., opts = true

    opts@extrapolate = true

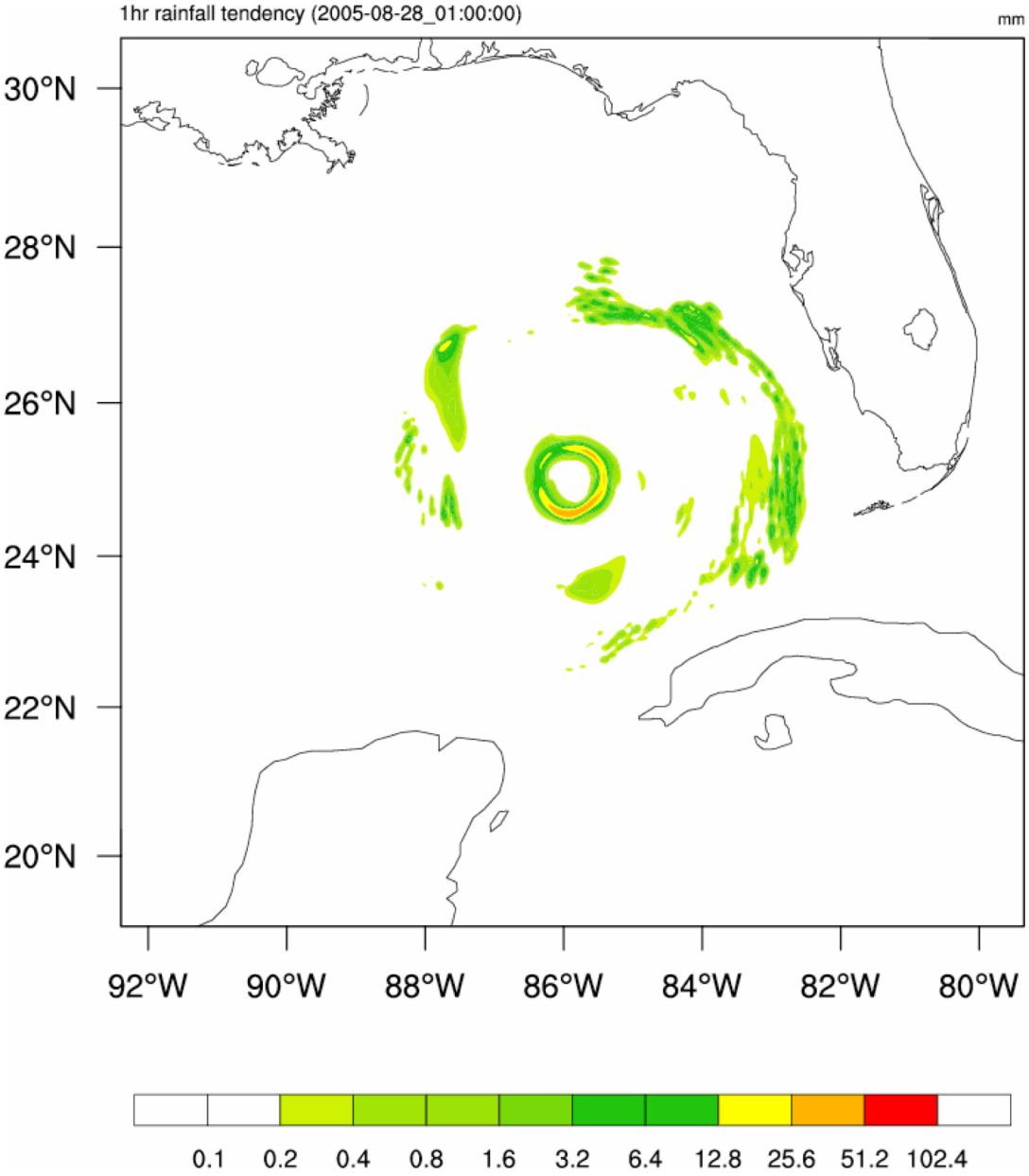
NCL: Change Fields in netCDF File



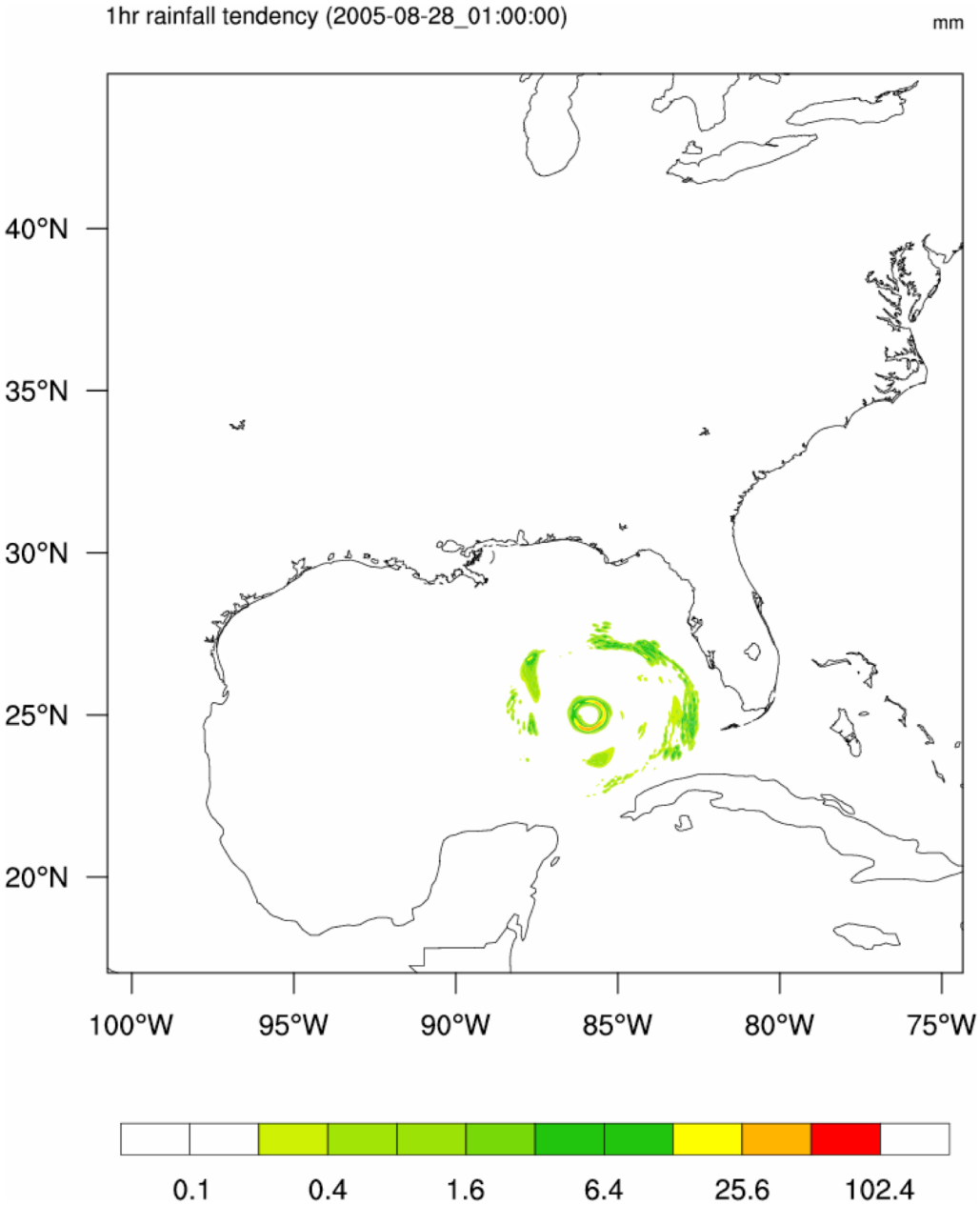
```
a = addfile("./geo_em.d01.nc","w")
var= a->LANDUSE
var(:,63:64,83:84) = 7
var(:,62,84) = 7
a->LANDUSE = var
```



NCL: Moving Nests



NCL: Moving Nests



NCL Resources

- The special WRF functions have unique resources:
  http://www.ncl.ucar.edu/Document/Functions/wrf.shtml

- All general NCL resources can also be used to control the plot:
  http://www.ncl.ucar.edu/Document/Graphics/Resources

am (anotation manager)
app (app)
ca (coordinate array)
cn (contour)
ct (coordinate array table)
dc (data comm)
err (error)
gs (graphic style)
gsn (gsn high-level interfaces)
lb (label bar)
lg (legends)
mp (maps)
pm (plot manager)
pr (primitives)
sf (scalar field)
st (streamline)
tf (transformation)
ti (title)
tm (tickmark)
tf (irregular transformation)
tx (text)
vc (vectors)
vf (vector fields)
vp (view port)
wk (workstation)
ws (workspace)
xy (xy plots)

WRF Python

- A collection of diagnostic and interpolation routines for use with WRF-ARW
