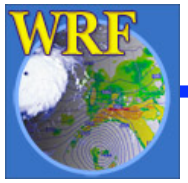
Post-processing Tools

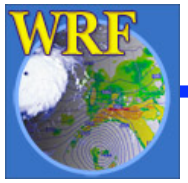
Cindy Bruyère



Graphical Packages

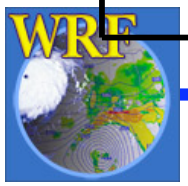
- **NCL** — UG: 9-2
  - Graphical package
- **ARWpost** — UG: 9-28
  - Converter (GrADS & vis5d)
- **RIP4** — UG: 9-19
  - Converter and interface to graphical package NCAR Graphics
- **WPP** — UG: 9-35
  - Converter (GrADS & GEMPAK)
- **VAPOR** — UG: 9-50
  - Converter and graphical package
  - *Support: 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



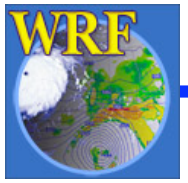
Graphical Packages

	NCL	RIP4	ARWpost (GrADS / Vis5D)	WPP	VAPOR
Directly ingest WRF data	***Y***	**N** *converter*	**N /** ***(Y)*** *converter*	**N** *converter*	**N** *converter*
Intermediate files	***N***	lots	large file	**Y**	large file
WPS DATA	***Y***	***Y***	***Y***	**N**	**N**
wrfinput data	***Y***	***Y***	***Y***	**N**	**N**
Idealized data files	***Y***	***Y***	***Y***	**N**	**N**
Input format	*netCDF*	*netCDF*	*netCDF / GRIB1*	*netCDF / binary*	*netCDF*
Vertical Output Coordinate	eta pressure height	eta pressure height	eta pressure height	pressure	eta
Software required ***(All binaries are free)***	*NCL*	*NCARG*	*GrADS/vis5d*	*GrADS / GEMPAK*	*VAPOR*
Diagnostics	*some*	*> 100*	*some*	*some*	*limited*



RIP4

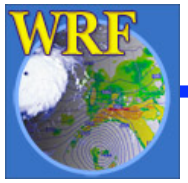
- **Read Interpolate Plot version 4**
- **Develop by Mark Stoelinga (3TIER/UW/NCAR) & MMM/NCAR Staff**
- **Originally developed for the MM5 model**
- **Generate a number of graphical plots**
  - Horizontal, cross-section, skewT
- **Current Version: 4.6**
  - configure / compile



RIP4 - General Information

- **Documentation**
  - In program tar file under the Doc/ directory
  - ***http://www.mmm.ucar.edu/wrf/users/docs/ripug.htm***
  - ***http://www.dtcenter.org/wrf-nmm/users/docs/user_guide/RIP/ripug.htm***

- **OnLine Tutorial:**
  - ***http://www.mmm.ucar.edu/wrf/users/graphics/RIP4/RIP4.htm***
  - ***http://www.dtcenter.org/wrf-nmm/users/OnLineTutorial/NMM/RIP/index.php***

