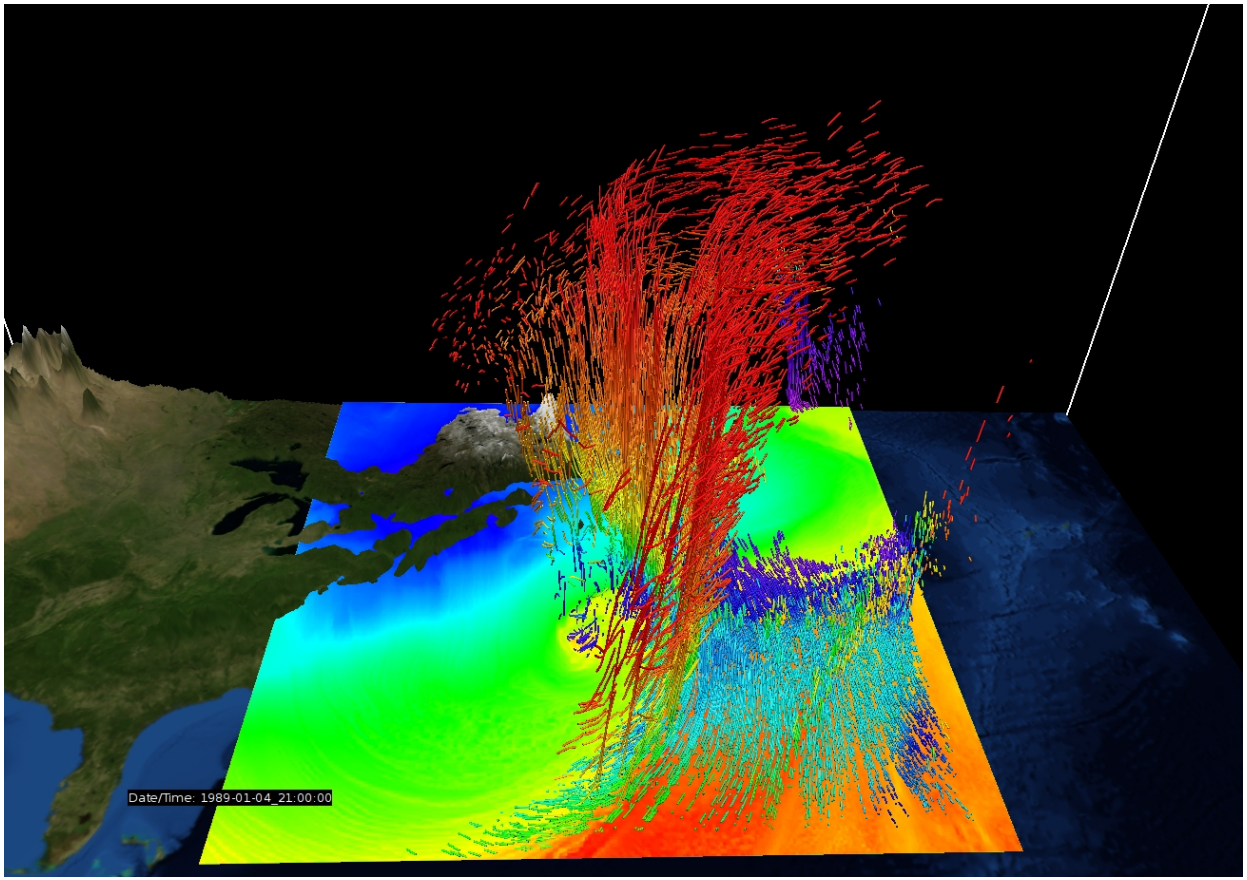
Post-processing Tools

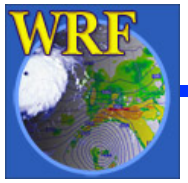
Cindy Bruyère



Graphical Packages

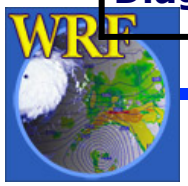
- **NCL** — UG: 9-2
  - Graphical package
- **ARWpost** — UG: 9-28
  - Converter (GrADS)
- **RIP4** — UG: 9-19
  - Converter and interface to graphical package NCAR Graphics
- **UPP** — UG: 9-35
  - Converter (GrADS & GEMPAK)
- **VAPOR** — UG: 9-58
  - Converter and graphical package
  - *Support: VAPOR*
- **IDV** — unidata.ucar.edu
  - GRIB (from UPP)
  - GEMPAK (from wrf2gem)
  - Vis5d
  - CF complaint data (from wrf_to_cf)
  - *Support: unidata*
- **GEMPAK**
  - Data from wrf2gem or UPP
  - *Support: unidata*

MatLab / IDL / R / ferret

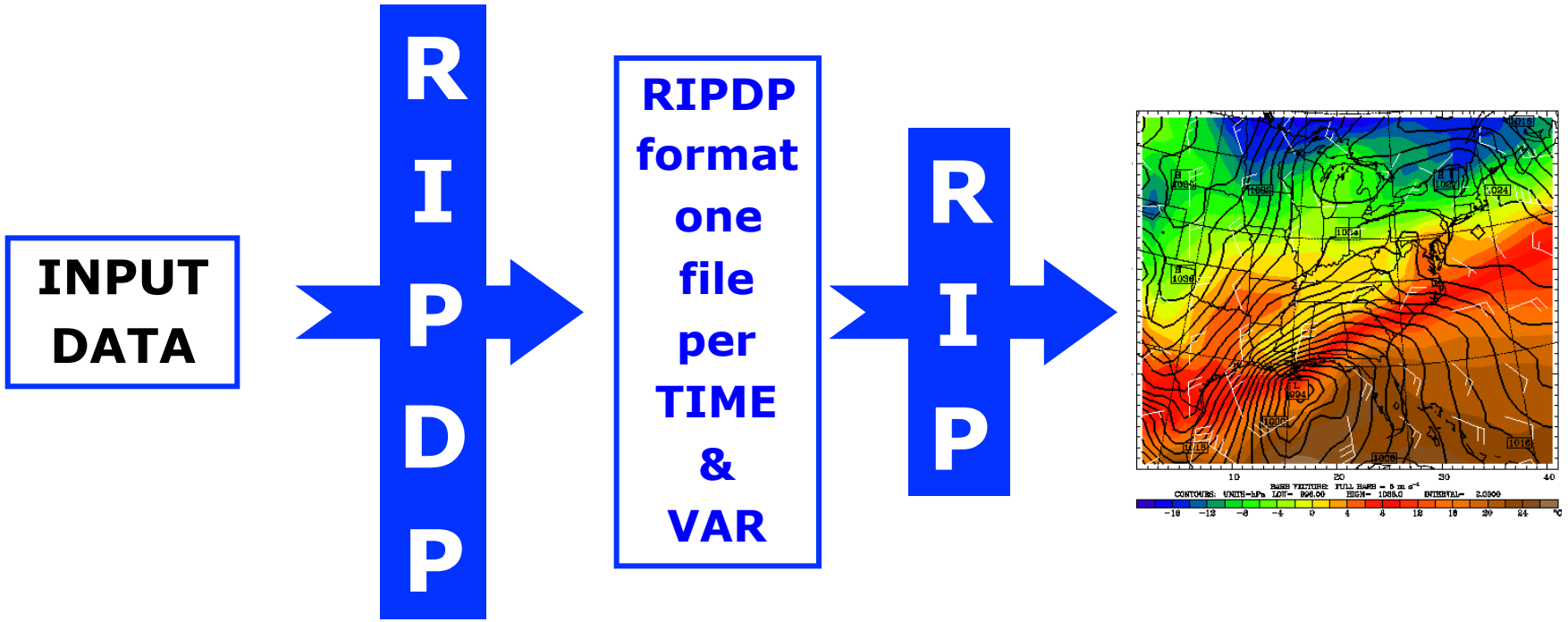


Graphical Packages

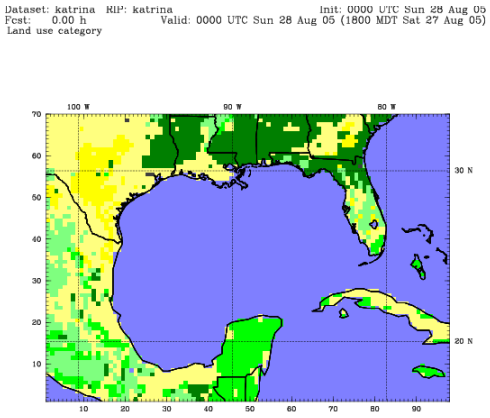
	NCL	RIP4	ARWpost (GrADS)	UPP	VAPOR	IDV
Directly ingest WRF data	***Y***	**N** *converter*	**N /*(Y)*** *converter*	**N** *converter*	**N /*(Y)*** *converter*	**N /*(Y)***
Intermediate files	***N***	lots	large file	**Y**	large file	**Y**
WPS DATA	***Y***	***Y***	***Y***	**N**	**N**	**Y**
wrfinput data	***Y***	***Y***	***Y***	**N**	**N**	**Y**
Idealized data files	***Y***	***Y***	***Y***	**N**	**N**	**Y**
Input format	*netCDF*	*netCDF*	*netCDF*	*netCDF / binary*	*netCDF*	*netCDF*
Vertical Output Coordinate	eta pressure height	eta pressure height	eta pressure height	pressure	eta	if pre-processed
Software required *(All binaries are free)*	*NCL*	*NCARG*	*GrADS*	*GrADS / GEMPAK*	*VAPOR*	*JAVE*
Diagnostics	*some*	*> 100*	*some*	*some*	*limited*	*limited*