- Functionality is very similar to what is provided by the WRF NCL functions
- When coupled with either matplotlib or PyNGL you can create plots very similar to what you make with NCL

https://github.com/NCAR/wrf-python

NCL: Support

- WRF & MPAS-A Support Forum
  - For all WRF-related NCL questions
  - Look for NCL section in the Post-processing/Utilities section

https://forum.mmm.ucar.edu/

- For all generic NCL questions:

ncl-talk@ucar.edu

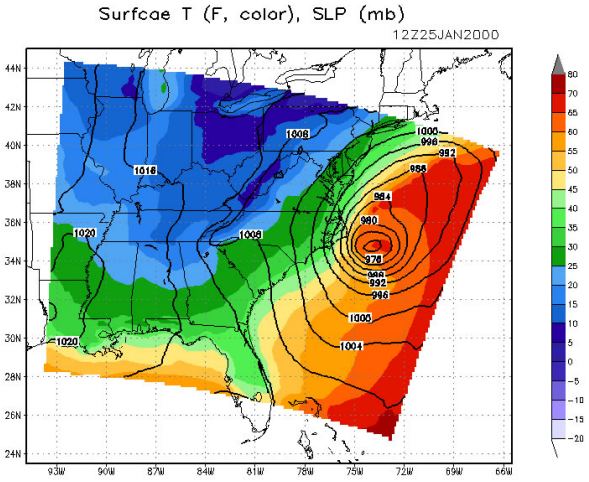
ARWpost

ARWpost: General Information

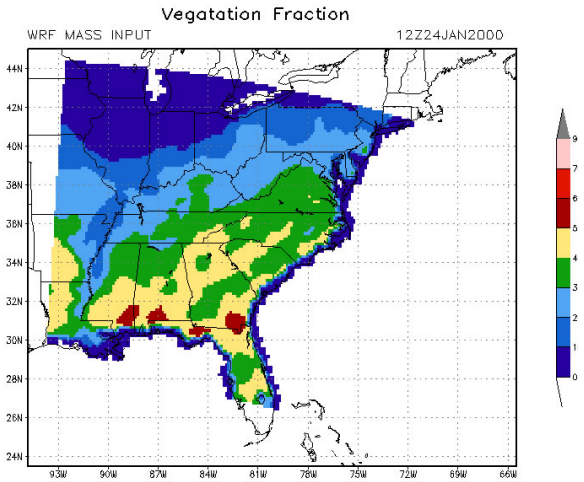
- Converter
  - Reads-in wrf-arw model data, creates GrADS output files
  - Requires GrADS to display
- GrADS software is only needed to display data, not needed to compile the code
  - http://www.iges.org/grads/grads.html
- Generate a number of graphical plots
  - Horizonal
  - Cross-section
  - skewT
  - Meteogram
  - Panel
- Download Code
  - http://www2.mmm.ucar.edu/wrf/users/download/get_sources.html
- Online Tutorial
  - http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/ARWpost/index.php