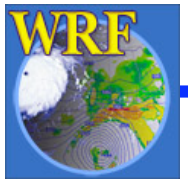


RIP4 - General Information

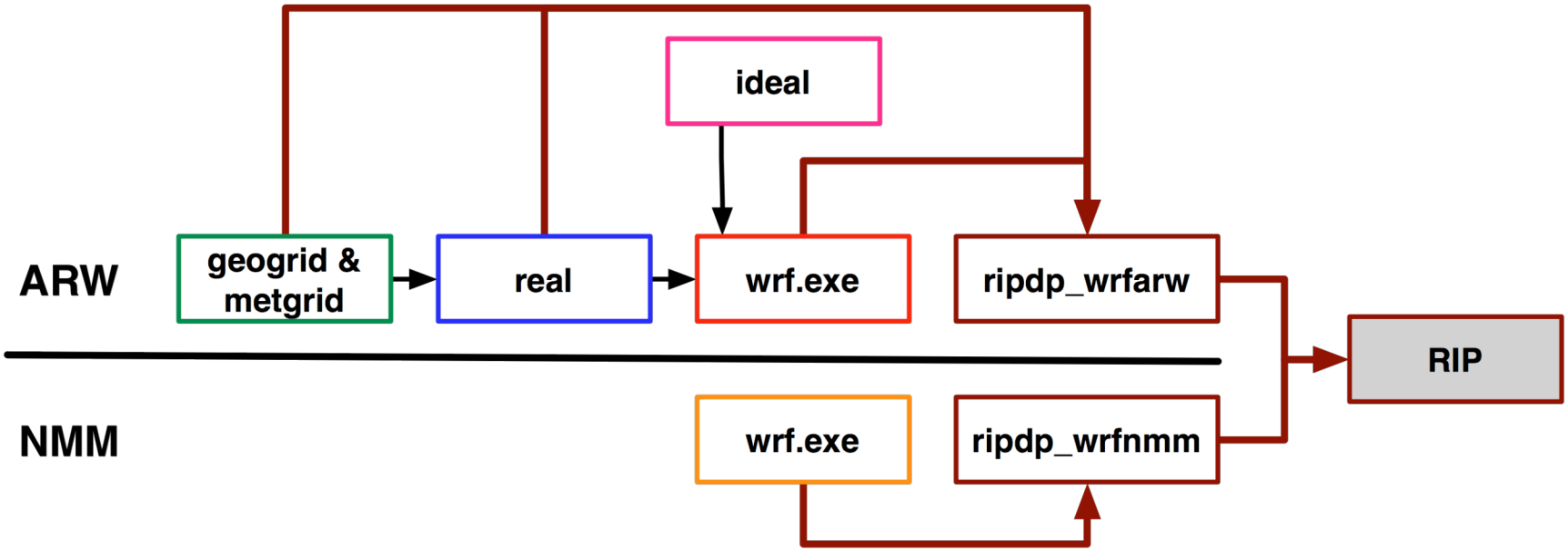
- **Requires NCAR Graphics low-level routines**
  - ***http://ngwww.ucar.edu***

- **NCL Version 5:**
  - ***http://www.ncl.ucar.edu***
  - Released November 2007
  - Combine NCL and NCAR Graphics
  - Open Source
  - Recommended

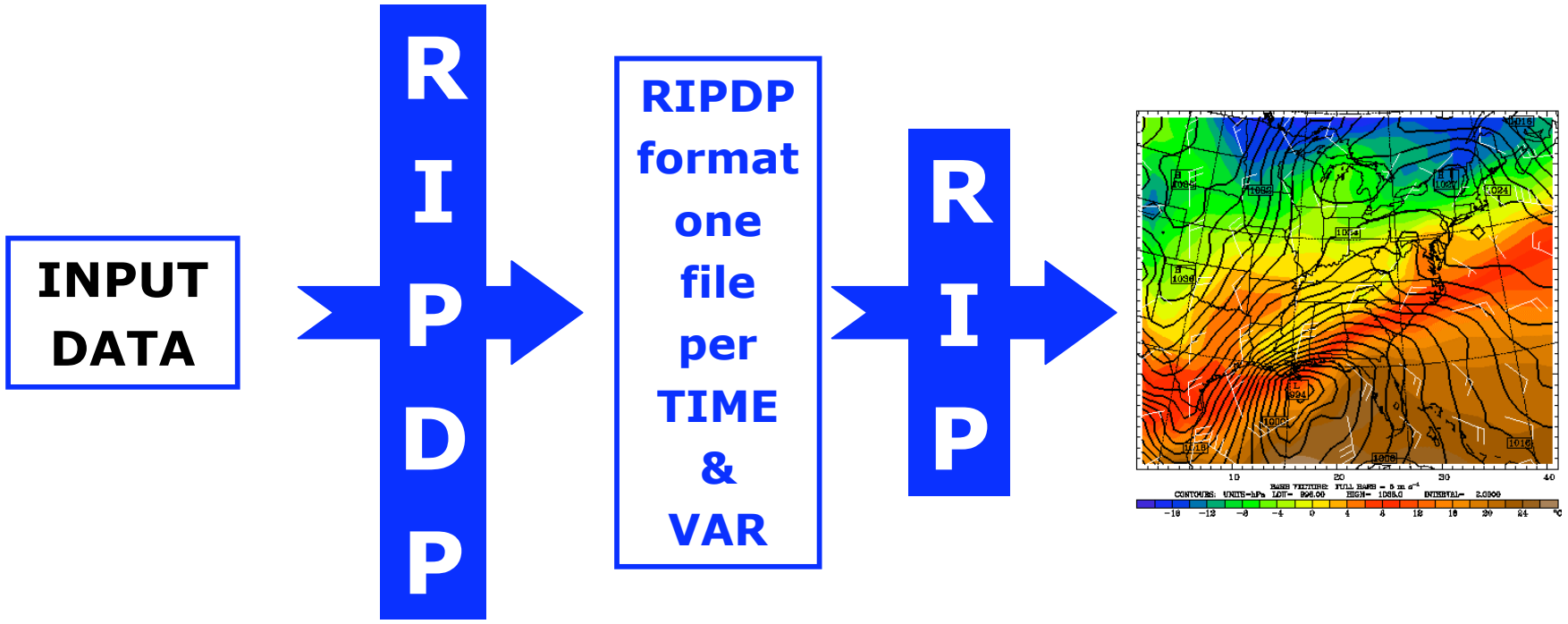
- **Download Code:**
  - ***http://www.mmm.ucar.edu/wrf/users/download/get_source.html***
  - ***http://www.dtcenter.org/wrf-nmm/users/downloads/index.php***