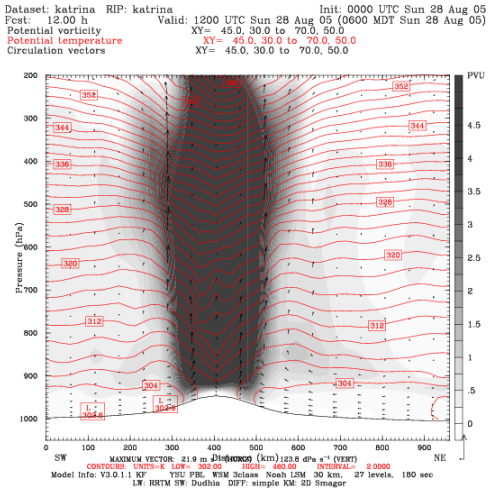


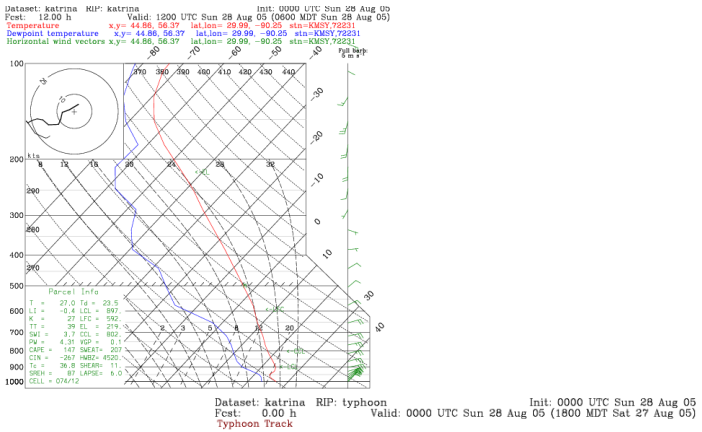
RIP4

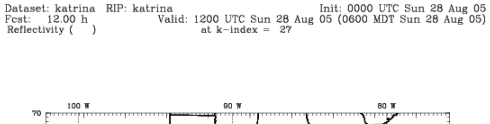

INPUT DATA
R I P D P
RIPDP format one file per TIME & VAR
R I P
Model Dependent

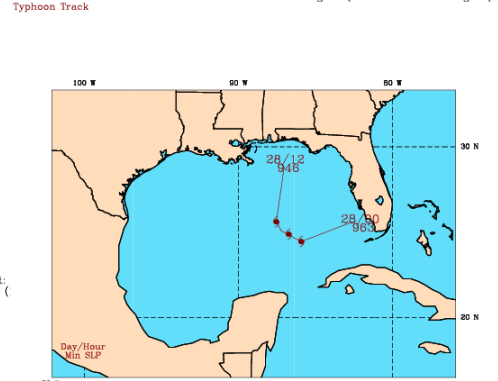
RIP4 - Examples

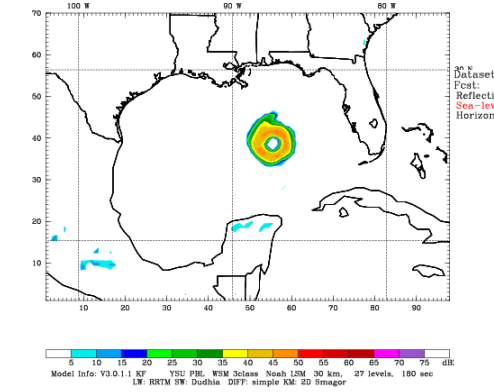


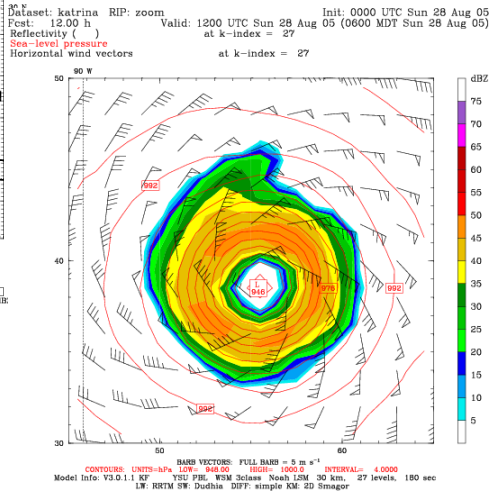


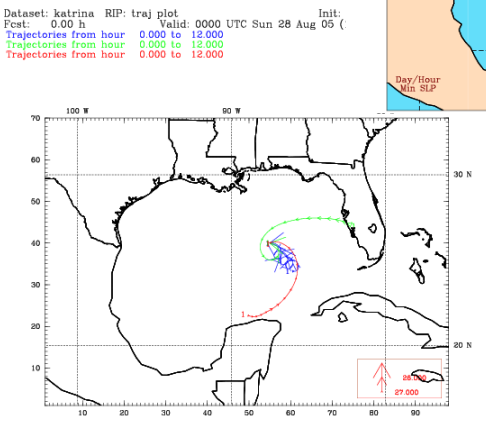






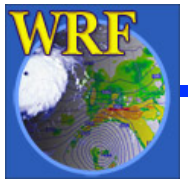






RIP4 General Information

- Requires NCL
  - *http://www.ncl.ucar.edu*
- Source Code:
  - *http://www.mmm.ucar.edu/wrf/users/download/get_source.html*
- Documentation
  - In program tar file under the Doc/ directory
  - *http://www.mmm.ucar.edu/wrf/users/docs/ripug.htm*
- OnLine Tutorial:
  - *http://www.mmm.ucar.edu/wrf/users/graphics/RIP4/RIP4.htm*



RIP4 on your computer

- set environment variables
  setenv RIP_ROOT /usr/$USER/RIP4
  setenv NCARG_ROOT /usr/local/ncl

- Configure
  ./configure
  check configure.rip to ensure netCDF paths are correct
  gfortran ; z and png libraries may be required

```
-L<path_to_png_lib> -lpng
-L<path_to_z_lib> -lz
-L<path_to_gfortran_lib> -lgfortran
```

- Compile
  ./compile

- RIP4 has 2 parts (RIPDP and RIP)

Running ripdp & rip

ripdp_wrfarw [-n namelist-file] <model_data_name> [basic/all] <input_file(s)>

optional

rip [-f] <model_data_name> rip-execution-name

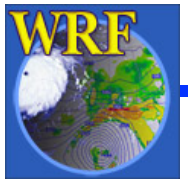
Example:

ripdp_wrfarw RIPDP/CaseX all wrfout*

rip [-f] RIPDP/CaseX rip_sample.in

output

[rip_sample.out]
rip_sample.TYPE

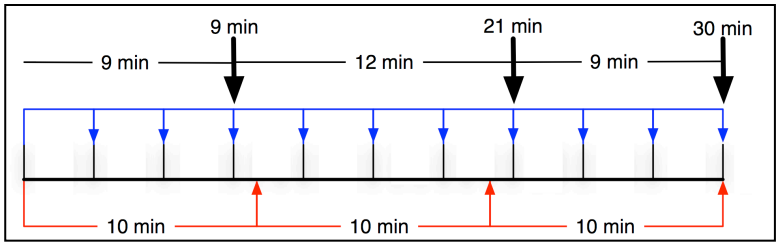


ripdp namelist

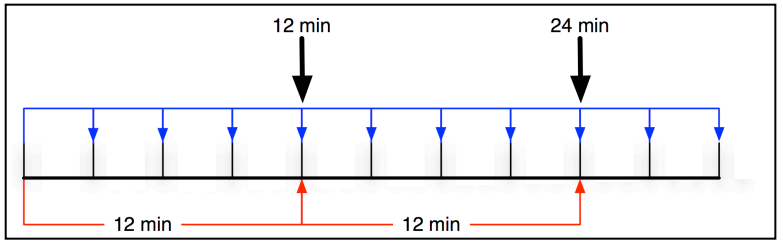
- **ptimes** *(times for ripdp to process)*
  - *List of time* OR *(start, -end, interval)*
  - `0,1,2,3,4,5,6` (0,1,2,3,4,5,6)
  - `0,-6,1` (0,1,2,3,4,5,6)
  - `0,  2,-4,1,  6` (0, 2,3,4, 6)

- **tacc**: *input files not on exact times*
  - 2013-01-31_00
    2013-01-31_09
    2013-01-31_21
    2013-01-31_30
  - Delta Times =
    9 ; 12 ; 9 minutes

history_interval=10 ; time_step=180 (3 min)



history_interval=12 ; time_step=180 (3 min)



RIP4 User Input File

&userin
.....
&end

Namelist controlling general parameters

&trajcalc
.....
&end

Namelist for trajectory calculations
Only used if itrajcalc=1, in userin namelist

==========

---------- Plot Specification Table ----------

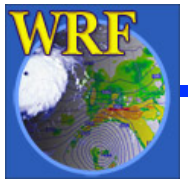
==========

feld= .....
feld= .....

==========

feld= .....
feld= .....

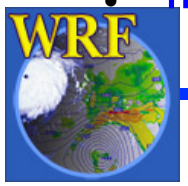
==========



rip 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

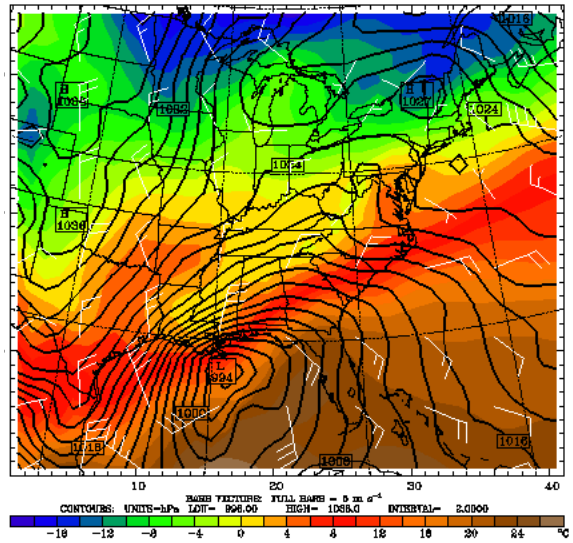


Creating a Plot with RIP4

feld=
diagnostics - *tmc*
native - *PSFC*

vcor=s; **levs=2fb**
vcor=s; **levs=1,2,3**
vcor=p; **levs=800,500**
vcor=p; **levs=800,-300,100**

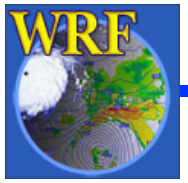
```
==========================================
feld=tmc; ptyp=hc; vcor=s; levs=1fb; >
   cint=2; cmth=fill; >
   cosq=32,light.violet,-16,blue, >
   0,yellow,16,orange,32,light.gray
feld=slp; ptyp=hc; cint=2; linw=2
feld=uuu,vvv; ptyp=hv; vcmx=1; >
    colr=white;intv=5
feld=map; ptyp=hb
feld=tic; ptyp=hb
==========================================
```



RIP4 Common Error Message

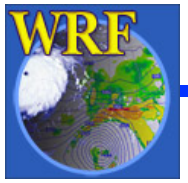
- **Most often this is NOT a graphics error.**
- **More often this is an error with the times you are asking RIP to process**
  - Check the ptimes in your .in file
  - Check the xtimes files created by RIPDP

```
GKS ERROR NUMBER   2 ISSUED FROM SUBROUTINE
GCLKS :--GKS NOT IN PROPER STATE: GKS SHALL BE IN
STATE GKOPFORTRAN STOP
```