ARWpost: Example Plots

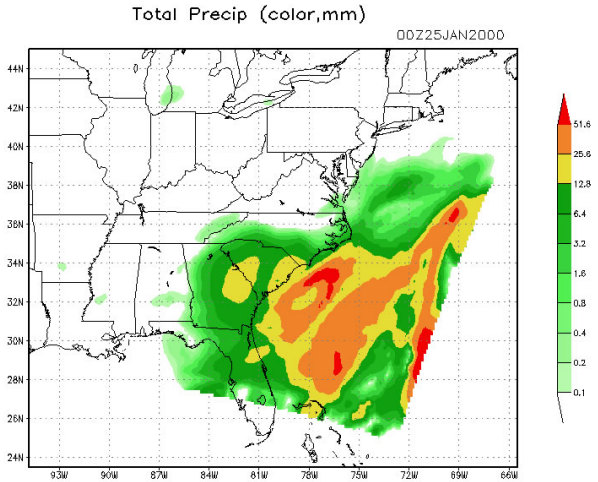
Surface Temp



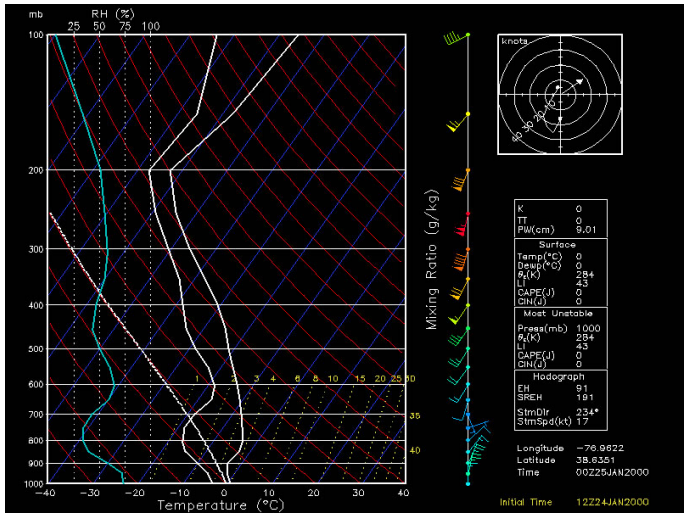
Vegetation Fraction



Total Precipitation

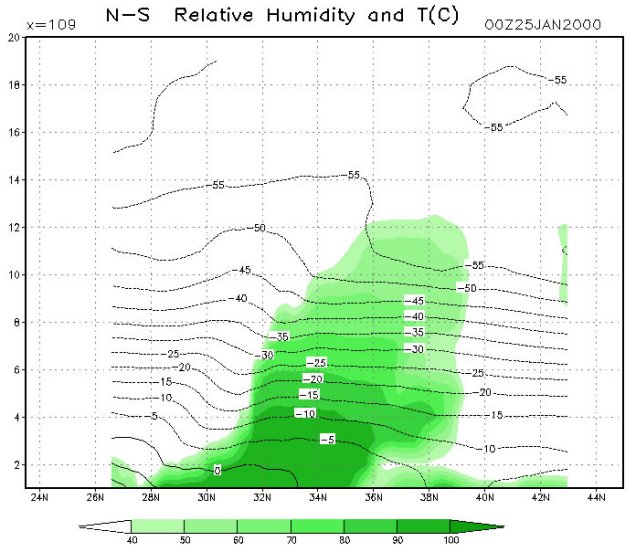


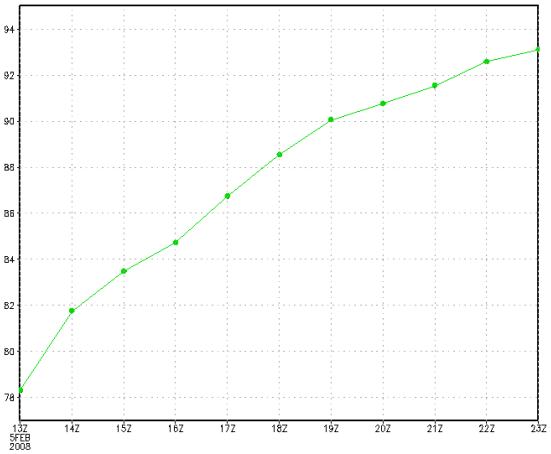
Skew-T Diagram



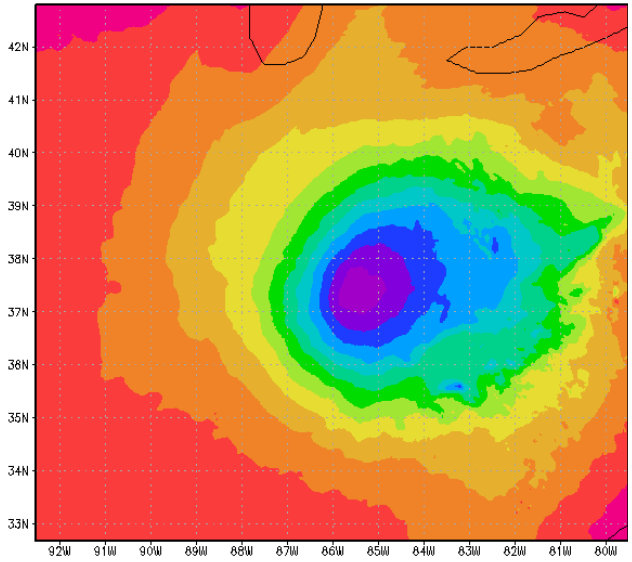
ARWpost: Example Functions

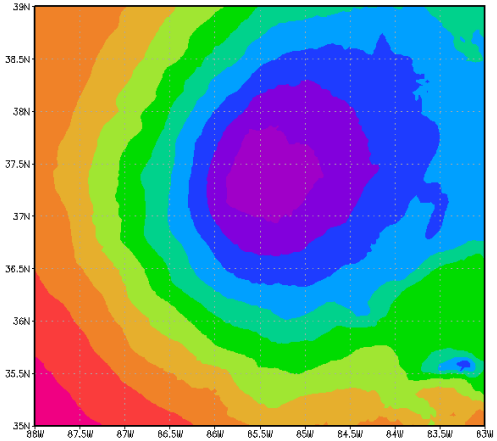
Cross-Sections





Zooming





ARWpost: Diagnostics

- cape – 3d cape
- cin – 3d cin
- mcape – maximum cape
- mcin – minimum cin
- clfr – low/middle/high cloud fraction
- dbz – 3d reflectivity
- max_dbz – maximum reflectivity
- geopt – geopotential
- height – model height in km
- lcl – lifting condensation level
- lfc – level of free convection
- pressure – full model pressure in hPa
- rh – relative humididy
- rh2 – 2 m relative humidity
- theta – potential temperature
- tc – temperature in degrees C
- tk – temperature in degrees K
- td – dew point temperature in degrees C
- td2 – 2m dew point temperature in degrees C
- slp – sea level pressure
- umet & vmet – winds rotated to Earth coordinates