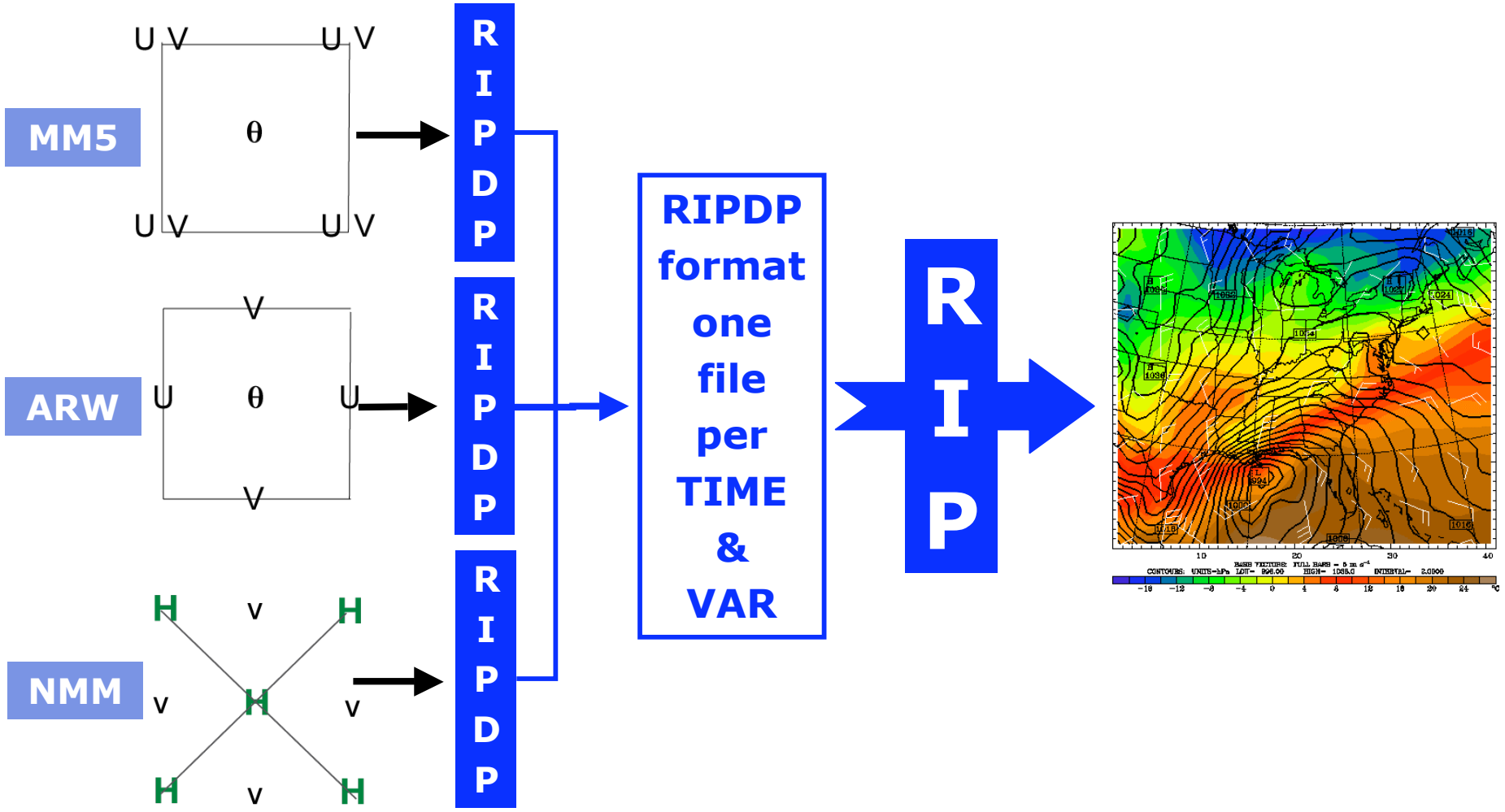
RIP4 Input Data



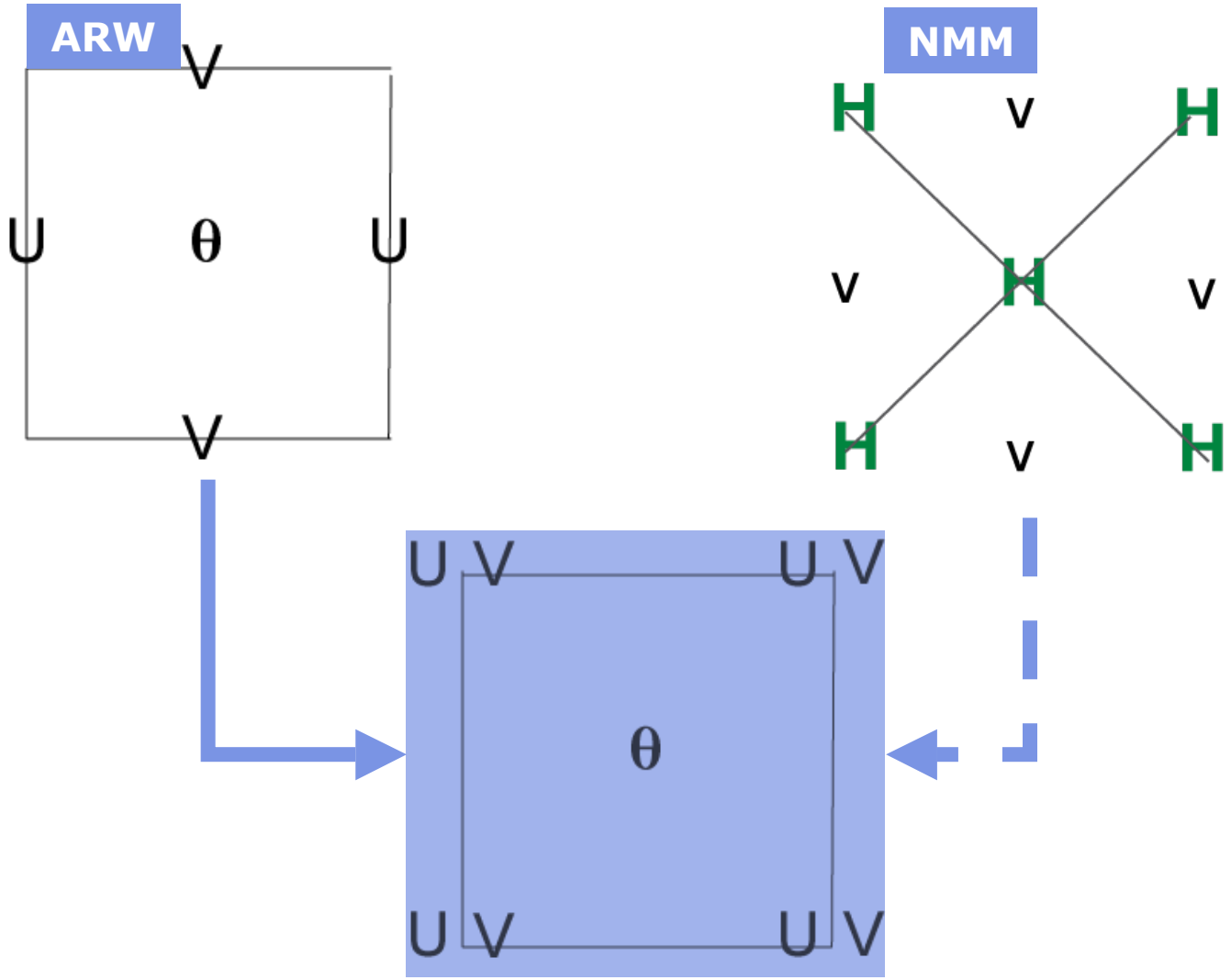
RIP4