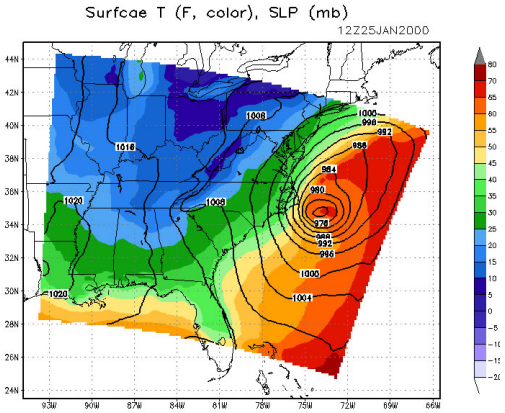


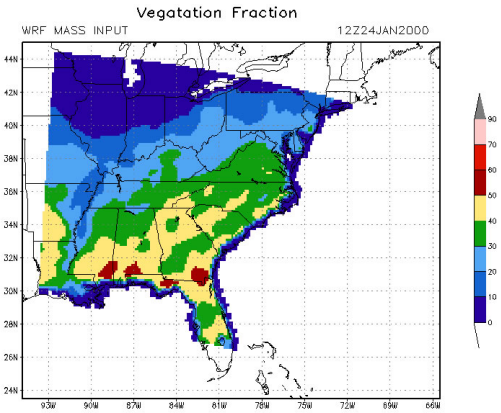
ARWpost

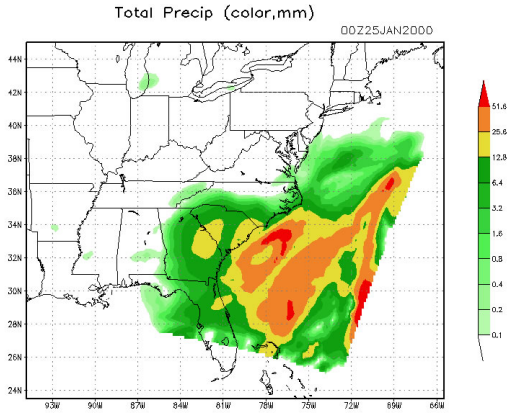
- Converter
  - Requires GrADS to display data
- GrADS software only needed to display data, not needed to compile the code
- Generate a number of graphical plots
  - Horizontal, cross-section, skewT, meteogram, panel
- Version 2 (old – not recommended)
  - Could produce vis5d output
  - Needed WRFV3 complied

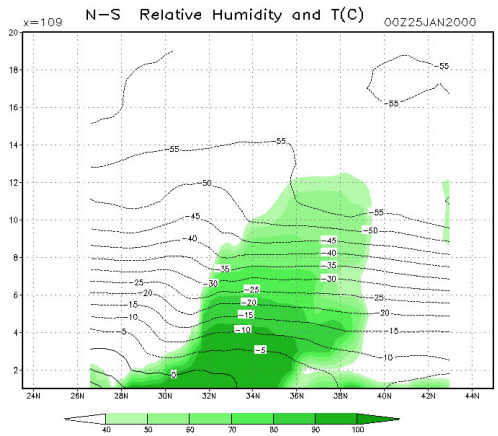


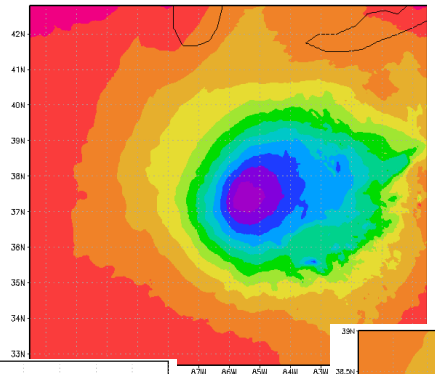
ARWpost - Examples

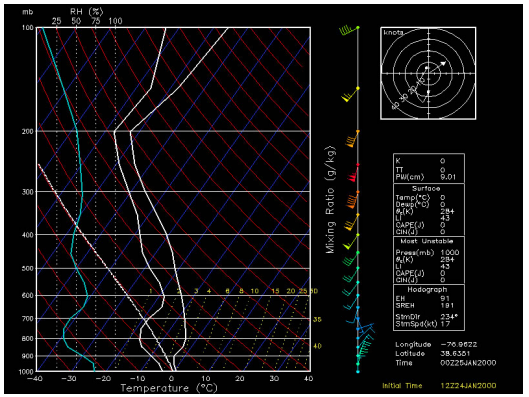


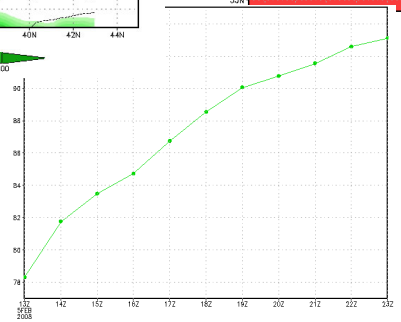


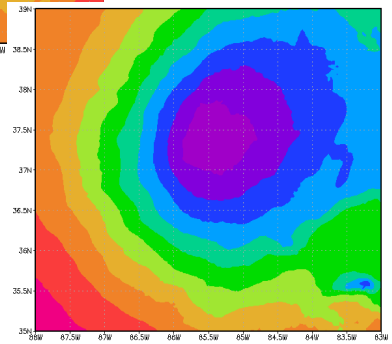












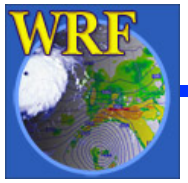
ARWpost - *converter*

- **Download Code** (http://www.mmm.ucar.edu/wrf/users)

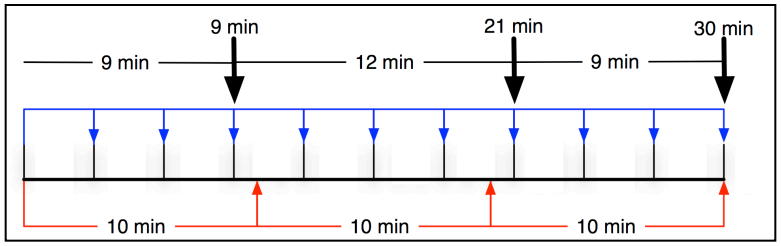
- **OnLine Tutorial**
  http://www.mmm.ucar.edu/wrf/users/graphics/ARWpost/ARWpost.htm

- **Compile** ***(similar to WPS)***
  ./configure & ./compile

- **For GrADS output**
  - GrADS libraries only needed to display data *(freely available)*
  - http://grads.iges.org/grads/grads.html

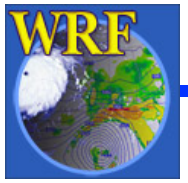


namelist.ARWpost

start_date *end_date*	Start & end date Format: *YYYY-MM-DD_HH:mm:ss*
interval_seconds	Seconds between times to process. *Code will skip times not required. Data can be in multiple files.*
tacc	If model output is not at regular intervals, use closest time within *tacc* seconds of time requested. *(150 sec)* 
debug_level	Set high for extra information

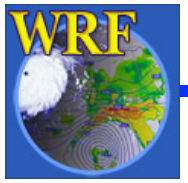
namelist.ARWpost

input_root_name	**Path** and **root** name of files to use as input. *Do not only provide directory name.* Can use wild characters.
output_root_name	Output root name. output_root_name.**dat** & output_root_name.**ctl**
mercator_defs	Set to true if mercator plots are distorted

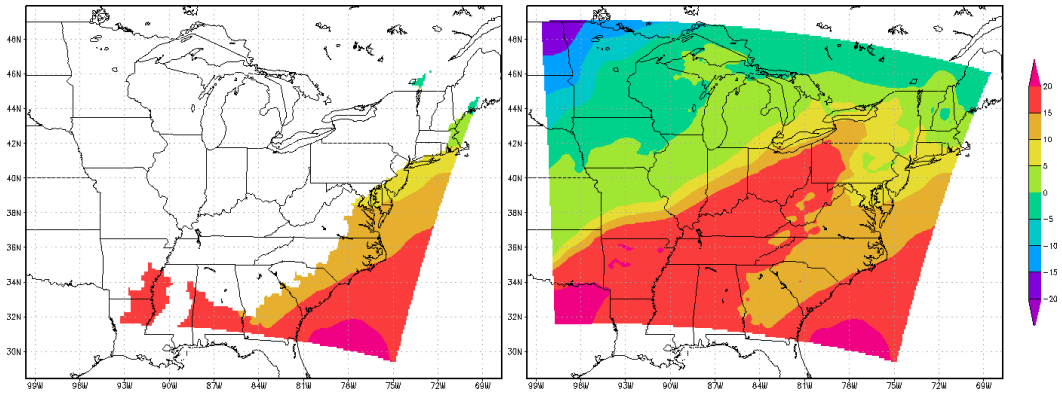


namelist.ARWpost

split_output	Split your GrADS output files into a number of smaller files *(a common .ctl file will be used for all .dat files).*
frames_per_outfile	If *split_output* is .**True.**, how many time periods are required per output (.dat) file.
plot	Which fields to process. (*all, list, all_list)* Order has no effect, i.e., "all_list" and "list_all" "**list**" – list variables in "**fields**"
fields	Fields to plot. Only used is list was used in the "plot" variable. **Must use to generate diagnostics.**
Available diagnostics: cape, cin, mcape, mcin, clfr, dbz, max_dbz, geopt, height, lcl, lfc, pressure, rh, rh2, theta ,tc, tk, td, td2, slp, umet, vmet, u10m, v10m, wdir, wspd, wd10, ws10	



namelist.ARWpost

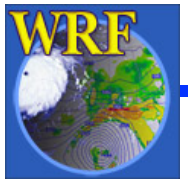
interp_method	0 = sigma levels, -1 = code defined "nice" height levels, 1 = user defined height or pressure levels
interp_levels	Only used if interp_method=1 Supply levels to interpolate to, in hPa (*pressure*) or km (*height above sea level*) Supply levels bottom to top
extrapolate	Extrapolate below ground *(default .false.)* 

