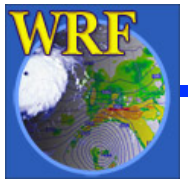
Post-processing Tools

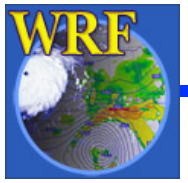
Cindy Bruyère



Graphical Packages

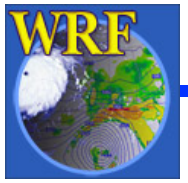
- **NCL** — UG: 9-2
  - Graphical package
- **ARWpost** — UG: 9-28
  - Converter (GrADS)
- **RIP4** — UG: 9-19
  - Converter and interface to graphical package NCAR Graphics
- **WPP** — UG: 9-35
  - Converter (GrADS & GEMPAK)
- **VAPOR** — UG: 9-50
  - Converter and graphical package
  - *Support: VAPOR*
- **IDV** — unidata.ucar.edu
  - GRIB (from WPP)
  - GEMPAK (from wrf2gem)
  - vis5d (from ARWpost)
  - CF complaint data (from wrf_to_cf)
  - *Support: unidata*
- **GEMPAK**
  - Data from wrf2gem or WPP
  - *Support: unidata*

MatLab / IDL / R / ferret



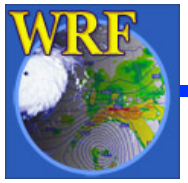
Graphical Packages

	NCL	RIP4	ARWpost (GrADS)
Directly ingest WRF data	***Y***	**N** *converter*	**N /** ***(Y)*** *converter*
Intermediate files	***N***	lots	large file
WPS DATA	***Y***	***Y***	***Y***
wrfinput data	***Y***	***Y***	***Y***
Idealized data files	***Y***	***Y***	***Y***
Input format	*netCDF*	*netCDF*	*netCDF / GRIB1*
Vertical Output Coordinate	eta pressure height	eta pressure height	eta pressure height
Software required ***(All binaries are free)***	*NCL*	*NCARG*	*GrADS*
Diagnostics	*some*	*> 100*	*some*



NCL

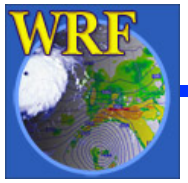
- NCAR Command Language
- http://www.ncl.ucar.edu
- Read WRF-ARW data directly
- Generate a number of graphical plots
  - Horizontal, cross-section, skewT, meteogram, panel
- Download
  - http://www.ncl.ucar.edu/Download
  - Fill out short registration form *(there is a short waiting period)*
  - Read and agree to OSI-based license
  - Get version 5.1.1 or later *(current v5.2)*
- NCARG_ROOT environment variable
  - setenv NCARG_ROOT /usr/local/ncl



Home Directory → ~/.hluresfile ← Very Important

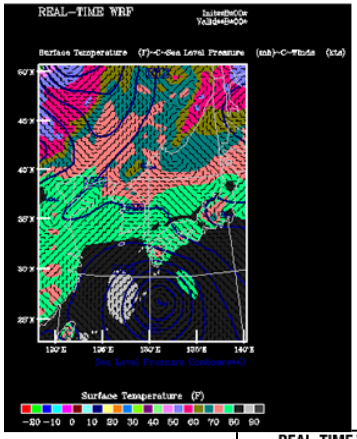
- Required by NCL libraries
- Must be in your "~/" directory *(home directory)*
- Control
  - color table ; font
  - white/black background
  - size of plot
  - control characters

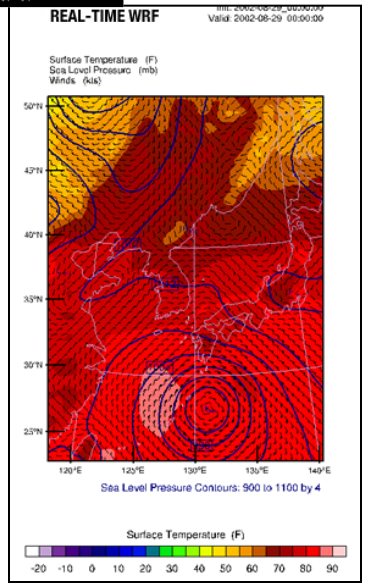
- http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml



~/.hluresfile

```
*wkColorMap           : BlAqGrYeOrReVi200
*wkBackgroundColor    : white
*wkForegroundColor    : black
*FuncCode             : ~
*TextFuncCode         : ~
*Font                 : helvetica
*wkWidth              : 900
*wkHeight             : 900
```



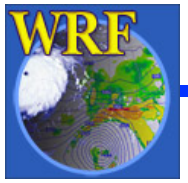



```
PLCHHQ - CHARACTER NUMBER 26 (.) IS NOT A LEGAL FUNCTION CODE
PLCHHQ - CHARACTER NUMBER 29 (t) IS NOT A LEGAL FUNCTION CODE
PLCHHQ - CHARACTER NUMBER 30 (o) IS NOT A LEGAL FUNCTION CODE
PLCHHQ - CHARACTER NUMBER 32 (.) IS NOT A LEGAL FUNCTION CODE
PLCHHQ - CHARACTER NUMBER 35 (b) IS NOT A LEGAL FUNCTION CODE
PLCHHQ - CHARACTER NUMBER 36 (y) IS NOT A LEGAL FUNCTION CODE
```

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

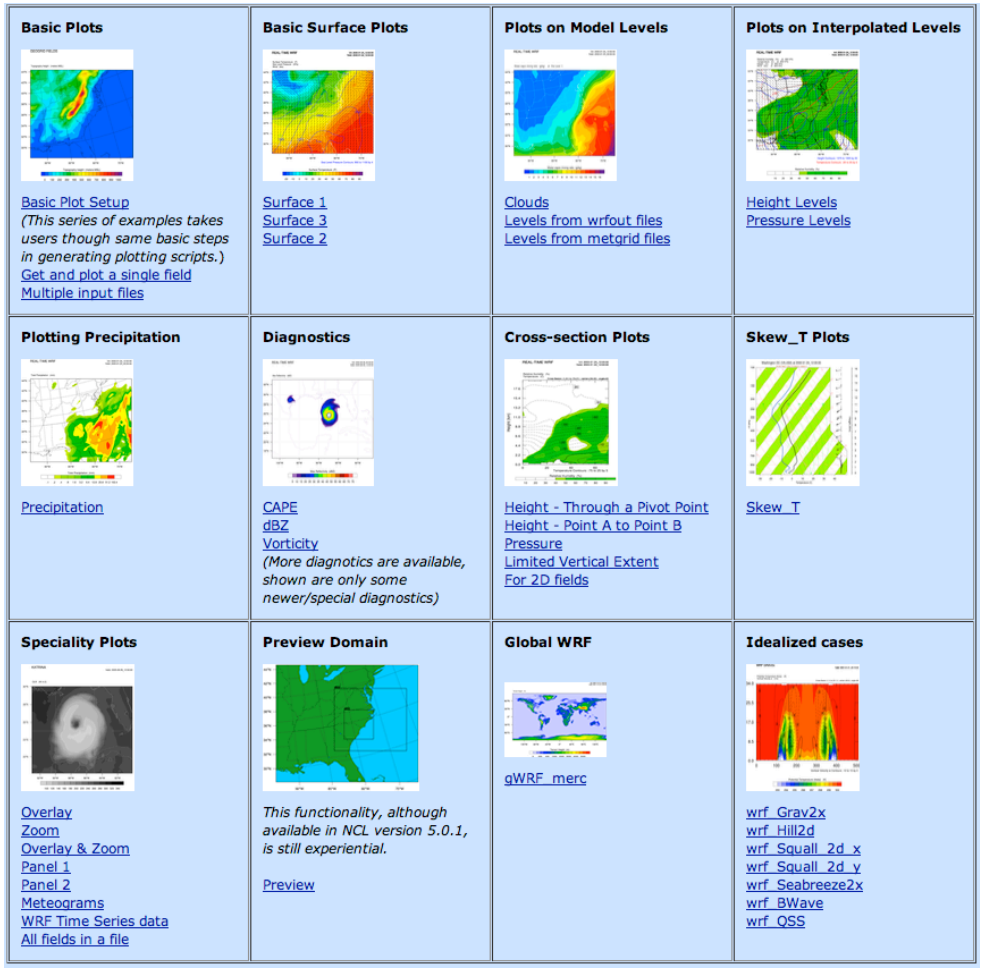
Generate Plots with NCL

- Set NCARG_ROOT environment variable:
  - *setenv NCARG_ROOT /usr/local/ncl*
- Ensure you have an ~/.hluresfile file
- Create a script
  - *wrf_real.ncl*
    (start with a sample script)
- Run NCL script
  - *ncl wrf_real.ncl*



Generate Plots: *A good start - OnLine Tutorial*

http://www.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/index.html



Basic Plots

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

Basic Surface Plots

Surface 1
Surface 3
Surface 2

Plots on Model Levels

Clouds
Levels from wrfout files
Levels from metgrid files

Plots on Interpolated Levels

Height Levels
Pressure Levels

Plotting Precipitation

Precipitation

Diagnostics

CAPE
dBZ
Vorticity
(More diagnotics are available, shown are only some newer/special diagnostics)

Cross-section Plots

Height - Through a Pivot Point
Height - Point A to Point B
Pressure
Limited Vertical Extent
For 2D fields

Skew_T Plots

Skew_T

Speciality Plots

Overlay
Zoom
Overlay & Zoom
Panel 1
Panel 2
Meteograms
WRF Time Series data
All fields in a file

Preview Domain

This functionality, although available in NCL version 5.0.1, is still experiential.