- u10m & v10m – 10 m winds rotated to Earth coordinates
- wdir – wind direction
- wspd – wind speed coordinates
- wd10 – 10 m wind direction
- ws10 – 10 m wind speed

ARWpost: Scripts

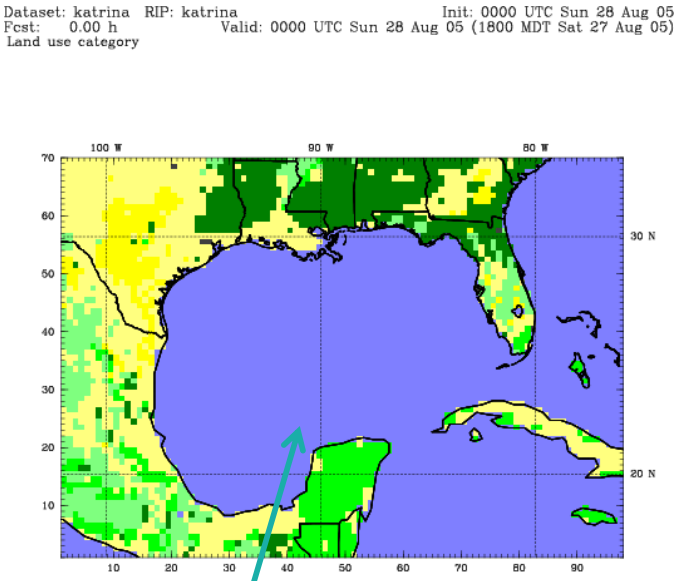
Script Name	Description
cbar.gs	Plots a color bar on shaded plots
rgbset.gs	Allows you to add/change colors from color # 20 – 99
skew.gs	Program to plot a skewT
plot_all.gs	Automatically finds all .ctl files in the directory and lists them so the user can pick when to use, will plot all fields chosen
rain.gs (real data only)	Plots total rainfall (must have data that contain fields RAINC and RAINNC)
cross_z.gs (real data only)	Plots a NS and EW cross section of RH and T (C)

RIP4

RIP4: General Information

- Requires NCAR Graphics Libraries
  - http://www.ncl.ucar.edu
- Source Code
  - http://www2.mmm.ucar.edu/wrf/users/download/get_source.html
- Documentation
  - Included in program's tar file (in Doc/ directory)
  - http://www2.mmm.ucar.edu/wrf/users/docs/ripug.htm
- Online Tutorial
  - http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/RIP4/index.php