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
```

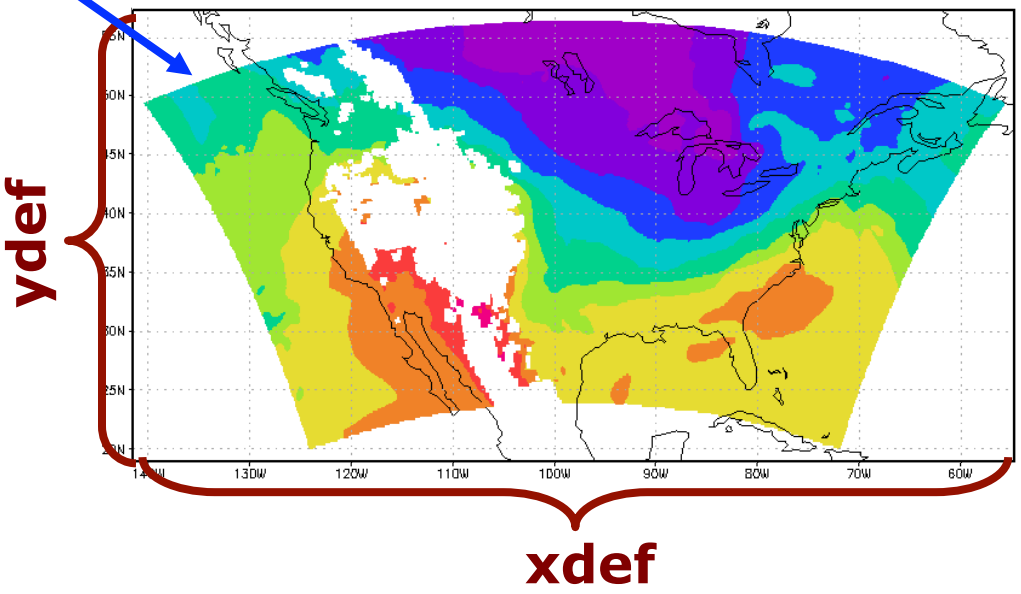
options byteswapped

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



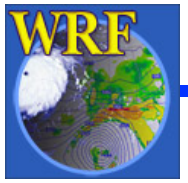
GrADS - .ctl file

```
dset ^test.dat
options byteswapped
title  OUTPUT FROM WRF V3.2 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
```



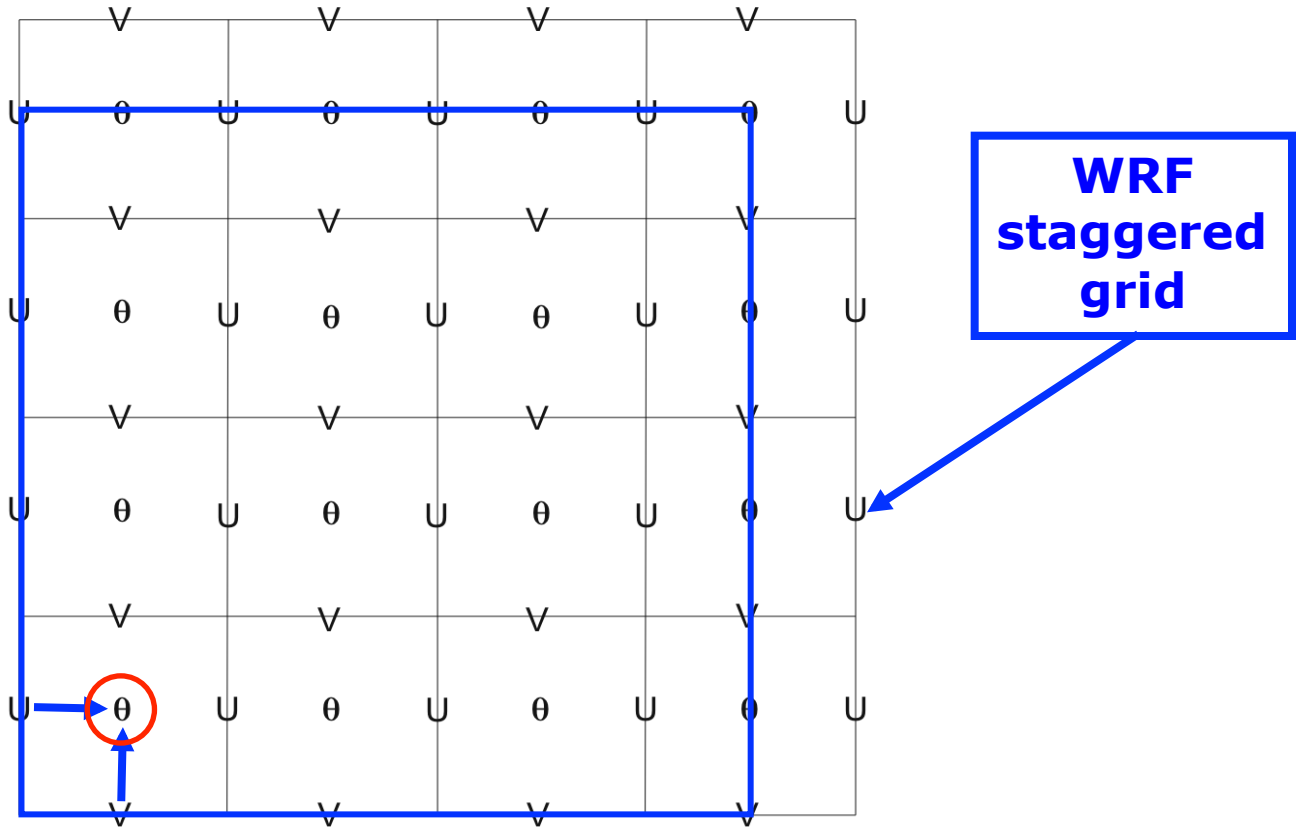
GrADS conversion - question

- **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
  - GRIB model output can also be read directly by GrADS, but above issues are still valid
  - For GRIB, there is also a stagger problem



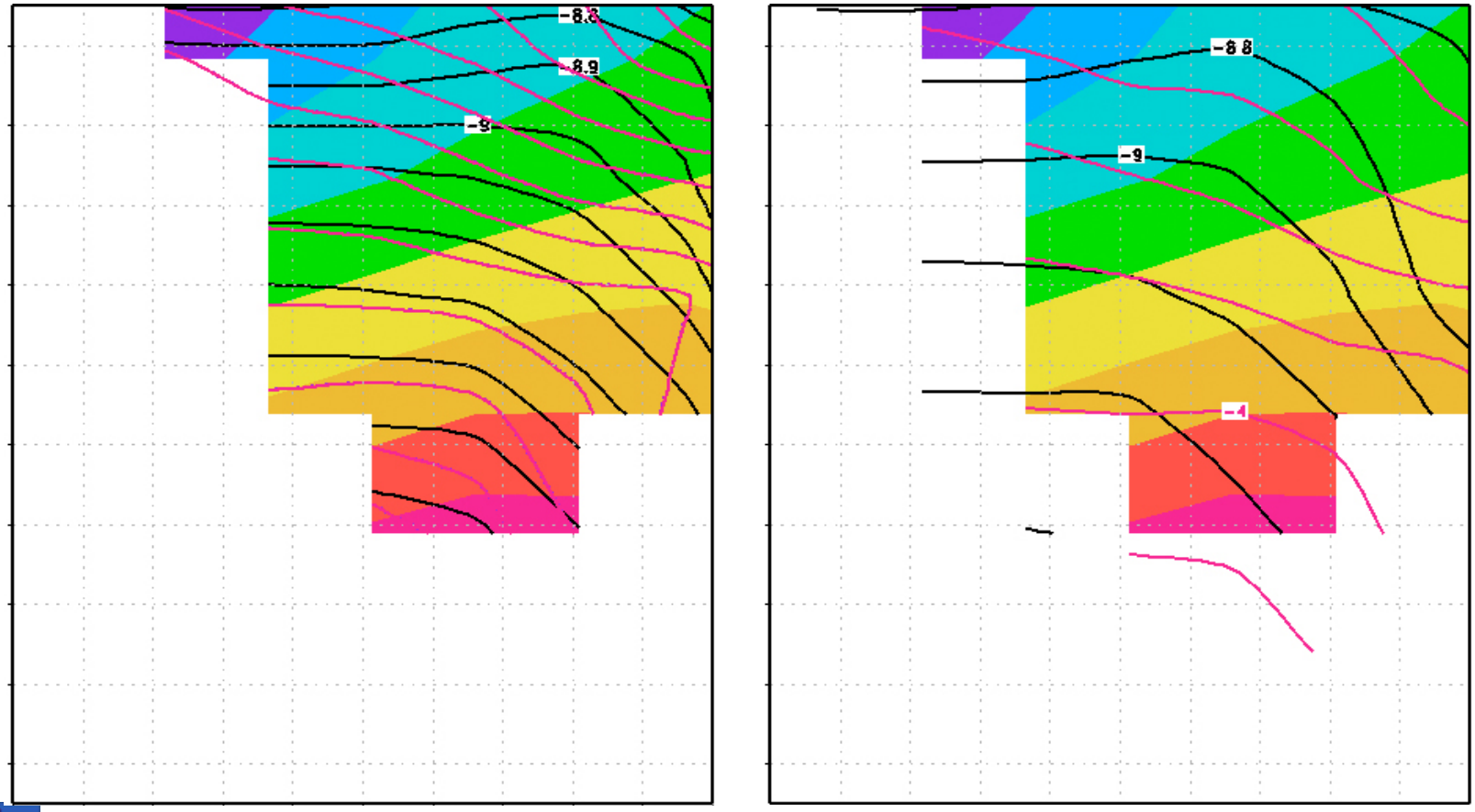
GrADS conversion - question

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



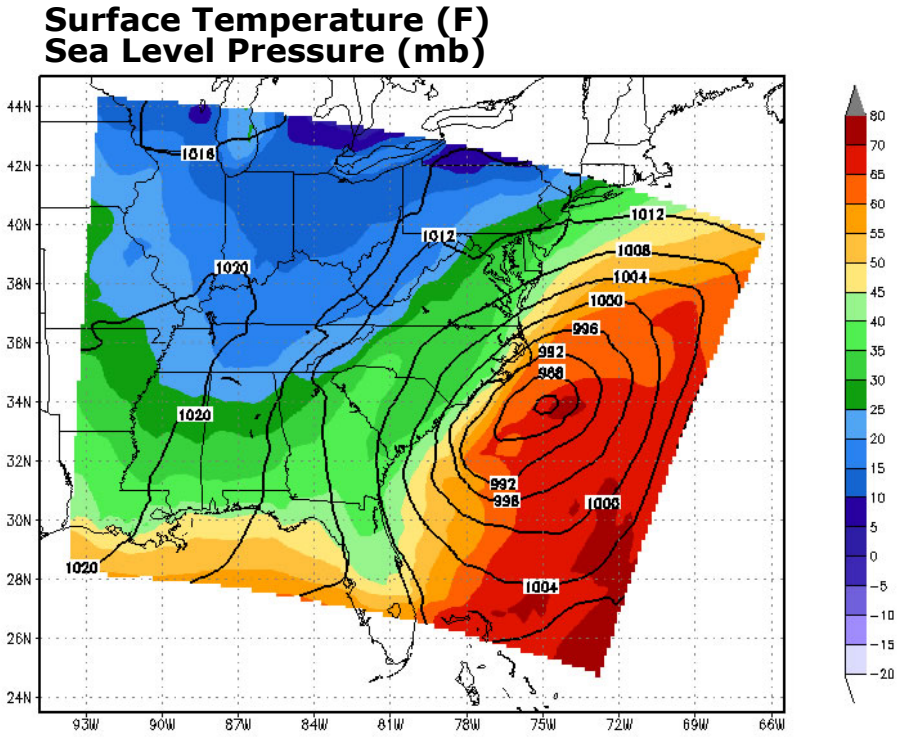
Staggering

shaded=T ; black=U ; red=V



Creating a Plot

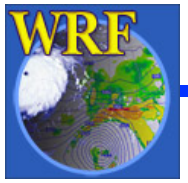
```
open em_real.ctl
set mpdset hires
set display color white