Preview

Global WRF

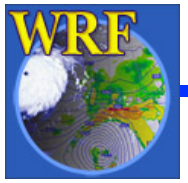
gWRF_merc

Idealized cases

wrf_Grav2x
wrf_Hill2d
wrf_Squall_2d_x
wrf_Squall_2d_y
wrf_Seabreeze2x
wrf_BWave
wrf_QSS

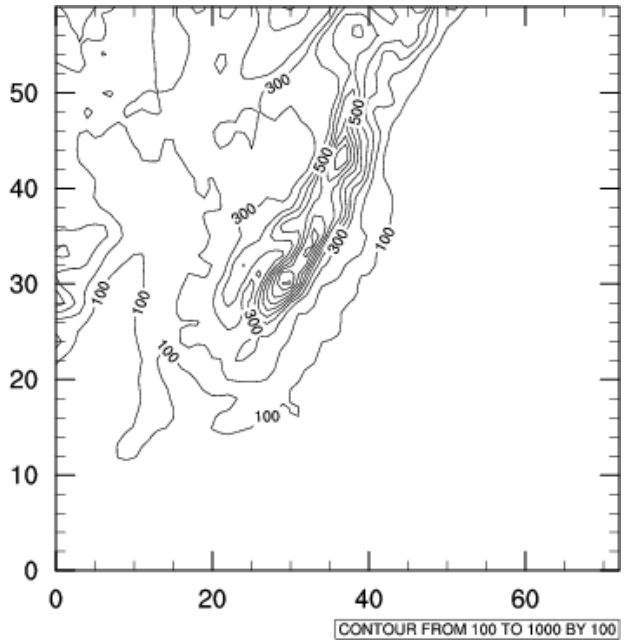
Creating a Plot : NCL script

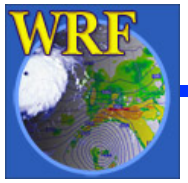
```
load ncl library scripts

begin

  ; Open graphical output
  ; Open input file(s)

  ; Read variables

  ; Set up plot resources & Create plots
  ; Output graphics

end
```



Generate Plots

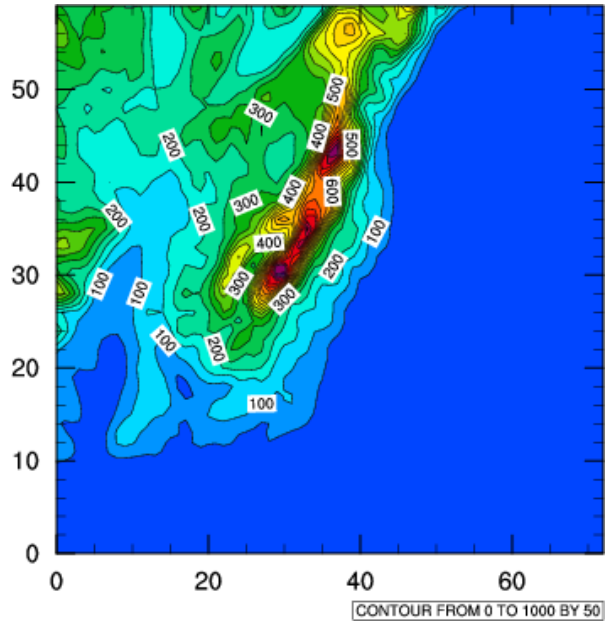
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"

begin
  a = addfile("./geo_em.d01.nc","r")
  wks = gsn_open_wks("pdf","plt_ter1")
  ter = a->HGT_M(0,:,:)
  plot = gsn_contour(wks,ter,True)
end
```





Generate Plots

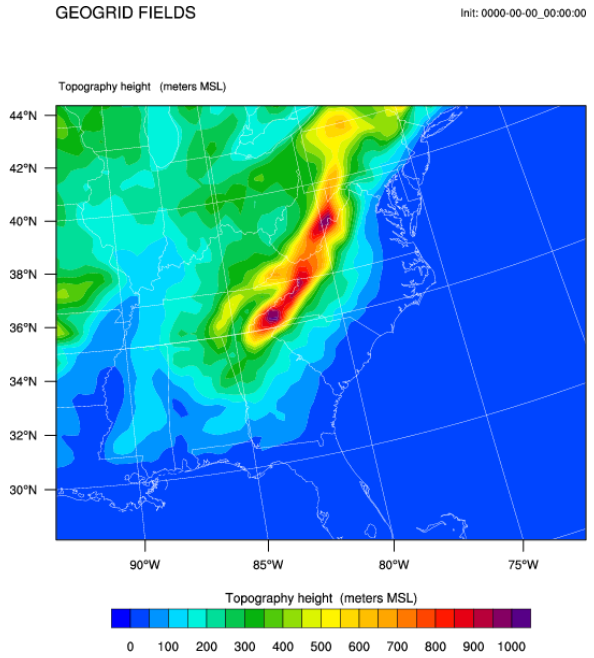
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"

begin
  a = addfile("./geo_em.d01.nc","r")
  wks = gsn_open_wks("pdf","plt_ter1")
  ter = a->HGT_M(0,:,:)
  res = True
  res@cnFillOn = True
  res@gsnSpreadColors = True
  res@cnLevelSelectionMode = \
     "ManualLevels"
  res@cnMinLevelValF = 0.
  res@cnMaxLevelValF = 1000.
  res@cnLevelSpacingF = 50.
  plot = gsn_contour(wks,ter,res)
end
```



Basic NCL resources - *there are over 1400 controlling contours, labelbars, legends, maps, etc.*

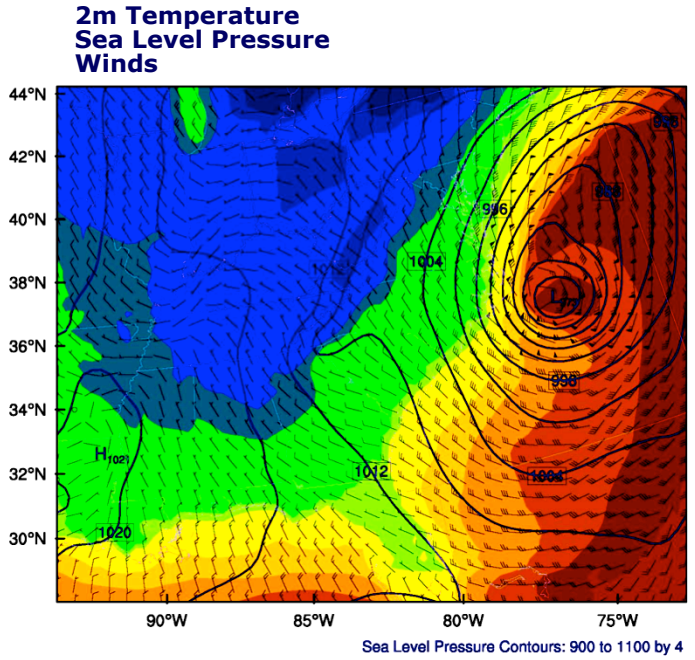
Generate Plots

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

begin
  a = addfile("./geo_em.d01.nc","r")
  wks = gsn_open_wks("pdf","plt_ter5")
  res = True
  res@MainTitle = "GEOGRID FIELDS"
  pltres = True
  mpres = True

  ter = wrf_user_getvar(a,"HGT_M",0)
  res@cnFillOn = True
  res@ContourParameters = (/0.,1000.,50./)
  contour = wrf_contour(a,wks,ter,res)
  plot = wrf_map_overlays(a,wks,(/contour/),\
    pltres,mpres)
end
```



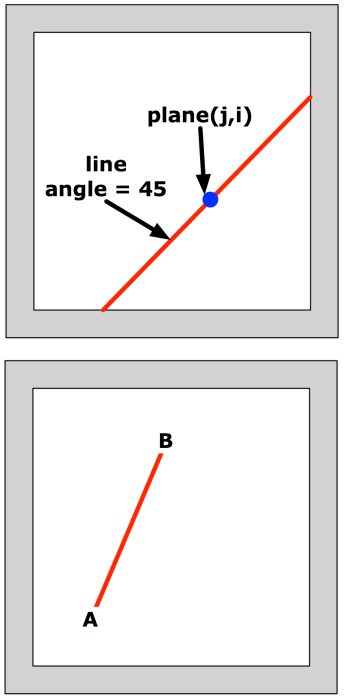
Generate Plots