RIP4: Example Plots

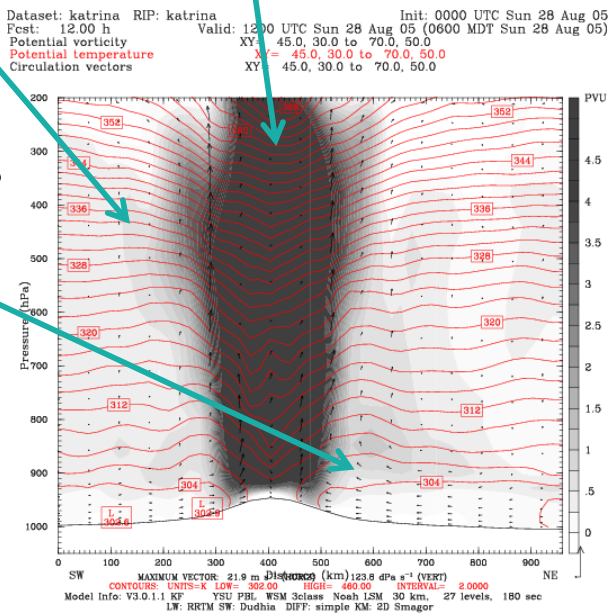


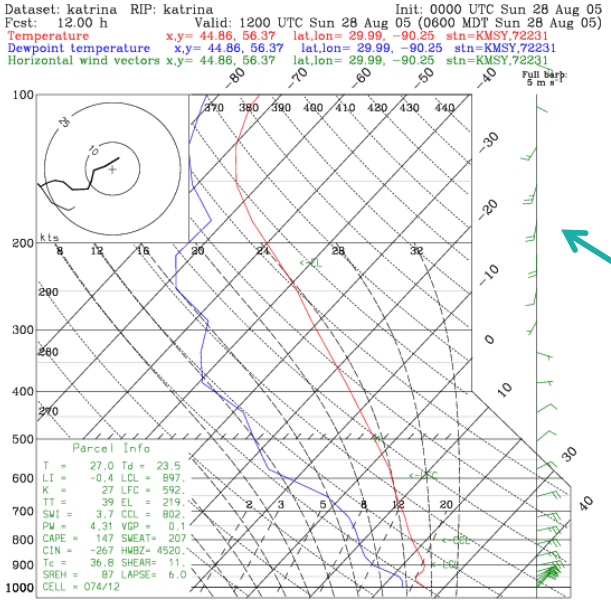
Landuse Category

Potential Temp

Potential Vorticity

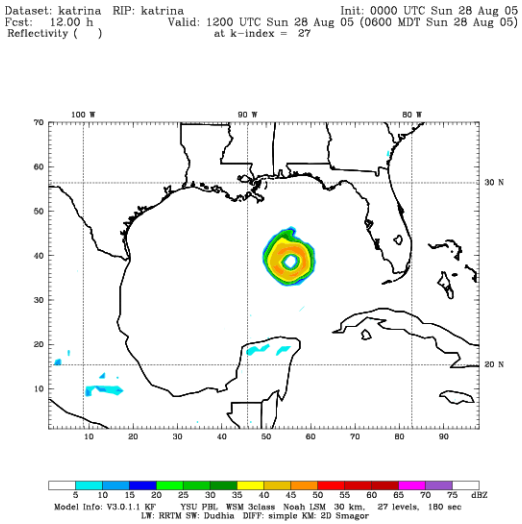
Circulation Vectors

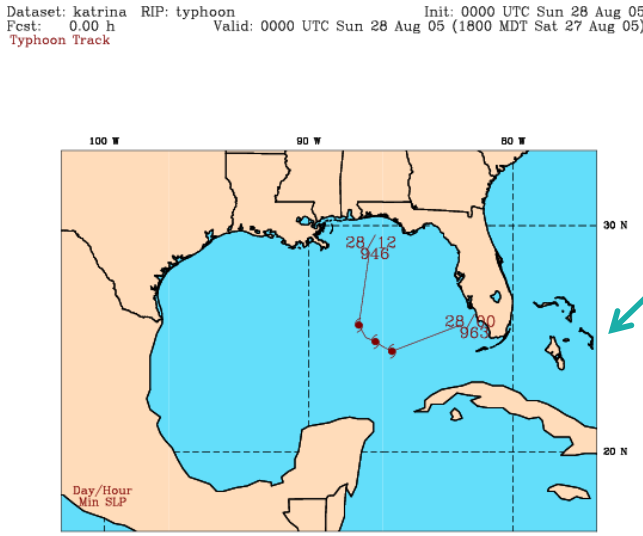




Skew-T Diagram

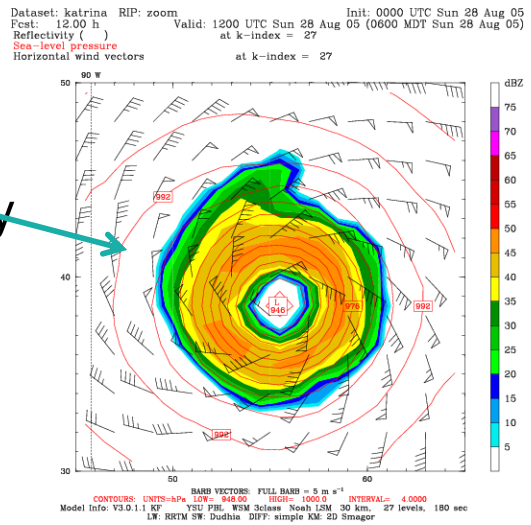
RIP4: Example Plots



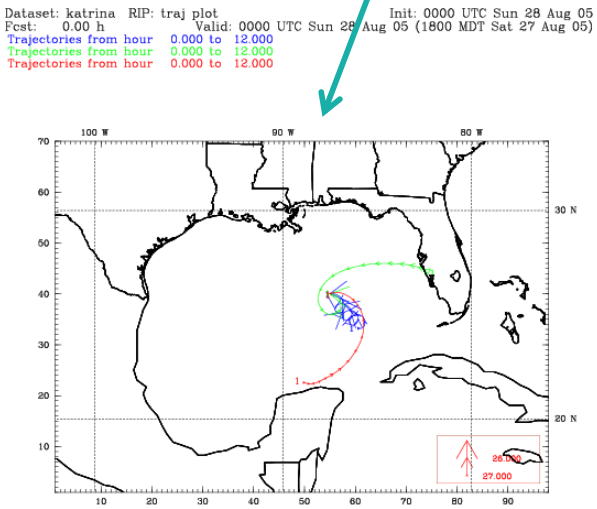


Cyclone Tracking

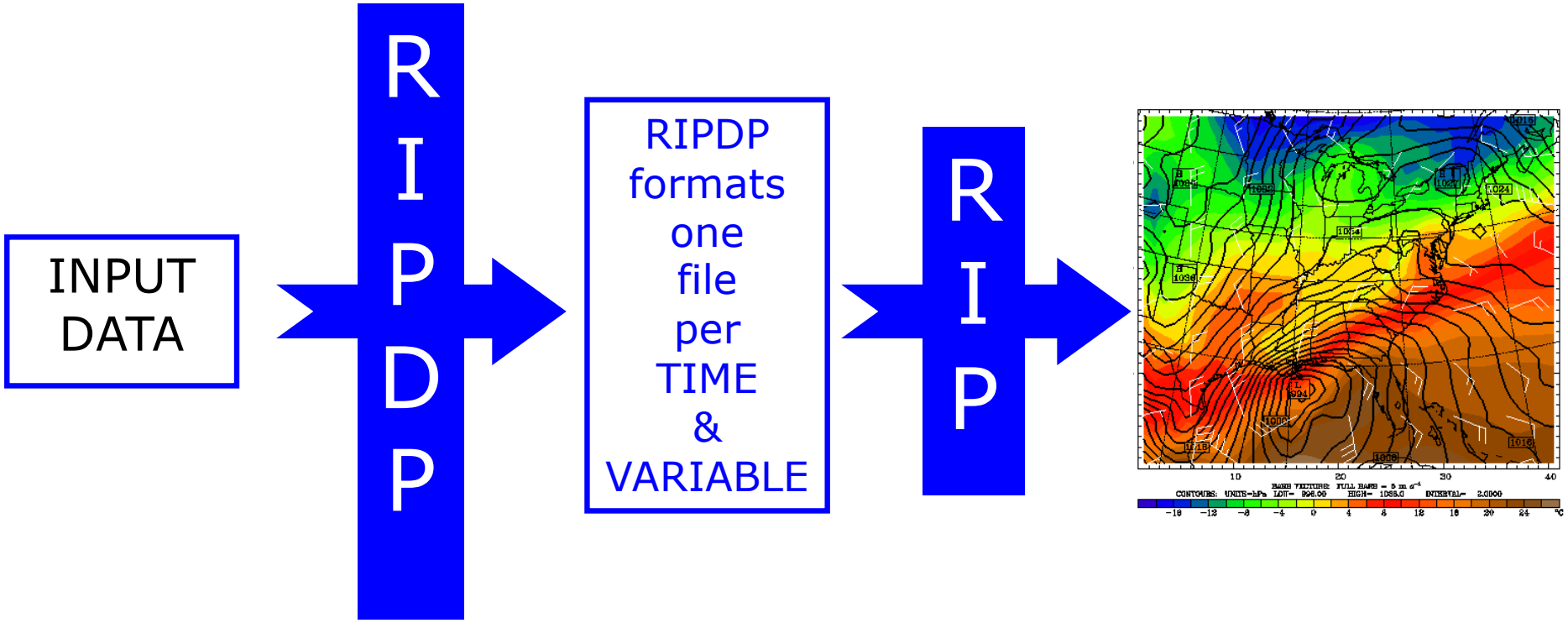
Zooming Capability



Trajectories



RIP4: Program Flow



RIP4: Namelist (*&userin*)

- Use namelist to control
  - processing times, intervals, title information, text quality on a plot
  - whether to do time series, trajectory, or to write output for Vis5D
  - *Full explanation for namelist variables is available in the user document*

- `ptimes, ptimeunits` – times to process
- `tacc` – tolerance for processing data
- `iusedaylightrule` – 1 applied, 0 not applied
- `idotser` – generate time series output
- `icgmsplit` – split metacode into several files
- `itrajcalc` – 0, 1 ONLY when doing trajectory calculations
- `rip_root` - override RIP_ROOT
- `ncarg_root` - output type: X11, cgm, pdf, ps