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
```



How to add diagnostics

- **RIP4**
  - Create a subroutine (note RIP4 expects the code to be in "j/I/-k" orientation)
  - Add links to the RIP4/src/fields.f routine
  - Add new subroutine to RIP4/src/Makefile

- **ARWpost**
  - Create a subroutine
  - Add links to ARWpost/src/module_diagnostics.f90
  - Add new subroutine to ARWpost/src/Makefile

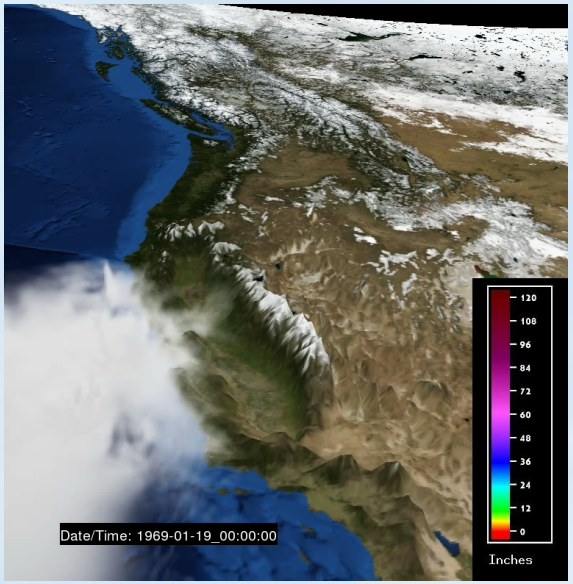


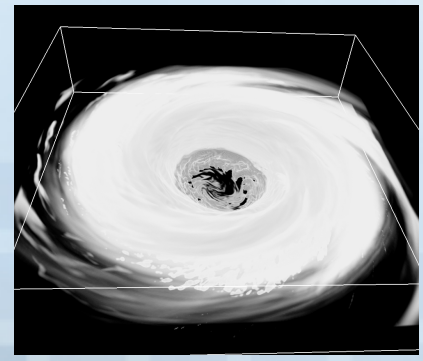
VAPOR visualization of WRF-ARW data

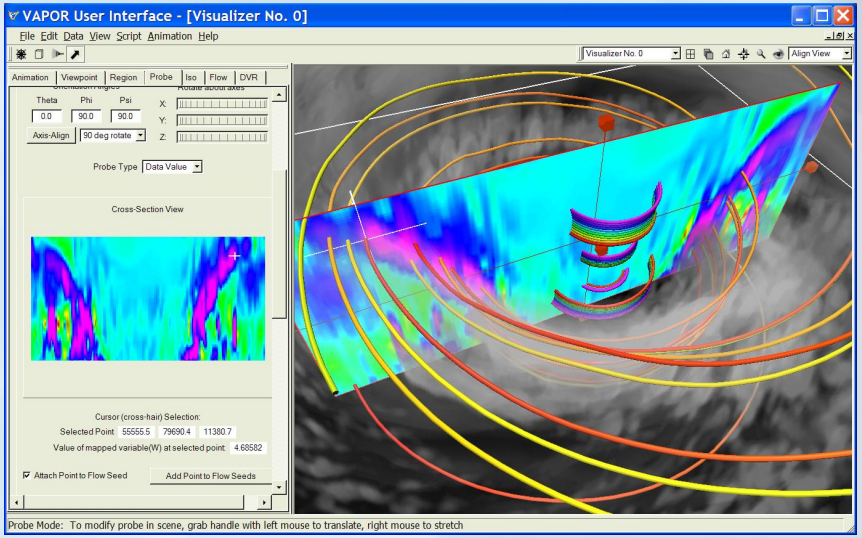
Visualization and Analysis Platform for Oceanic, atmospheric and solar Research

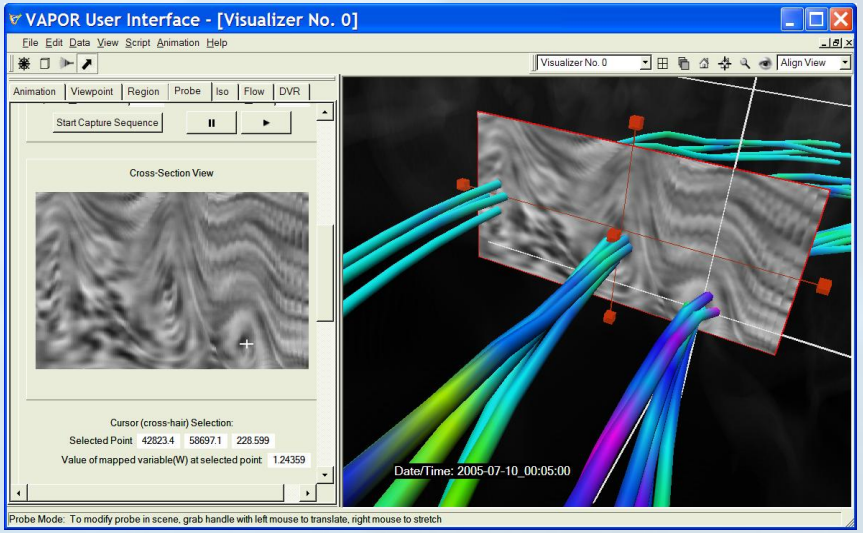
Alan Norton
alan@ucar.edu
vapor@ucar.edu
National Center for Atmospheric Research

VAPOR visualization of WRF-ARW data









VAPOR Installation

- Available for Linux, Windows, or Mac systems
- Should have a reasonably modern graphics card
  - nVidia, ATI or AMD graphics cards are good; others may not perform all visualizations.
- From the VAPOR website http://www.vapor.ucar.edu: Download appropriate binary installer from the VAPOR download page, follow the installation instructions.
- You will need Administrative privileges on Mac
- Note that on Linux and Mac you need to source vapor-install.csh in your shell before running any VAPOR commands.
- Run the vaporgui application to visualize your data

VAPOR visualization of WRF-ARW data

NCAR

A short summary of VAPOR capabilities

1. Read or convert WRF-ARW output files
2. Apply geo-referenced images to the terrain
3. Calculate 2D and 3D derived variables in Python
4. Volume render 3D variables
5. Display isosurfaces of 3D variables
6. Display color-mapped 2D variables on planes or terrain-mapped.
7. Use wind barbs to show flow direction and speed
8. Display streamlines or path lines in scene
9. Insert contour planes, use them to position flow seeds.