```
slp = wrf_user_getvar(a,"slp",5)
t2  = wrf_user_getvar(a,"T2",5)
u10 = wrf_user_getvar(a,"U10",5)

v10 = wrf_user_getvar(a,"V10",5)

os@cnLineColor = "NavyBlue"
c_slp = wrf_contour(a,wks,slp,os)
ot@cnFillOn = True
c_tc = wrf_contour(a,wks,t2,ot)
ov@NumVectors = 47
vec = wrf_vector(a,wks,u10,v10,ov)

plot = wrf_map_overlays(a, wks, \
       (/c_tc,c_slp,vec/), pltres, mpres)
```



Special WRF NCL 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_user_intrp3d / wrf_user_intrp2d**
  Interpolate horizontally to a given pressure/height (3d data only)
  Interpolate vertically along a given line
- **wrf_user_ll_to_ij / wrf_user_ij_to_ll**
  Convert: lat/lon ⟺ ij
- **wrf_user_list_times**
  Get list if times available in input file
- **wrf_user_unstagger**
  Unstaggers an array



Special WRF NCL Functions

- **wrf_user_getvar**
  Get fields from input file

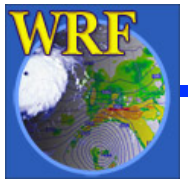
```
ter = wrf_user_getvar(a,"HGT",0)        {ter=a->HGT(0,:,:)}
t2  = wrf_user_getvar(a,"T2",-1)        {t2=a->T2}
slp = wrf_user_getvar(a,"slp",1)
```

avo/pvo: Absolute/Potential Vorticity,
cape_2d: 2D mcape/mcin/lcl/lfc,
cape_3d: 3D cape/cin,
dbz/mdbz: Reflectivity (3D and max),
geopt/geopotential: Geopotential,
p/pres/pressure: Pressure,
rh/rh2: Relative Humidity (3D and 2m),
slp: Sea Level Pressure,
td/td2: Dew Point Temperature (3D and 2m),
tc/tk: Temperature (C and F), th/theta: Potential Temperature,
z/height: Height, ua/va/wa: wind on mass points,
uvmet/uvmet10: wind rotated to earth coordinates (3D and 10m)

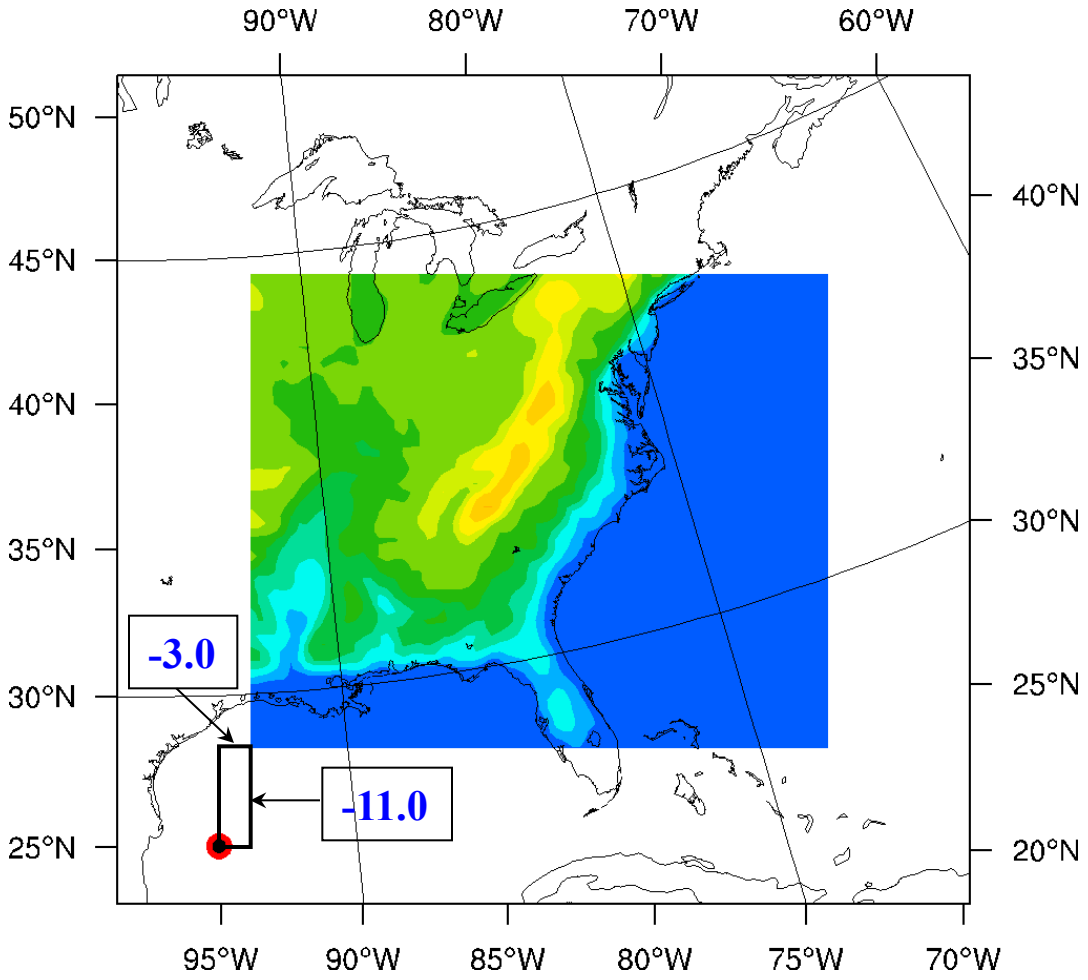
Diagnostics

```
PH      = nc_file->PH(5,:,:,:)
PHB     = nc_file->PHB(5,:,:,:)
   PH     = ( PH + PHB ) / 9.81
   z = wrf_user_unstagger(PH,PH@stagger)

P       = nc_file->P(5,:,:,:)
PB      = nc_file->PB(5,:,:,:)
   P = P + PB

T       = nc_file->T(5,:,:,:)
   T = T + 300.
   tk = wrf_tk( P , T )

QVAPOR = nc_file->QVAPOR(5,:,:,:)
   QVAPOR = QVAPOR > 0.000


slp = wrf_slp( z, tk, P, QVAPOR )
```

```
slp = wrf_user_getvar(nc_file,"slp",5)
```

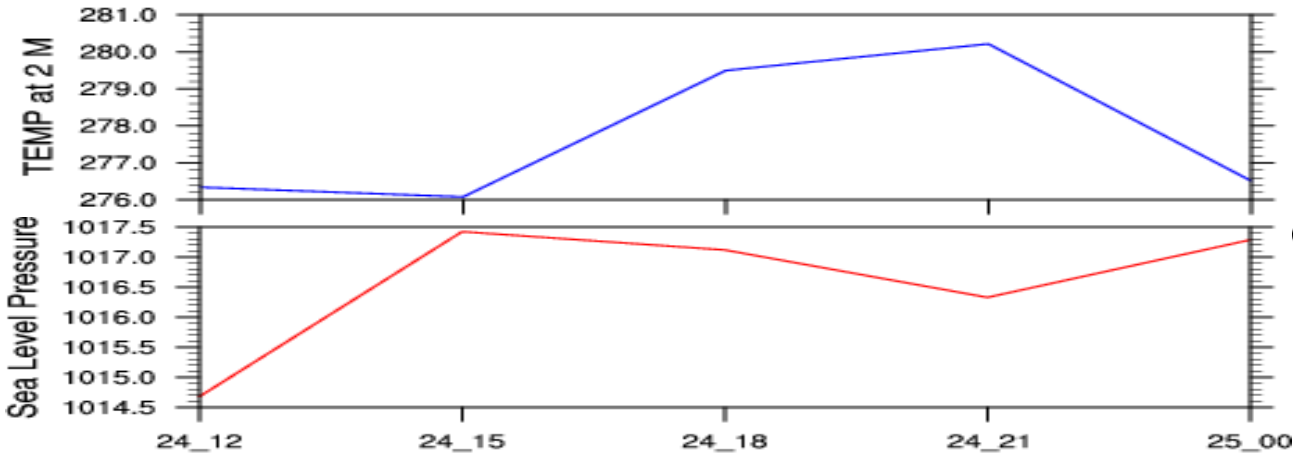


wrf_user_ll_to_ij



Time Series

```
locij = wrf_user_ll_to_ij(a, -87., 32.5, llres)
locij = locij - 1
locX = locij(0)
locY = locij(1)

t2_point  = a->T2(:,locY,locX)
t2_plot   = gsn_csm_xy(wks,taus,t2_point,t2_res)

slp   = wrf_user_getvar(a,"slp",-1)
slp_point  = slp(:,locY,locX)
slp_plot  = gsn_csm_xy(wks,taus,slp_point,t2_res)
```



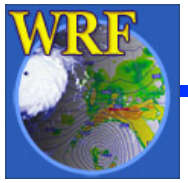
NCL and WRF_NCL

- Combine strength of WRF_NCL specific and NCL general capabilities

```
a = addfile("./wrfout.d01.nc","r")


t2 = a->T2(5,:,:)

t2 = wrf_user_getvar(a,"T2",5)


qv = a->QVAPOR(5,:,:,:)

qv = wrf_user_getvar(a,"QVAPOR",
5)


t2 = a->T2

t2 = wrf_user_getvar(a,"T2",-1)
```

```
res@cnLevelSelectionMode = \
     "ManualLevels"
res@cnMinLevelValF = 0.
res@cnMaxLevelValF = 1000.
res@cnLevelSpacingF = 50.
contour=wrf_contour(a,wks,ter,res)


res@ContourParameters = \
     (/0.,1000.,50./)
contour=wrf_contour(a,wks,ter,res)
```