VAPOR

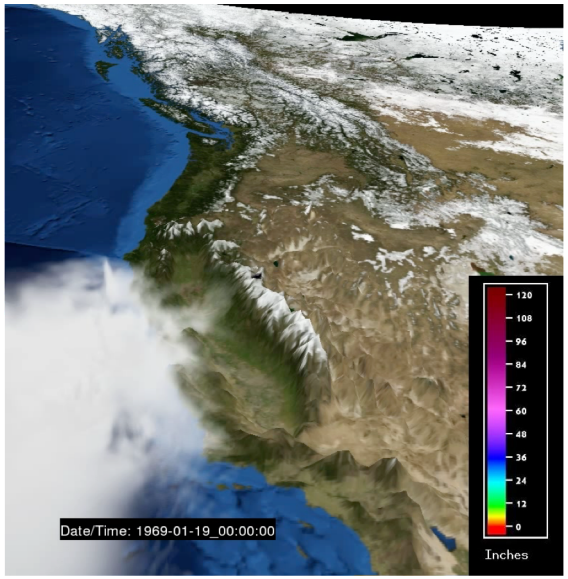
VAPOR: General Information

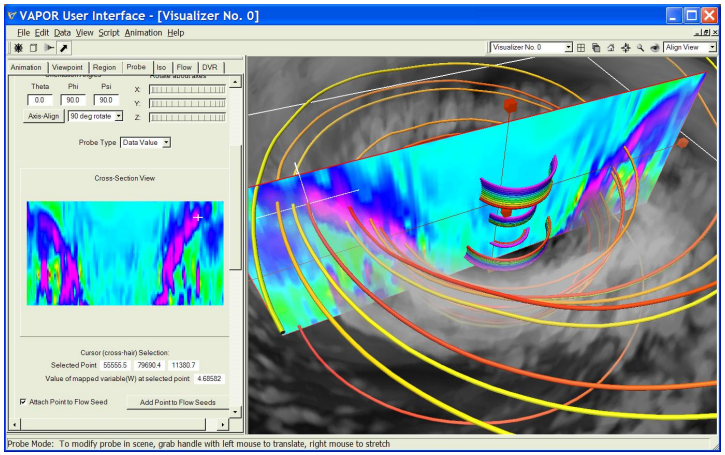
- Visualization and Analysis Platform for Oceanic, atmospheric, and solar Research

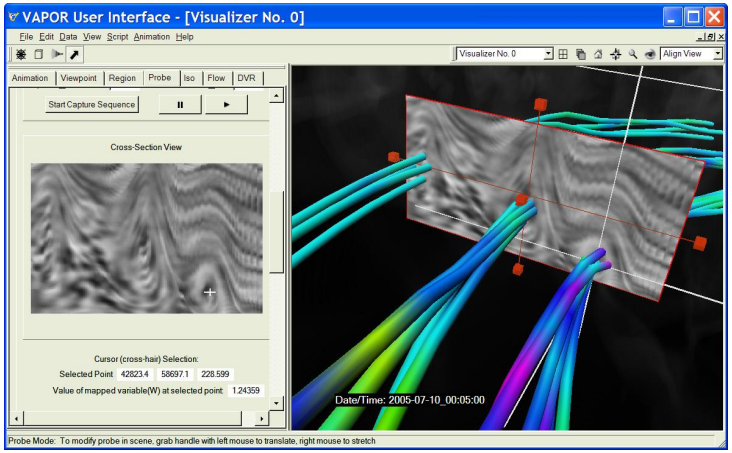
- 3D visualization tool

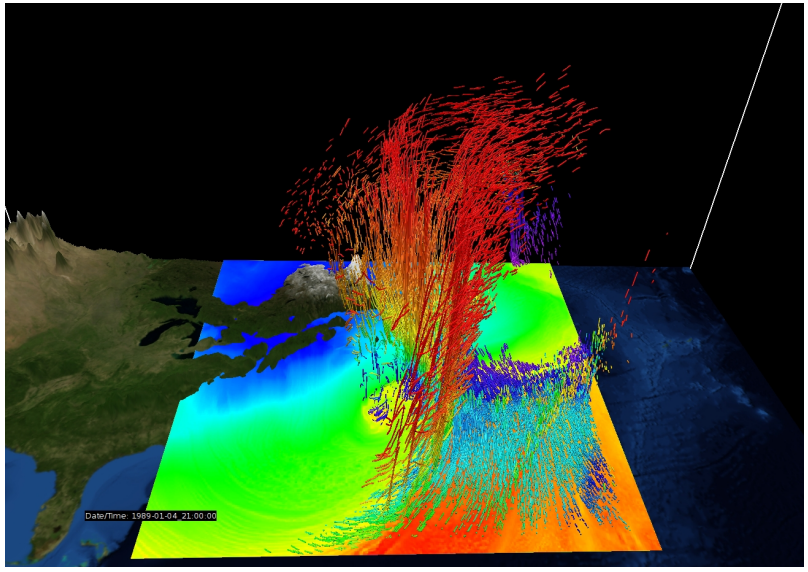
- Support:
  - Scott Pearse (vapor@ucar.edu)