10. Image-based flow shows flow motion in 2D slices
11. Create animated 3D sequences

CISL

VAPOR
Visualization & Analysis Platform

Reading or converting WRF-ARW output files

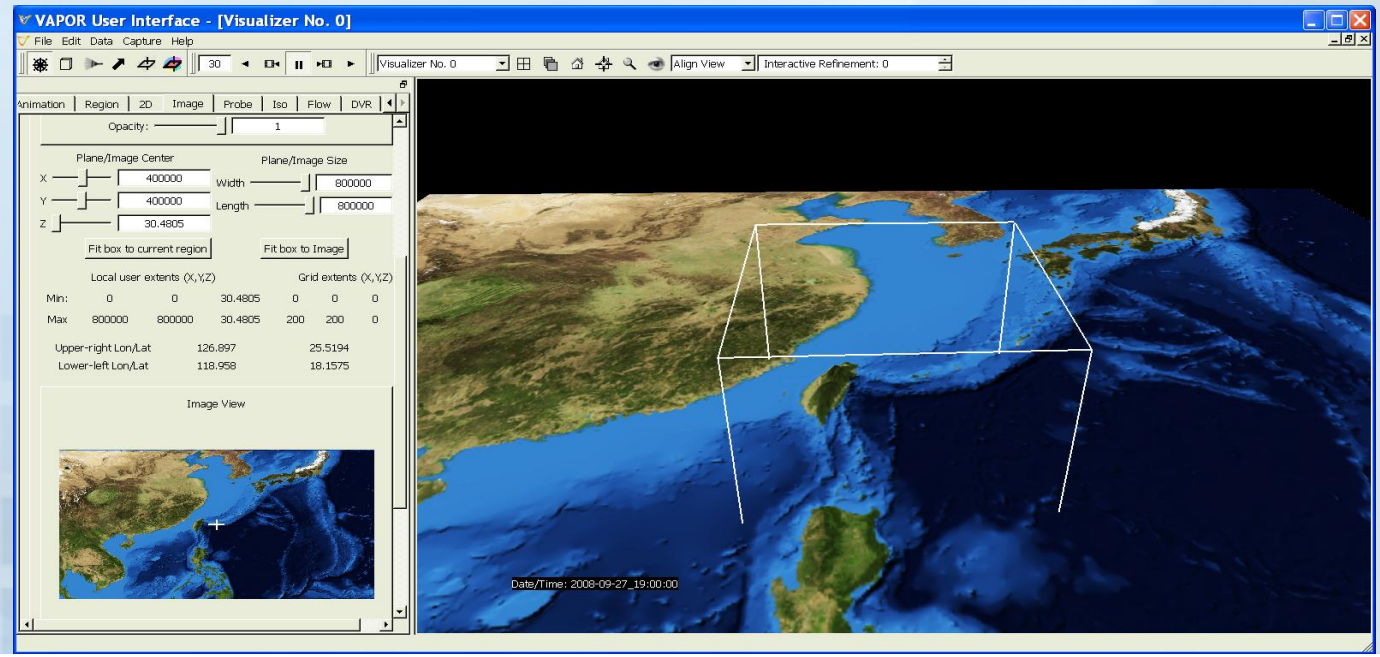
- To directly read WRF output:
  - Run vaporgui
  - All data must be on the same grid, using the same nesting level.
  - Specify "Import WRF-ARW output files" from the Data menu, and select all the wrfout files to visualize
- For interactive visualization of large WRF-ARW datasets, it's best to convert WRF data to the VAPOR data format, using wrfvdfcreate and wrf2vdf utilities.
  - `wrfvdfcreate wrfoutfiles… vdffile.vdf`

    creates a VAPOR metadata file "vdffile.vdf" that describes a set of wrfout files.
  - `wrf2vdf vdffile.vdf wrfoutfiles…`

    converts the specified wrfout files to a vapor data collection
  - From the vaporgui Data menu, load the file "vdffile.vdf" to visualize the converted data

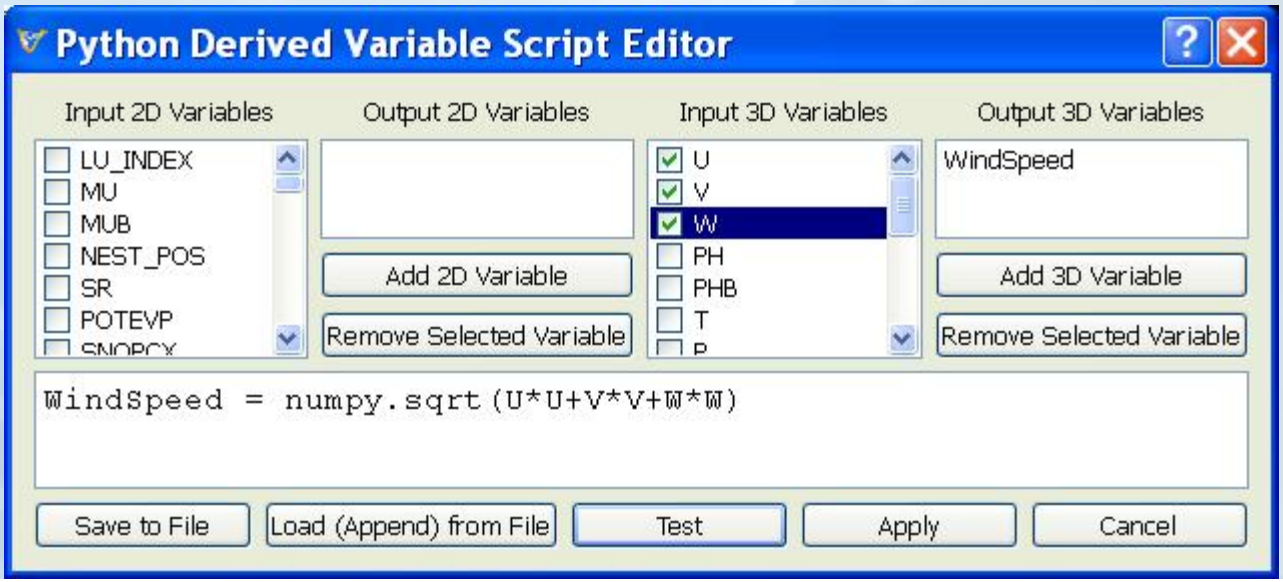
Apply images to use in the VAPOR scene

- Geo-referenced satellite images can be retrieved from the Web, and VAPOR will insert them at the correct world coordinates.
  - VAPOR provides a shell script "getWMSImage.sh" that can be used to retrieve Web Mapping Service images for a specified longitude/latitude rectangle
- Also, several useful images are installed with vapor; e.g. state or national boundary maps, NASA's Blue Marble image of the earth.
- From the image panel, specify the image file, apply to terrain.



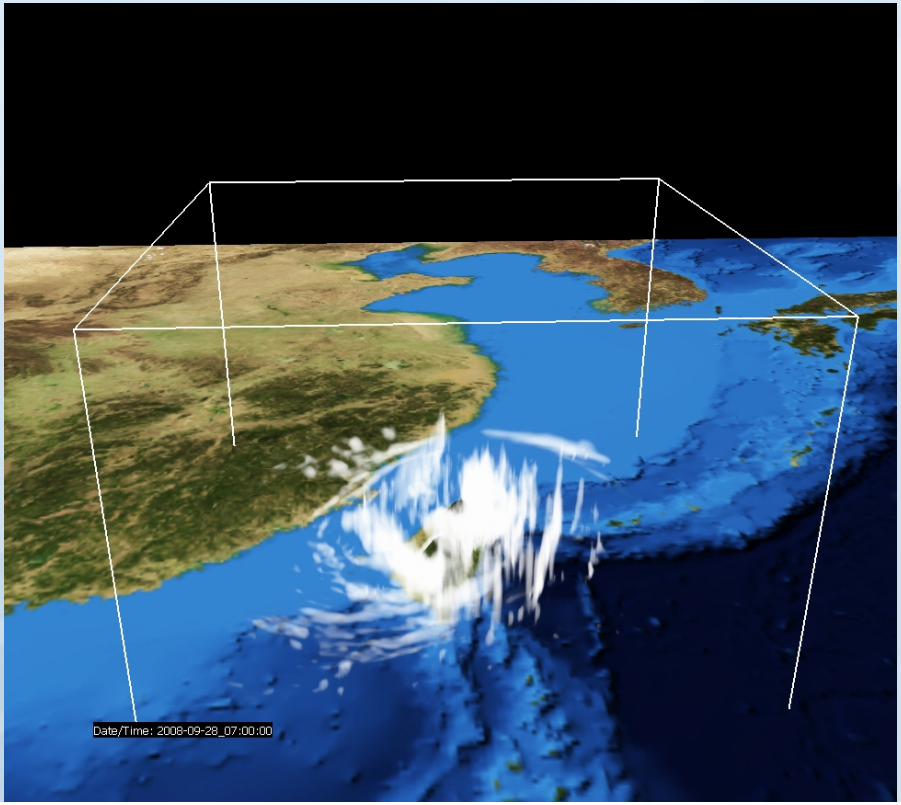
Create derived variables with Python

- From the Edit menu, “Edit Python program defining a new variable”
- Use Python script editor to define variables as arithmetic expressions of other variables.
- Variables are evaluated and cached as needed for visualization