```
plot = wrf_map_overlays \
  (a,wks,(/contour/),pltres,mpres)


mpres@mpGridSpacingF = 45
 plot = wrf_map_overlays \
  (a,wks,(/contour/),pltres,mpres)
```

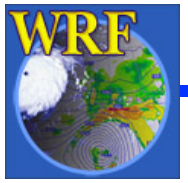


Resources

- **The special WRF functions have unique resources:**
  http://www.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/NCL_functions.htm

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

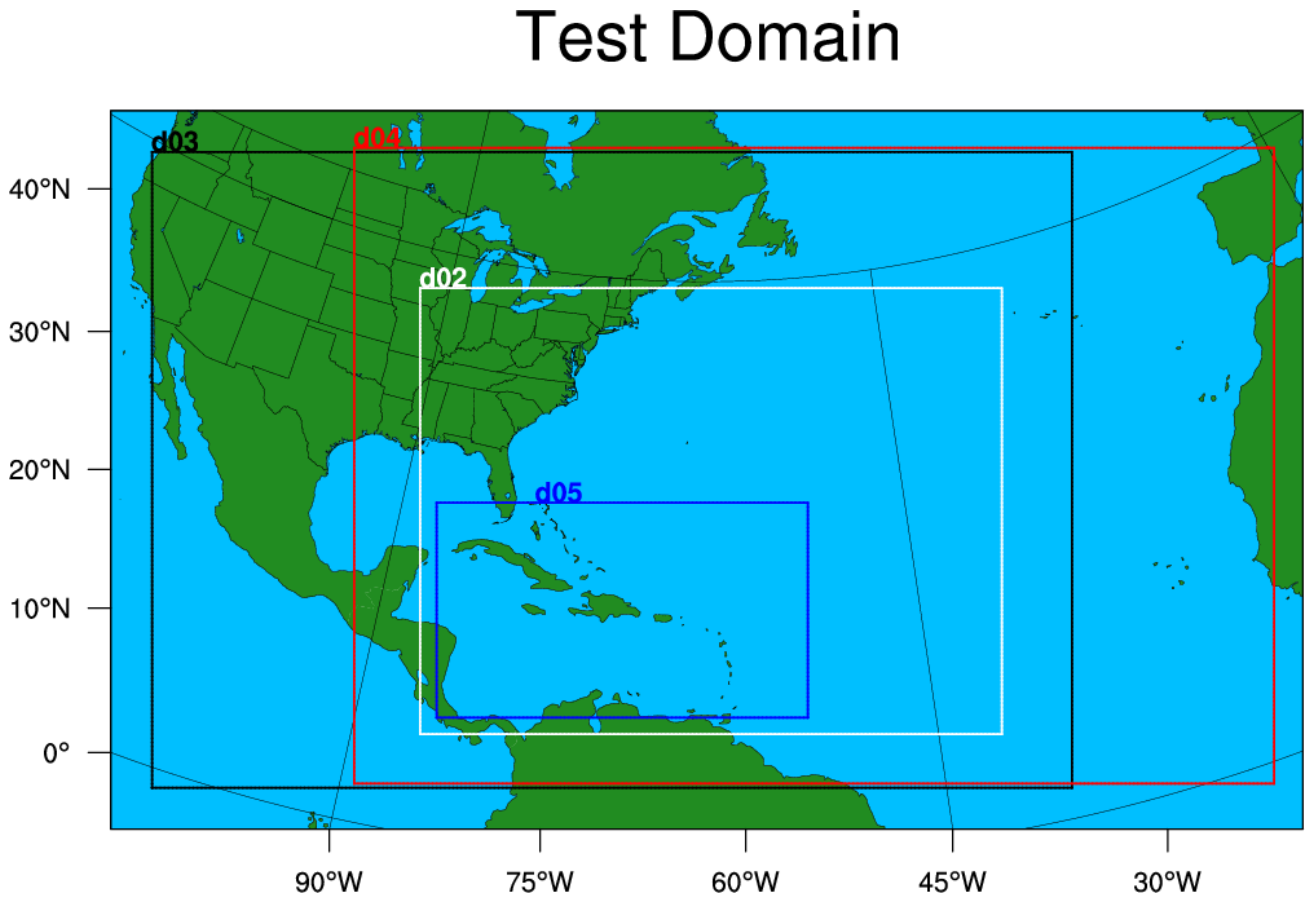
- am (annotation manager)
- app (app)
- ca (coordinate array)
- cn (contour)
- ct (coordinate array table)
- dc (data comm)
- err (error)
- gs (graphic style)
- gsn (gsn high-level interfaces)
- lb (label bar)
- lg (legends)
- mp (maps)
- pm (plot manager)
- pr (primitives)
- sf (scalar field)
- st (streamline)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



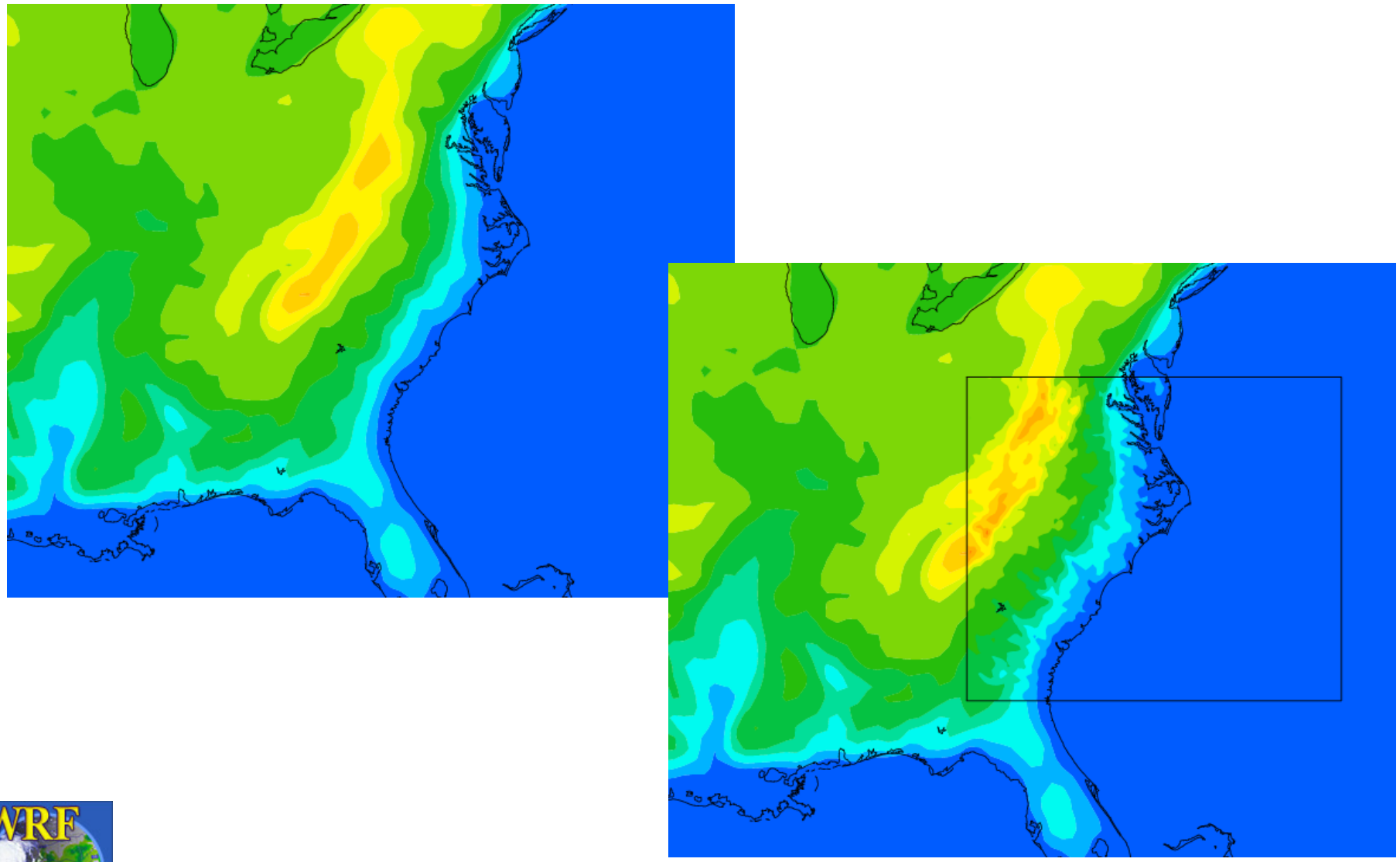
Domain Design

mp = wrf_wps_dom (wks, mpres, lnres, txres)

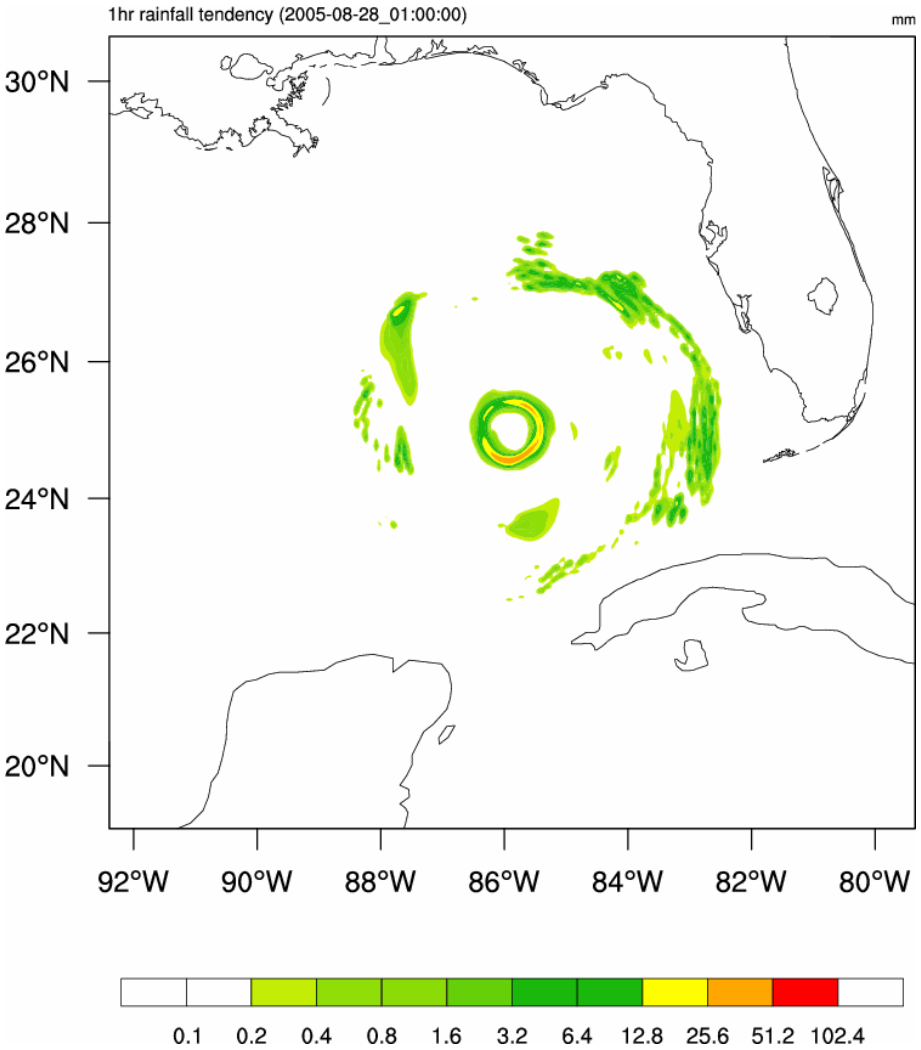
WPS/util/plotgrids.ncl



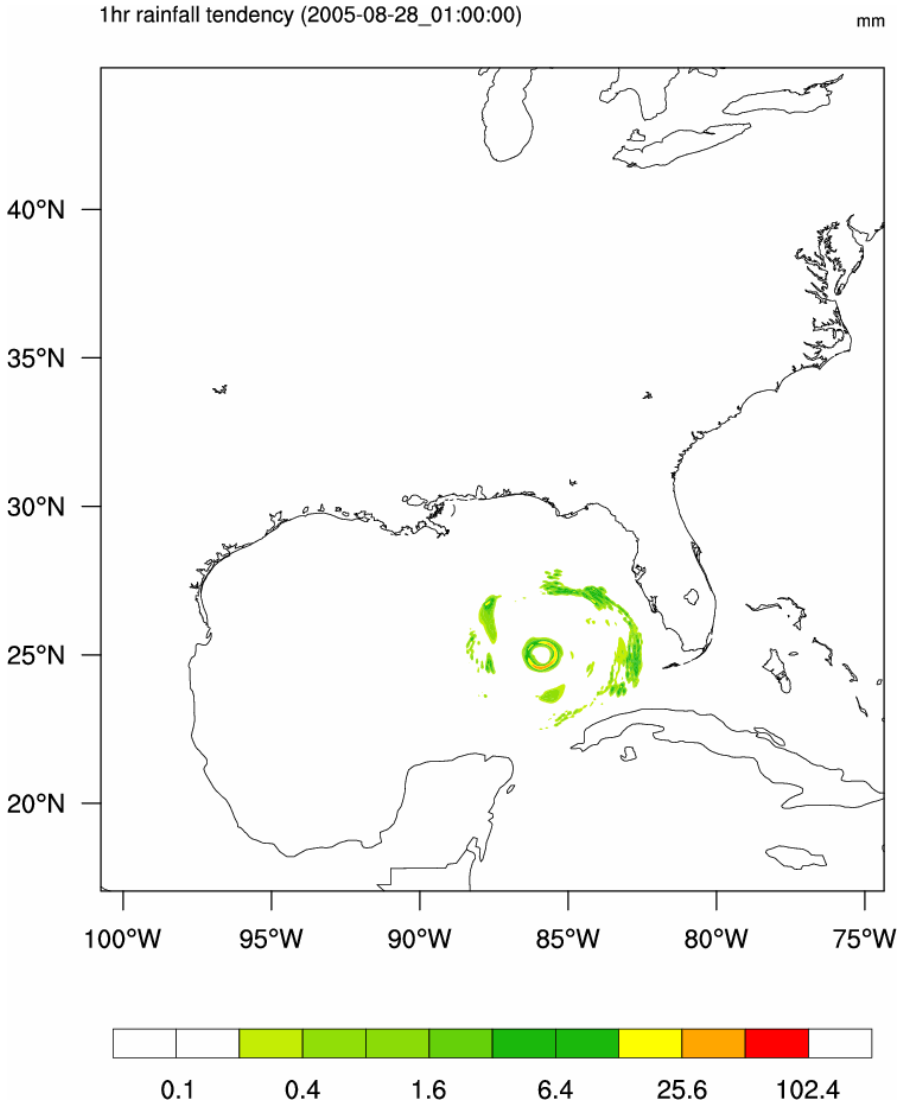
Over Domains



Moving Nests



Moving Nests

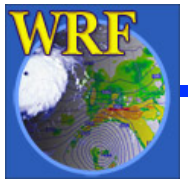


Calling FORTRAN code from NCL

- Easier to use F77 code, but works with F90 code
- Need to isolate definition of input variables and wrap it with special comment statements:

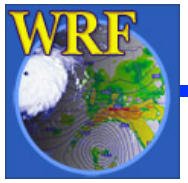
```
C NCLFORTSTART
C NCLEND
```

- Use a tool called `WRAPIT` to create a `*.so` file
- Load `*.so` file in NCL script with "`external`" statement
- Call Fortran function with special "`::`" syntax
- Must pre-allocate arrays!



myTK.f

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

        do k=1,nz
          do j=1,ny
          do i=1,nx
             pi=(pressure(i,j,k) / 1000.)**(287./1004.)
             tk(i,j,k) = pi*theta(i,j,k)
          enddo
          enddo
        enddo

        end
```