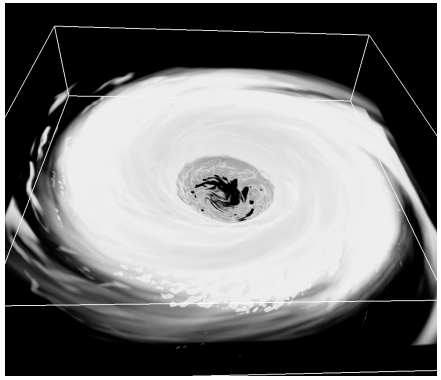
VAPOR: Example Plots











VAPOR Installation

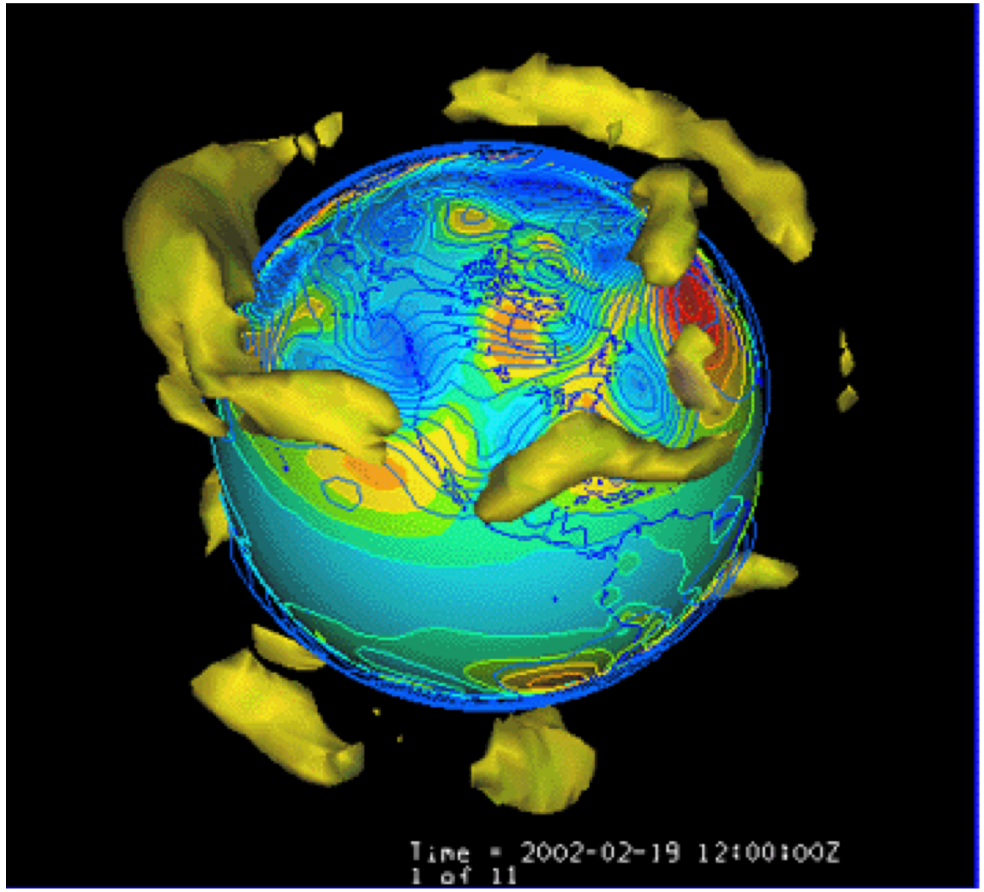
- Available for Linux, Windows, and Mac systems
- VAPOR website: http://www.vapor.ucar.edu
  - Download appropriate binary installer & follow installation instructions
  - *Note: You will need administrative privileges on a Mac
- Source vapor install.csh in your shell script before running any VAPOR commands (for Mac or Linux systems)

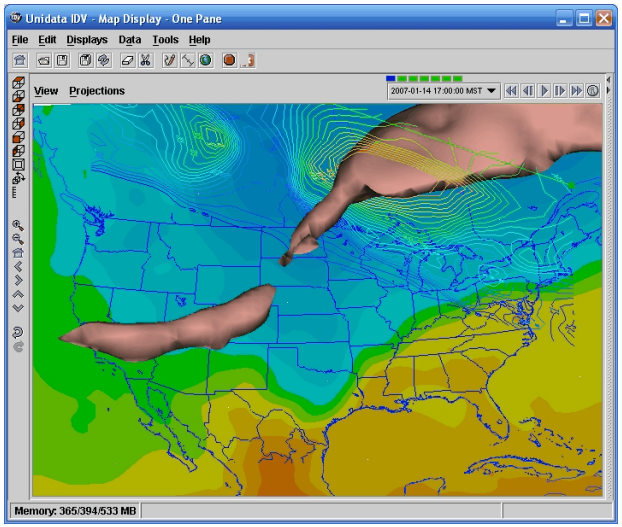

IDV

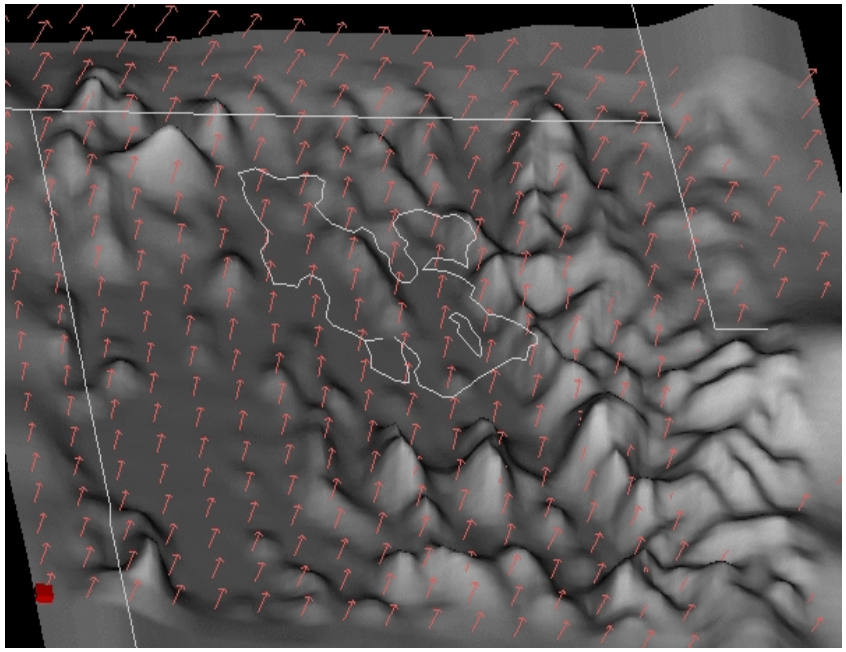
IDV: General Information

- Integrated Data Viewer
- 2D and 3D visualization tool
- Integrate model and observational data
- Visualize and analyze post-processed WRF output
- Has ability to read in a lot of different data (e.g., aircraft data), brings in as an overlay without having to change projections
- Uses a LOT of memory!
- Developed and supported by:
  - Yuan Ho and Julien Chastang (NCAR)
  - support@unidata.ucar.edu

IDV: Example Plots







IDV: Downloading

- https://www.unidata.ucar.edu/software/idv
  - Point-and-click installers
  - Windows (.exe), Mac (.dmg), and Linux (.sh) installers available for both 32 and 64-bit systems

- System requirements
  - 2+ GB of Ram
  - Java 1.6
  - *Latest* video card driver

Questions?