- Python functions are also provided to derive several useful variables from WRF data; e.g. cloud-top temperature, relative humidity, potential vorticity, sea-level pressure, dewpoint temperature, radar reflectivity, equivalent potential temperature, wind shear, temperature in degrees Kelvin.



Volume-render 3D variables to identify important features in the data

- Volume rendering can be used to identify significant 3D features of the WRF data.
- Use a transfer function to control transparency and color:
  - Make the unimportant features transparent, to highlight items of interest
  - Color can be used to distinguish different values of variable

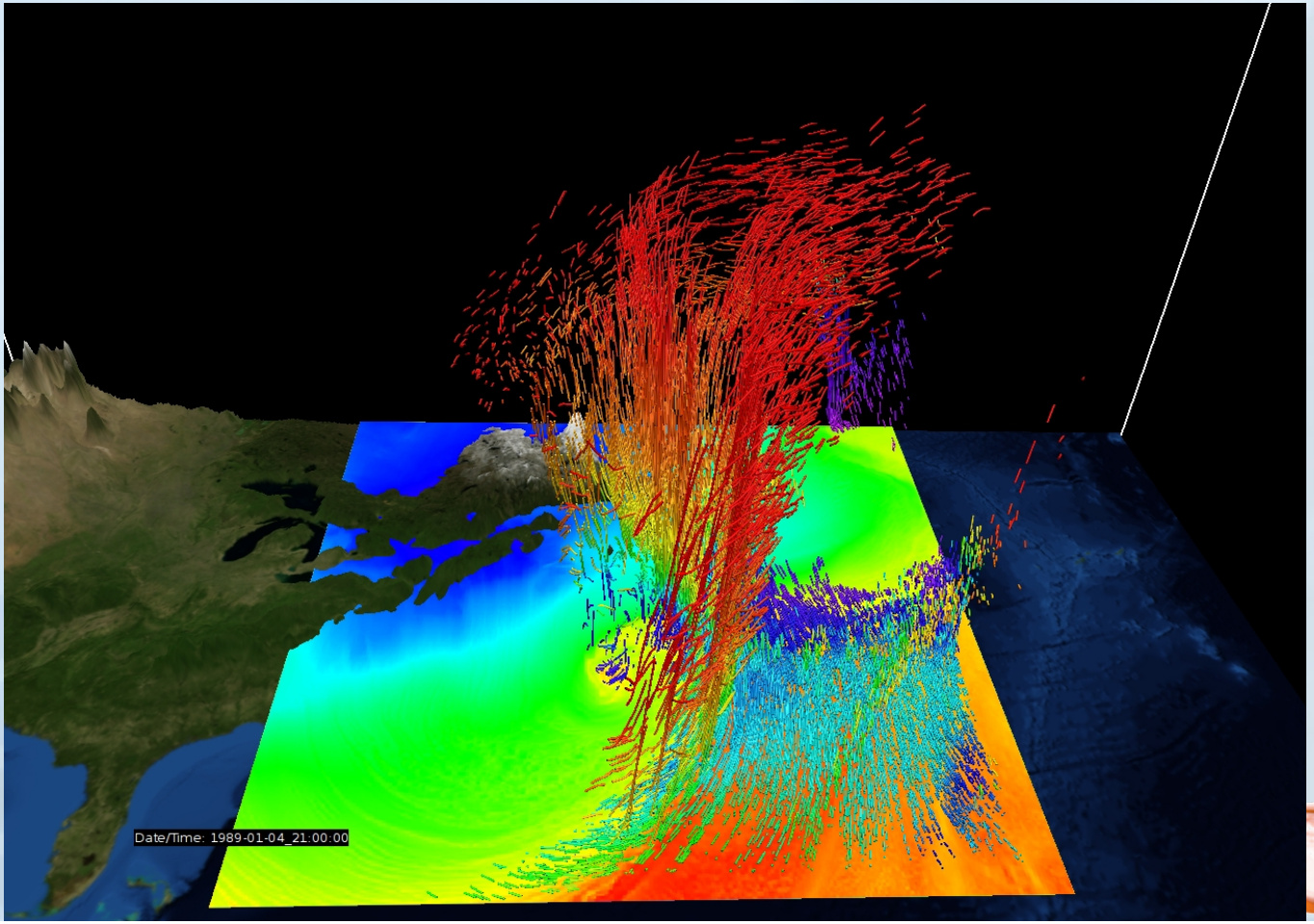


Wind Barbs

- Specify a grid of seed points in the scene where barbs will appear
  - This grid can be aligned to the WRF data grid
- Specify variables that define X, Y, and Z components of velocity field.
- Grid can be displaced by terrain height.

Streamlines and Path lines

- In the Flow panel, specify flow type and velocity components.
- Streamlines indicate the instantaneous direction of the wind flow.
- Path lines track the movement of massless particles over time.



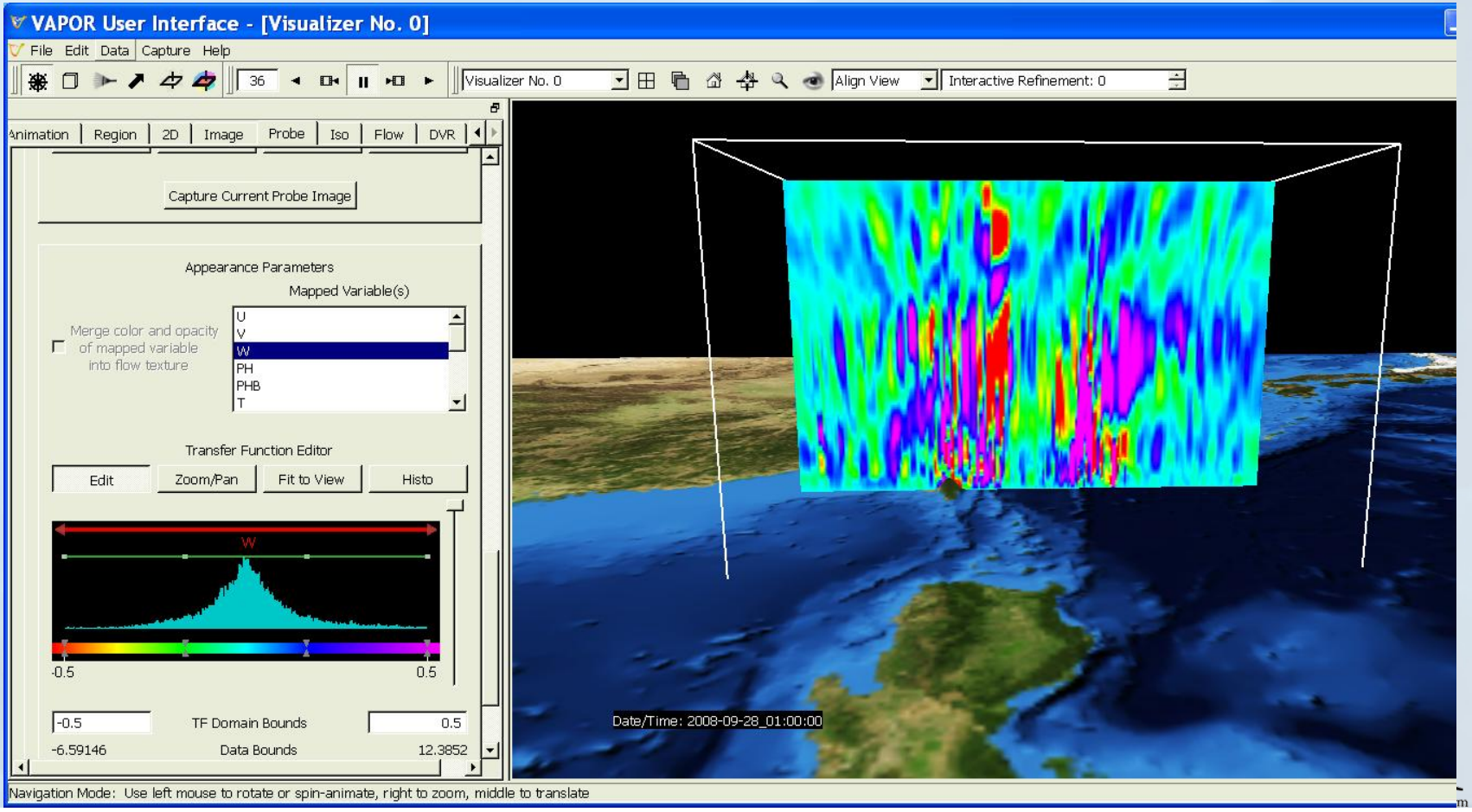
NCAR

CISL

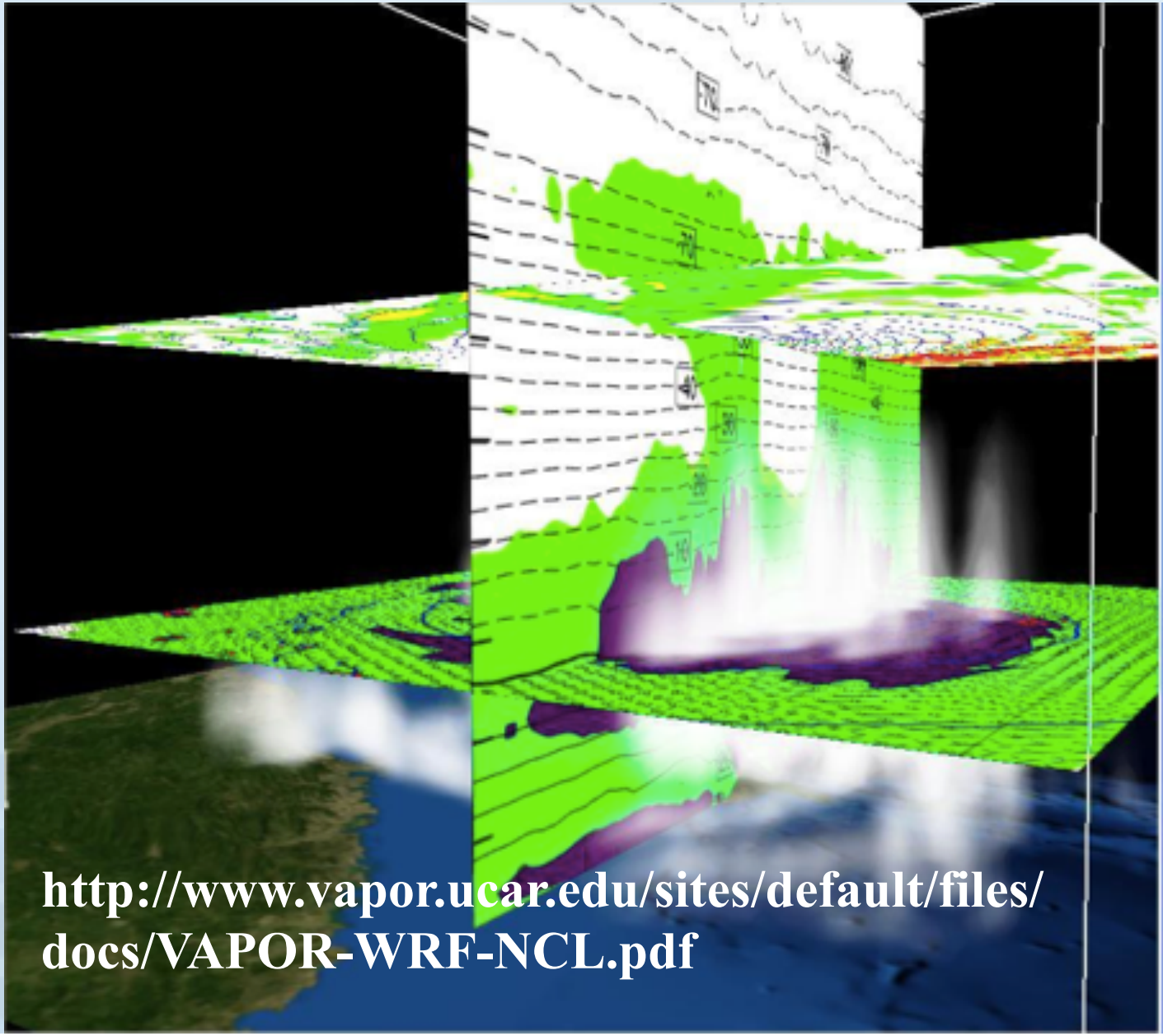
VAPOR
Visualization & Analysis Platform

Contour planes and the Probe

- Use the Probe panel to position rectangle in scene
- Use transfer function to color-map the rectangle
- Probe can also enable interactive specification of flow seeds.



VAPOR / NCL



http://www.vapor.ucar.edu/sites/default/files/docs/VAPOR-WRF-NCL.pdf

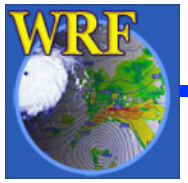
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl”


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",-1)
 times = wrf_user_getvar(a,"times",-1)
 ntimes = dimsizes(times)

 mpres = True
 pltres = True



 do it=0,ntimes-1
  opts = True
  opts@cnFillOn = True
  contour_t2 = wrf_contour(a,wks,T2(it,:,:),opts)
  plot = wrf_map_overlays(a,wks,(/contour_t2/),pltres,mpres)



 end do
```