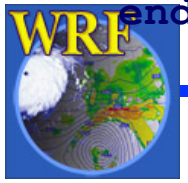


myTK.so - *Create & use in NCL script*

```
% WRAPIT myTK.f
```

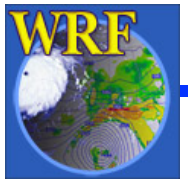
This will create a "myTK.so" file

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

begin
    t = wrf_user_getvar(a,"T",5)

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

    ; Must preallocate space for output arrays
    dim = dimsizes(t)
    tk = new( dimsizes(t), typeof(t) )

    ; Remember, Fortran/NCL arrays are ordered differently
        myTK :: compute_tk (tk,p,t,dim(2),dim(1),dim(0))
end
```



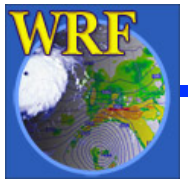
FORTRAN 90 code

- Can use simple FORTRAN 90 code
- Your FORTRAN 90 program may not contain any of the following features:
  - pointers or structures as arguments,
  - missing/optional arguments,
  - keyword arguments, or
  - recursive procedure.

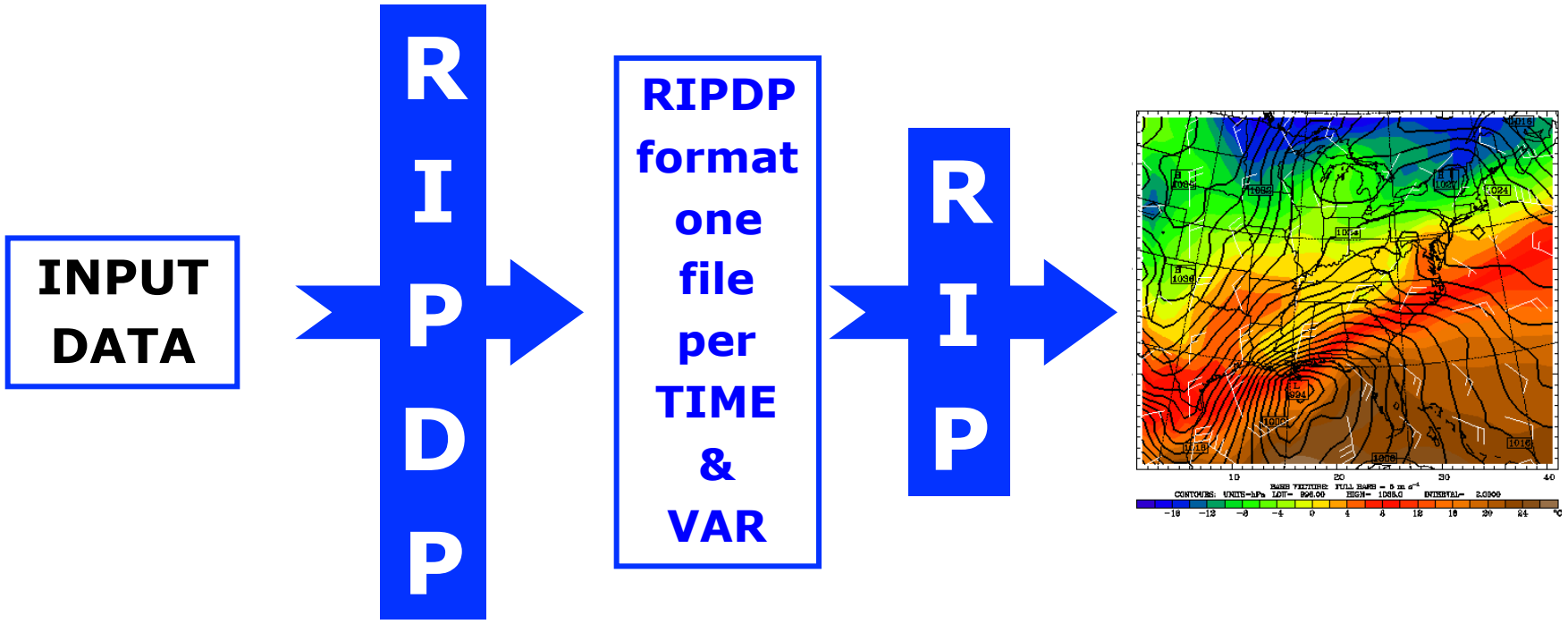


RIP4

- Read Interpolate Plot version 4
- Develop by Mark Stoelinga (3TIER/UW/NCAR) & MMM/NCAR Staff
- Originally developed for the MM5 model
- Generate a number of graphical plots
  - Horizontal, cross-section, skewT
- Current Version: 4.6
  - configure / compile

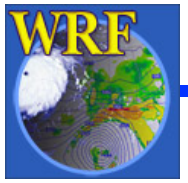


RIP4



RIP4 General Information

- Requires NCL Version 5 or NCAR Graphics low-level routines
  - *http://www.ncl.ucar.edu* *(Recommended)*
  - *http://ngwww.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 *(/usr/local/ncarg)*

- **Configure**
  ./configure
  (check configure.rip to ensure netCDF paths are correct)

- **Compile**
  ./compile

- **RIP4 has 2 parts (RIPDP and RIP)**

Compiling with ncl

ncl myscript.ncl

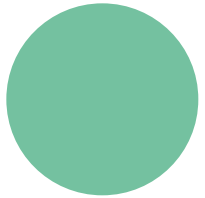
NCARG_ROOT

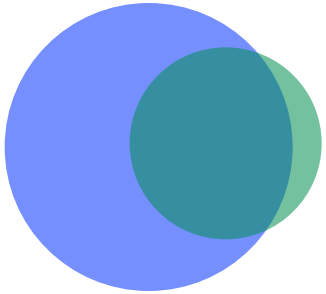
/usr/local/ncl

compile <util>

NCARG_ROOT

/usr/local/ncarg





ncl myscript.ncl

compile <util>

NCARG_ROOT

/usr/local/ncl

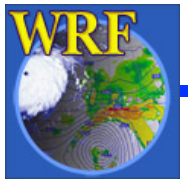
Compiling with ncl

```
In function `write_png':
  undefined reference to `png_create_write_struct'
  undefined reference to `png_create_info_struct'
  undefined reference to `png_destroy_write_struct'
  undefined reference to `png_destroy_write_struct'
```

-L<path_to_png_lib> -lpng -L<path_to_z_lib> -lz

```
/usr/local/ncl/lib/libncarg.a(agcurv.o): In function `agcurv_':
agcurv.f:(.text+0x69): undefined reference to `_gfortran_copy_string'
/usr/local/ncl/lib/libncarg.a(aggtch.o): In function `aggtch_':
aggtch.f:(.text+0x3e): undefined reference to `_gfortran_copy_string'
aggtch.f:(.text+0x7b): undefined reference to `_gfortran_copy_string'
```

-L<path_to_gfortran_lib> -lgfortran

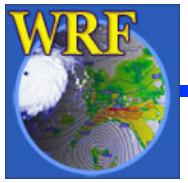


ripdp & rip

- **ripdp**
  - RIP Data Preparation
  - Converter
    RIPDP converts WRF formatted data (*netCDF*) into RIP format (*B – grid*)
  - Output
    RIPDP puts each Variable at each Time into a separate file
    Creates LOTS of files

mkdir RIPDP

- **rip**
  - Reads the output generated by *ripdp*
  - Makes use of a User Input File (UIF) *(rip_sample.in)* to control plots
  - Output: X11, pdf, ps, cgm



Running ripdp & rip

ripdp_wrf*xxx* [-n namelist-file]
<model_data_name> \ [basic/all]
<input_file(s)>

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

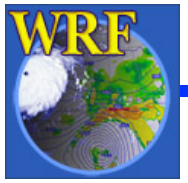
Example:

ripdp_wrfarw RIPDP/arw all wrfout*

rip [-f] RIPDP/arw rip_sample.in

output

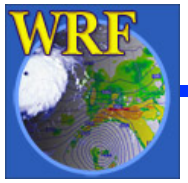
[rip_sample.out]
rip_sample.TYPE



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

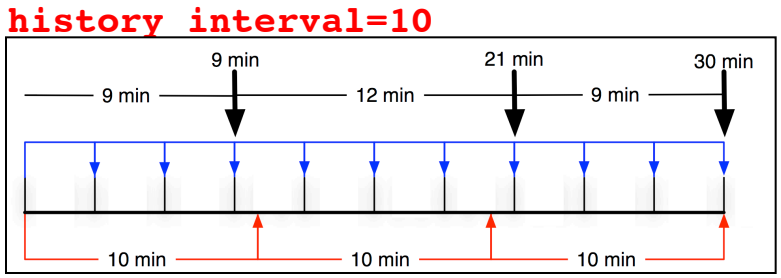
```
GKS ERROR NUMBER   2 ISSUED FROM SUBROUTINE
GCLKS :--GKS NOT IN PROPER STATE: GKS SHALL BE IN
STATE GKOPFORTRAN STOP
```

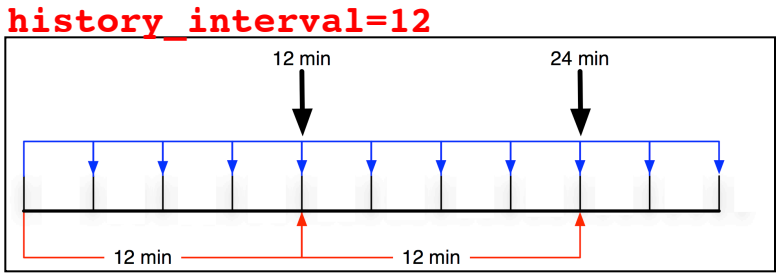


RIP4 namelist

- **ptimes** *(times for ripdp to process)*
  - `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* [`time_step=180` (3 min)]



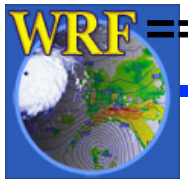


RIP4 User Input File

```
&userin
    .......
&end
&trajcalc
    .......
&end
==========================================
----------        Plot Specification Table        ----------
==========================================
feld= .....
feld= .....
==========================================
feld= .....
feld= .....
==========================================
```

Frame (overlay) of a number of fields

Plot a specific field

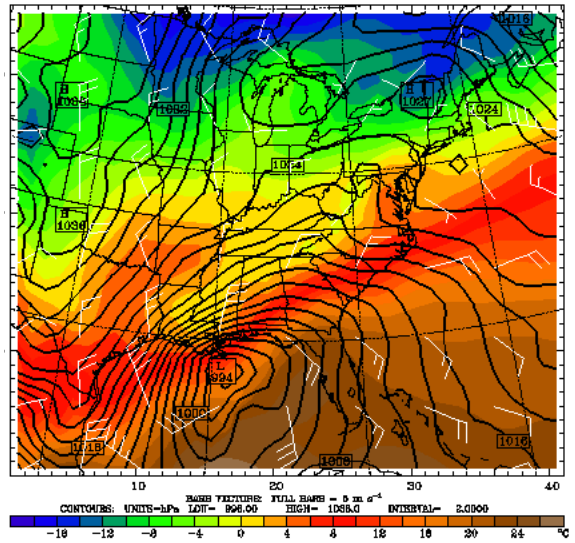


Creating a Plot with RIP4

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

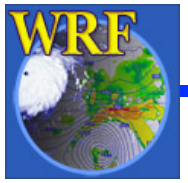
vcor=s; **levs=2fb**
vcor=s; **levs=1,2,3**
vcor=p; **levs=800,500**
vcor=p; **levs=800,-300,100**