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl”
load "$VAPOR_HOME/share/examples/NCL/wrf2geotiff.ncl"

begin
 a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r”)
 wks = gsn_open_wks(“ps”,"plt_Surface")
 wrf2gtiff = wrf2geotiff_open(wks)

 T2 = wrf_user_getvar(a,"T2",-1)
 times = wrf_user_getvar(a,"times",-1)
 ntimes = dimsizes(times)

 mpres = True
 pltres = True
 pltres@gsnFrame = False

 do it=0,ntimes-1
  opts = True
  opts@cnFillOn = True
  contour_t2 = wrf_contour(a,wks,T2(it,:,:),opts)
  plot = wrf_map_overlays(a,wks,(/contour_t2/),pltres,mpres)
  wrf2geotiff_write(wrf2gtiff,a,times(it), wks, plot, False)
  frame(wks)
 end do
```

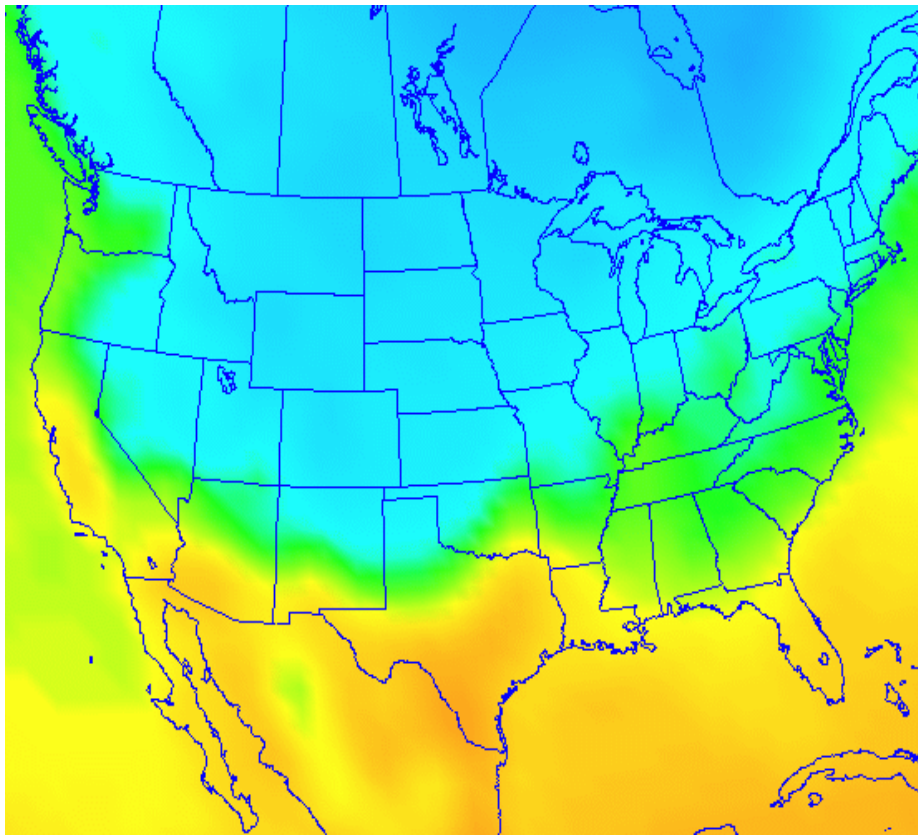
IDV

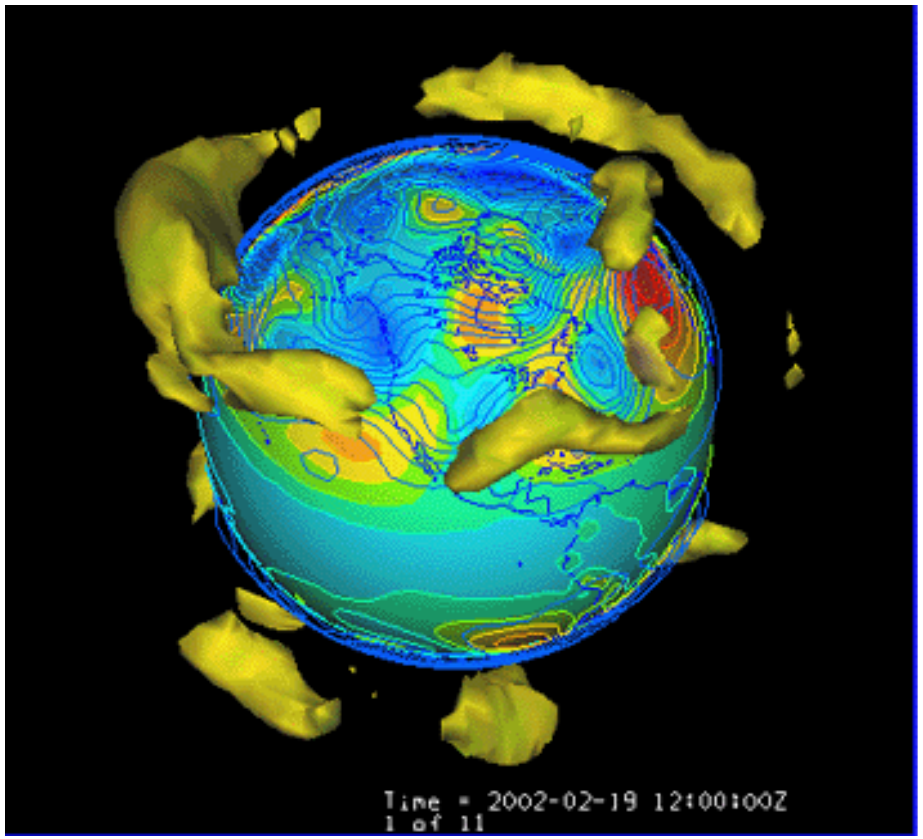
Integrated Data Viewer

Yuan Ho and Julien Chastang
Unidata Program Center/UCAR

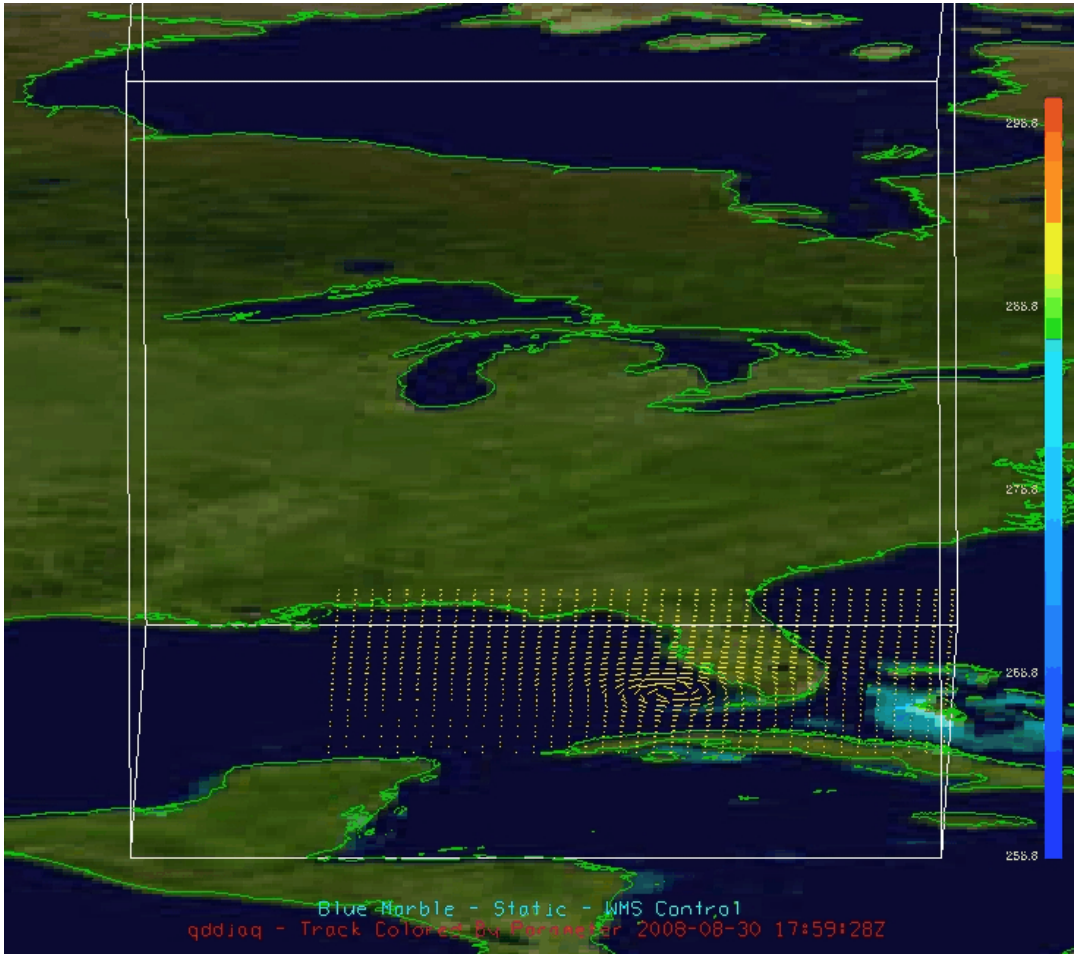


Unidata IDV – What can it do?



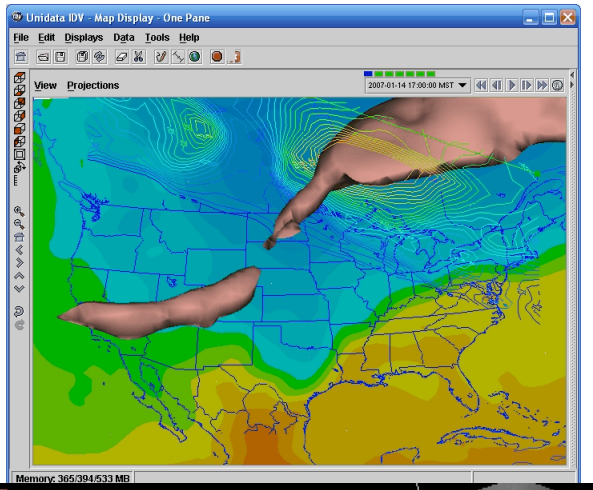


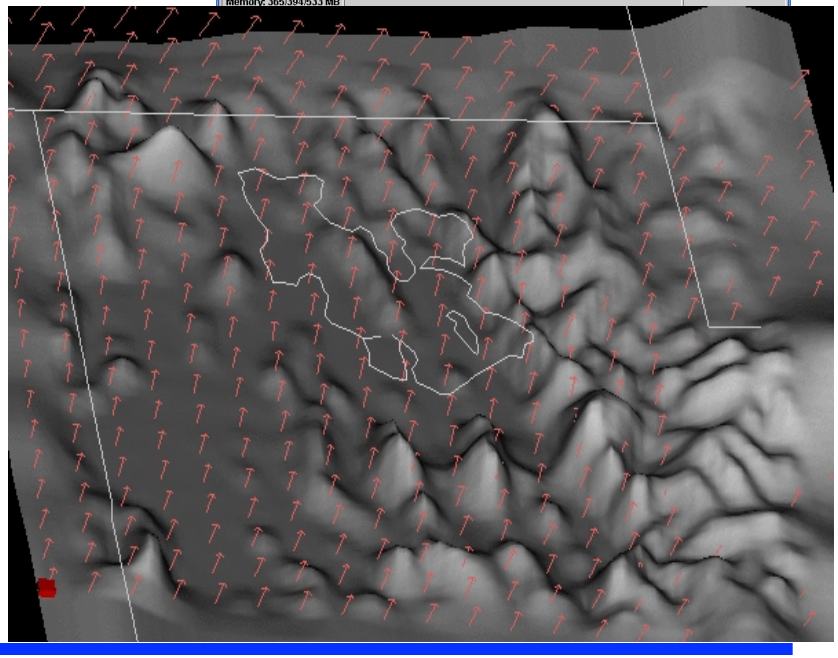
Unidata IDV – What can it do?



Unidata IDV – What is it?

- **Unidata Integrated Data Viewer**
  - 2D and 3D visualization
    - GUI or scripting (Jython, ISL) interface
  - Interactive probes for dataset exploration
    - Parameter readouts, vertical profiles, time/height displays, etc.
  - A rich set of analysis capabilities
  - Integrate model and observational data
  - Access local and remote datasets
  - Visualize and analyze post-processed WRF output
    - works best if grid is unstaggered





Supported Data Sources

- **Data Types:**
  - Gridded model output
  - Satellite imagery
  - Radar data
  - Point observations
  - Balloon soundings
  - NOAA Profiler Network winds
  - Aircraft Tracks
  - Fronts
  - GIS data (WMS, shapefile)
  - Quick Time movies
  - Web Cams
- **Vertical Coordinates**
  - Pressure
  - Height/Depth
  - Other (2D only)
- **Sample of Supported Formats:**
  - netCDF
  - GRIB
  - Vis5D
  - KML
  - CSV
  - GEMPAK grid
  - ADDE
- **Access Methods:**
  - Local files
  - HTTP
  - ADDE, TDS and OPeNDAP servers
  - WMS

ADDE = Abstract Data Distribution Environment
TDS (THREDDS) = Thematic Realtime Environmental Distributed Data Services

Unidata IDV – Where to get it?

- 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 RAM
  - Java 1.6
  - *Latest* video card driver