```
==========================================
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
==========================================
```

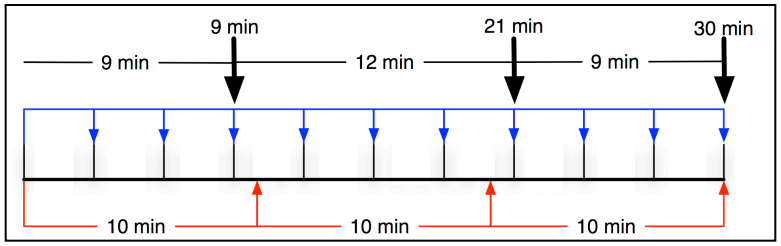


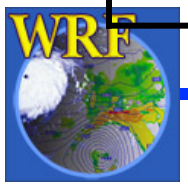
ARWpost - *converter*

- **Download Code** (http://www.mmm.ucar.edu/wrf/users)
- **OnLine Tutorial**
  http://www.mmm.ucar.edu/wrf/users/graphics/ARWpost/ARWpost.htm
- **For GrADS output**
  - GrADS libraries only needed to display data *(freely available)*
  - http://grads.iges.org/grads/grads.html
- **Version 2 (old – not recommended)**
  - Could produce vis5d output
  - Needed WRFV3 complied



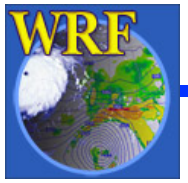
namelist.ARWpost

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



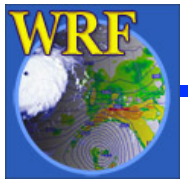
namelist.ARWpost

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

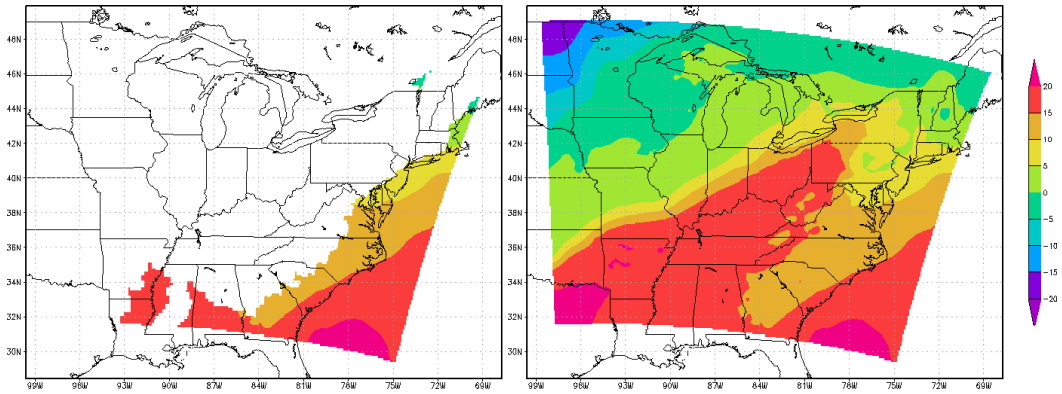


namelist.ARWpost

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	



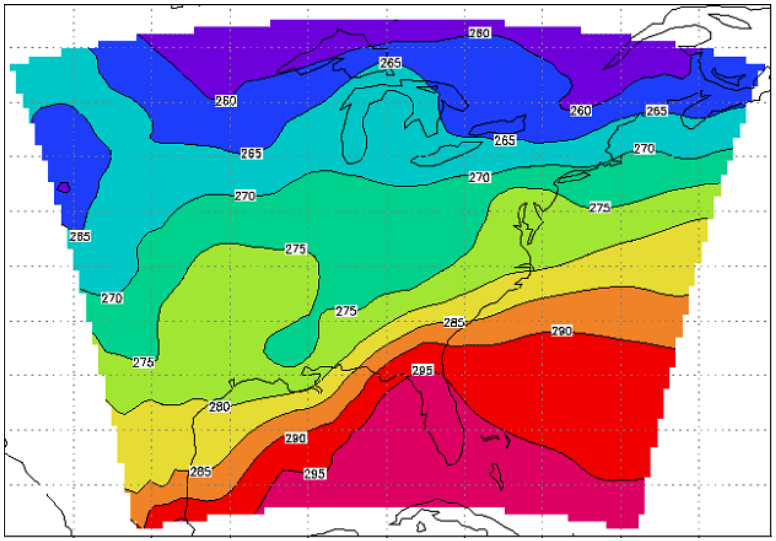
namelist.ARWpost

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 specific notes

- **To display images**
  - Requires GrADS software
  - Freely available from: http://grads.iges.org/grads/grads.html
  - Documentation: http://grads.iges.org/grads/gadoc/index.html

- **Projection**
  - Data is plotted on model projection

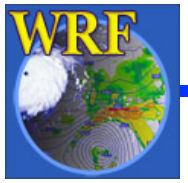


GrADS - .ctl file

```
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
```

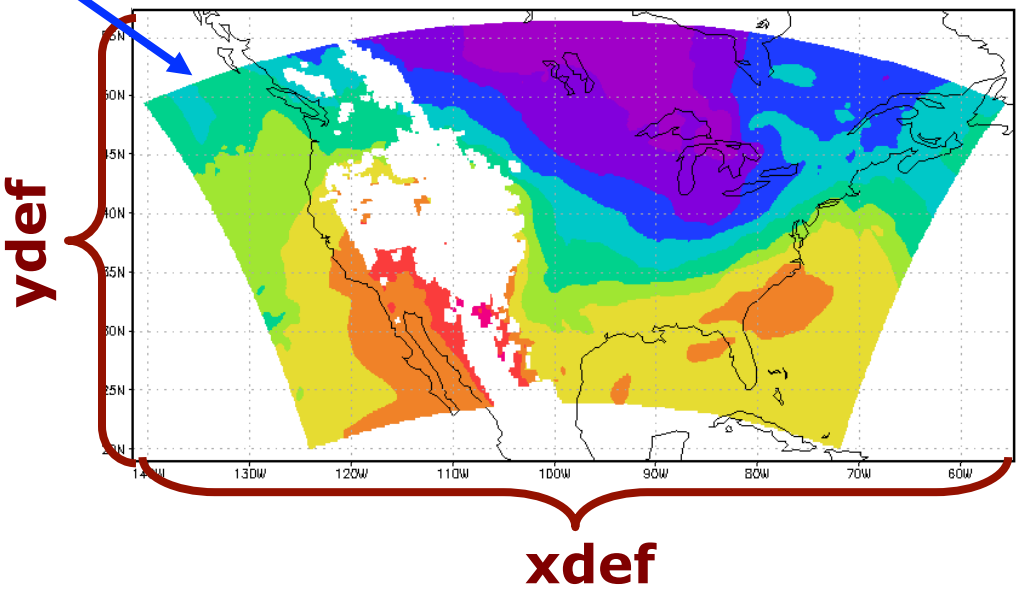
options byteswapped

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



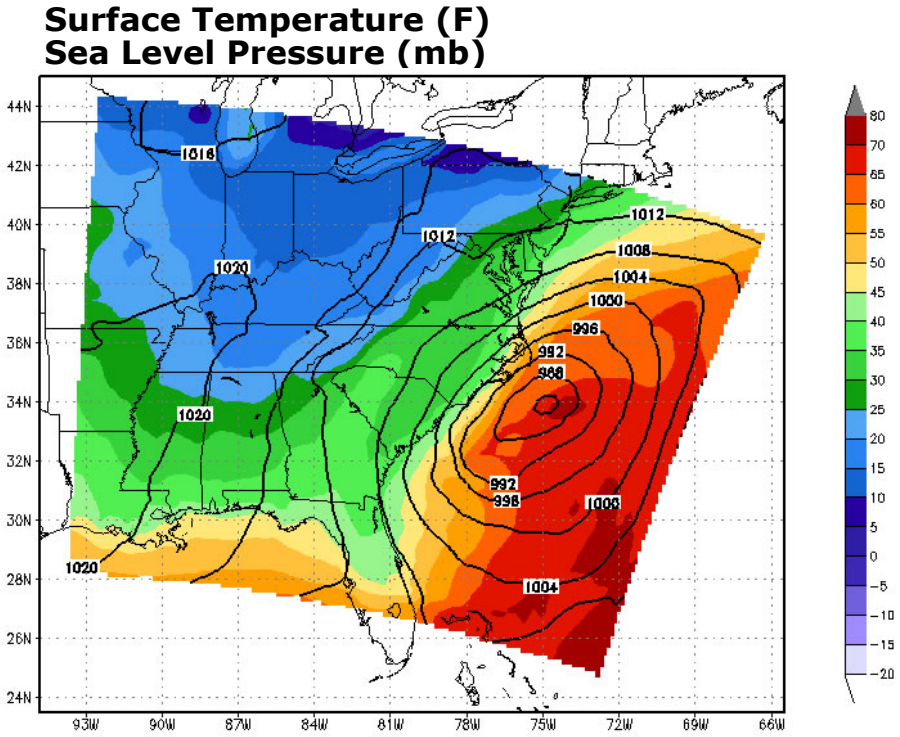
GrADS - .ctl file

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



Creating a Plot

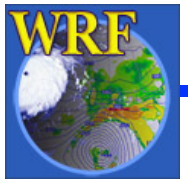
```
open em_real.ctl
set mpdset hires
set display color white

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
```



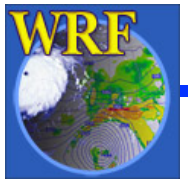
How to add diagnostics

- **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



Other Post-processing Tools

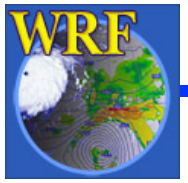


VAPOR

Computational and Information Systems Laboratory
National Center for Atmospheric Research

Visualization and Analysis Platform for Oceanic, atmospheric and solar Research

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



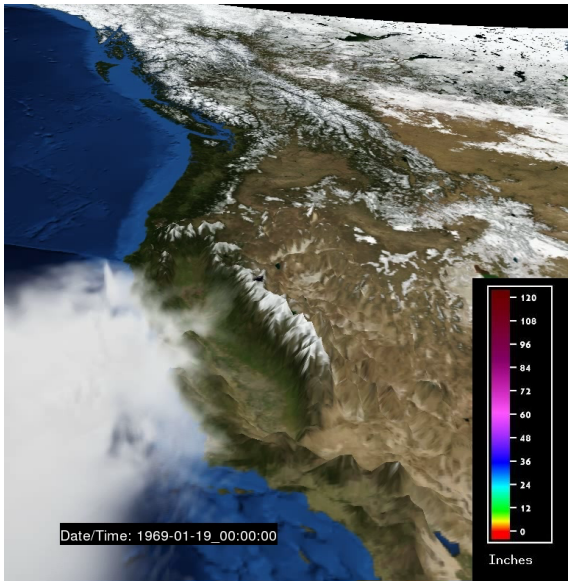
WRF in VAPOR

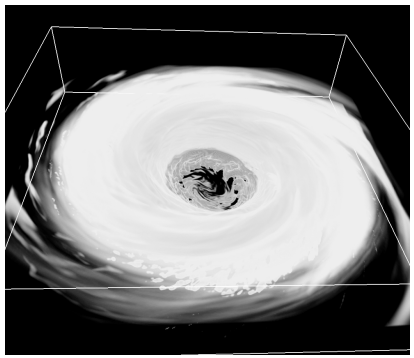
Computational and Information Systems Laboratory
National Center for Atmospheric Research

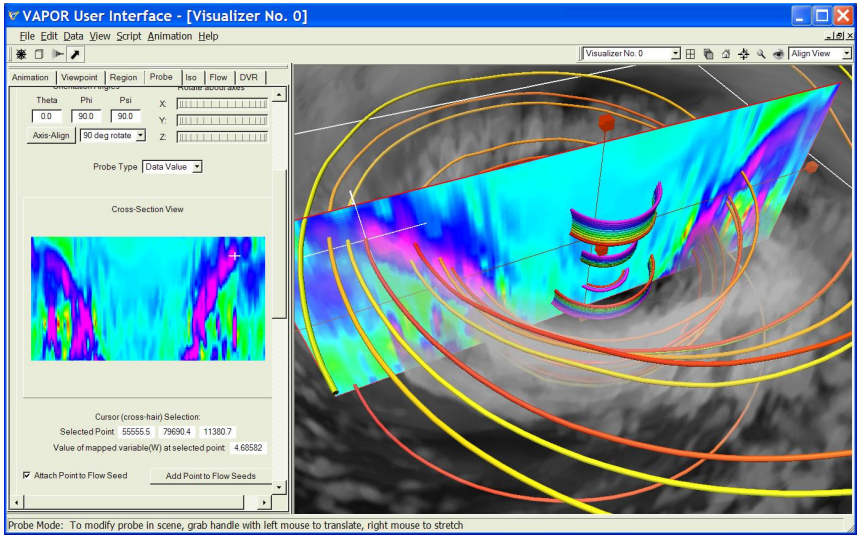
- Interactive 3D visualization of WRF-ARW data *(wrfout files only)*
- Available free on Linux, Windows, Mac
- Interactive rendering and animation (using GPU acceleration)
- Simple 2-step data conversion from WRF output to VAPOR
  - wrfvdfcreate & wrf2vdf
- Steady and unsteady flow integration
- Data probing
- Downloads, documentation, examples at: http://www.vapor.ucar.edu
- http://www.vapor.ucar.edu/doc/WRFsupport.pdf

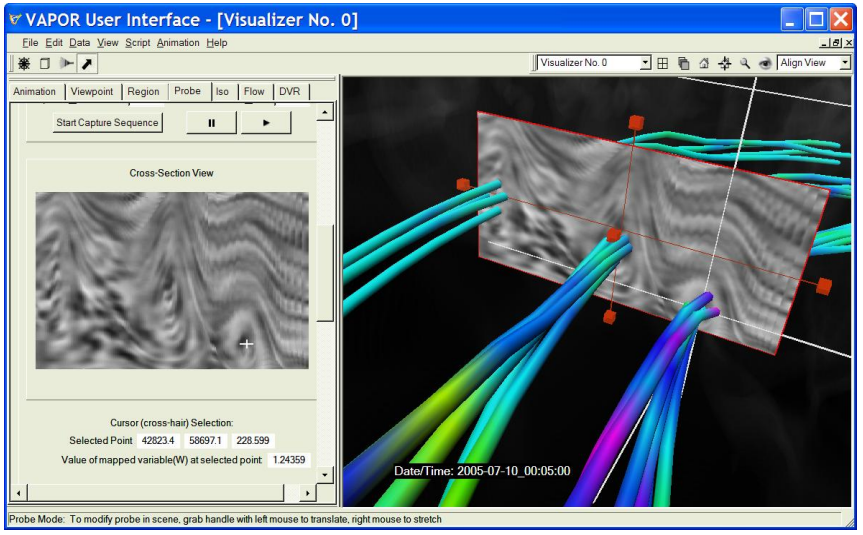
WRF in VAPOR

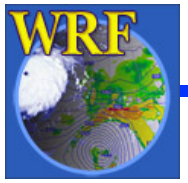
Computational and Information Systems Laboratory
National Center for Atmospheric Research











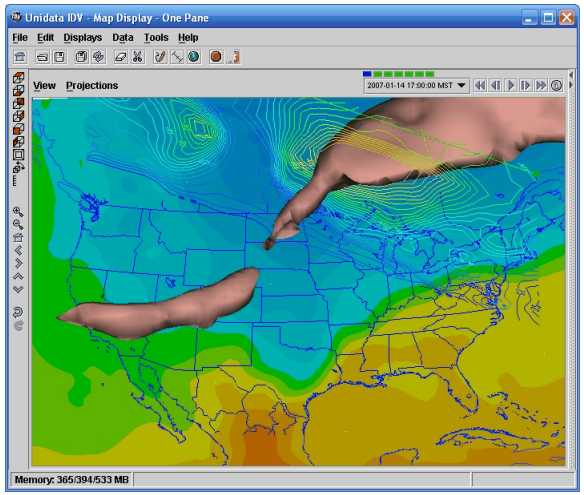
IDV

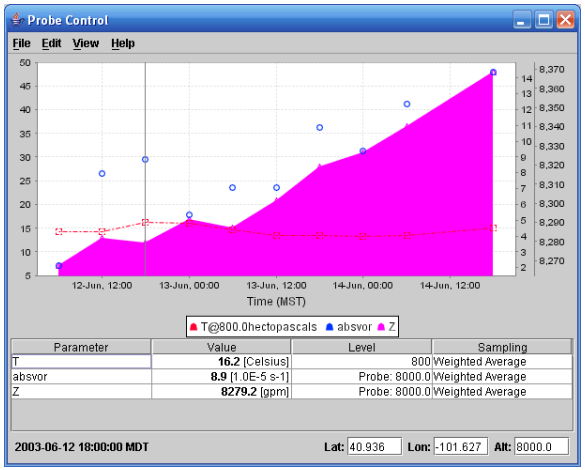
Integrated Data Viewer

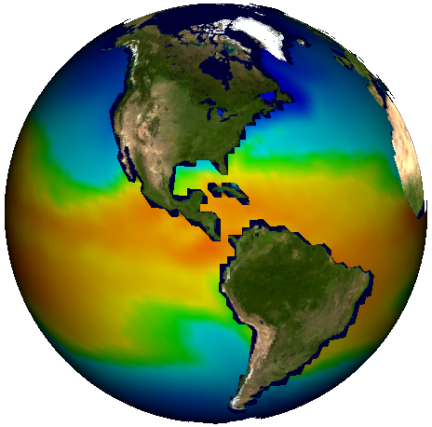
Unidata Program Center/UCAR

What is the IDV?

- Visualization and analysis tool for geoscience data developed and supported by Unidata
- Freely available Java™ framework and application
- Integrated 2D/3D displays of a wide range of data
- Built on VisAD library

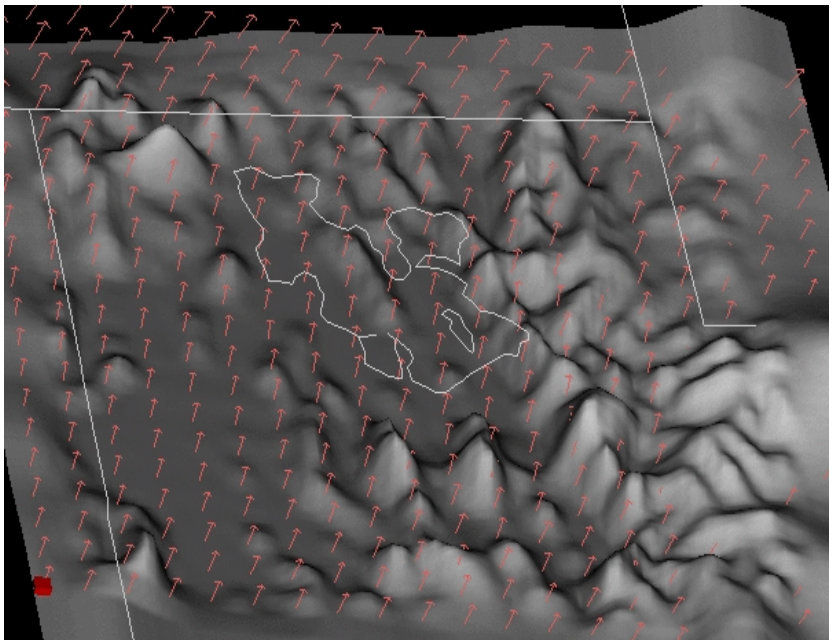






IDV Strengths

- Easy to download and install on any platform
- Remote and local access to datasets
- 2D/3D visualization
- Bundle mechanism
- Support for multi-disciplinary datasets integrated from a variety of sources
- Flexible framework supports customization (GEON-IDV, field projects, McIDAS-V)
- Extensive documentation
- Community driven development



Model simulation of wind, isentropic potential vorticity and low level moisture flow over the Great Salt Lake basin

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

For Further Information

- Integrated Data Viewer homepage
  - http://www.unidata.ucar.edu/software/idv
- RAMADDA homepage
  - http://www.unidata.ucar.edu/software/ramadda/
- VisAD homepage
  - http://www.ssec.wisc.edu/~billh/visad.html
- All IDV questions/comments
  - support-idv@unidata.ucar.edu