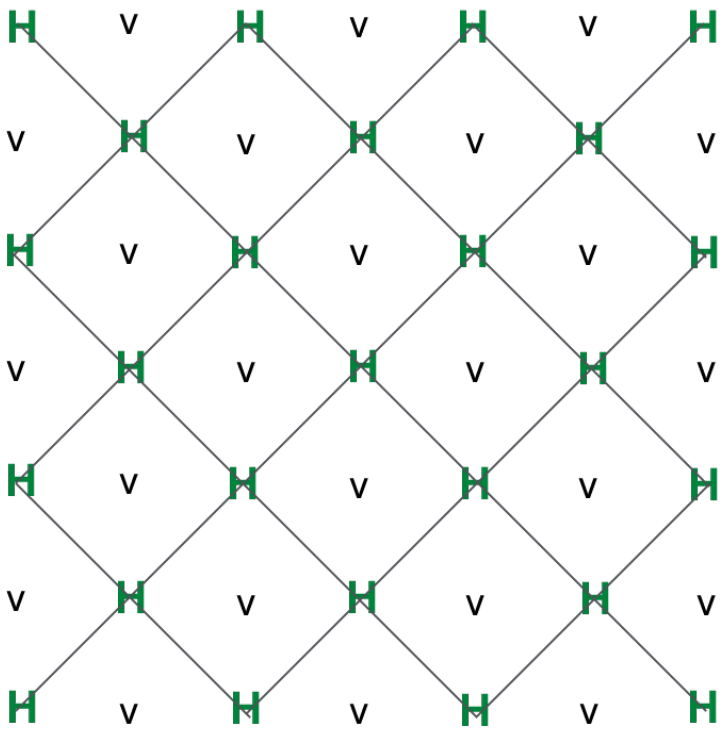
RIP4 - Grids

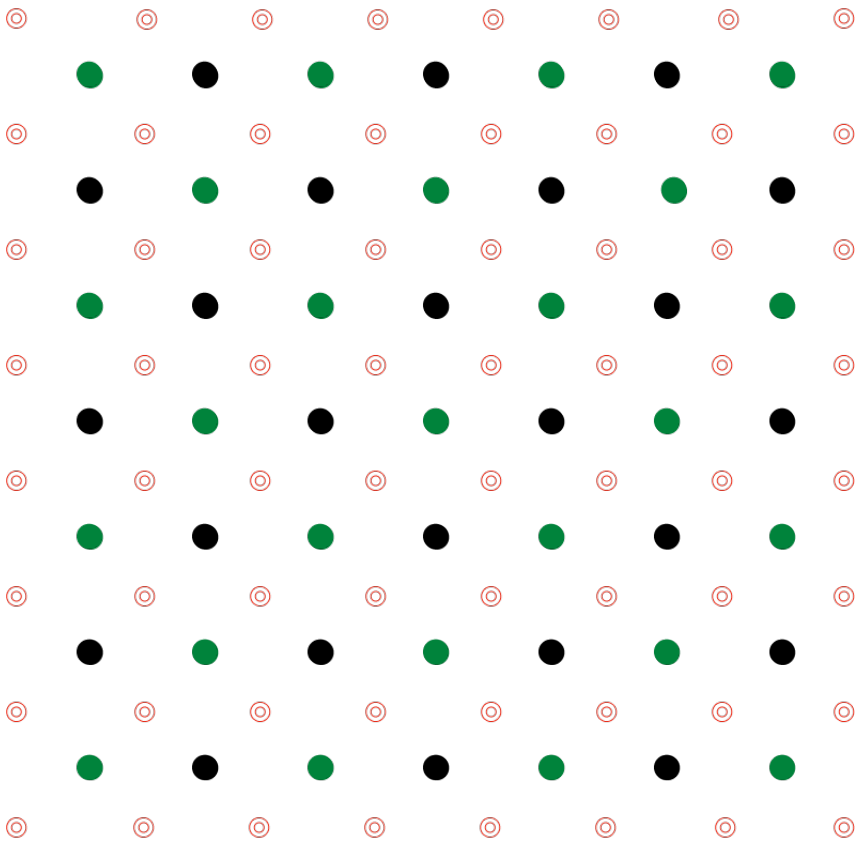

MM5
U V
θ
U V
U V
U V
ARW
V
U
θ
U
V
NMM
H
v
H
v
H
v
H
v
H
RIPDP
RIPDP
RIPDP
RIPDP format one file per TIME & VAR
RIP

RIP4 - WRF Grids

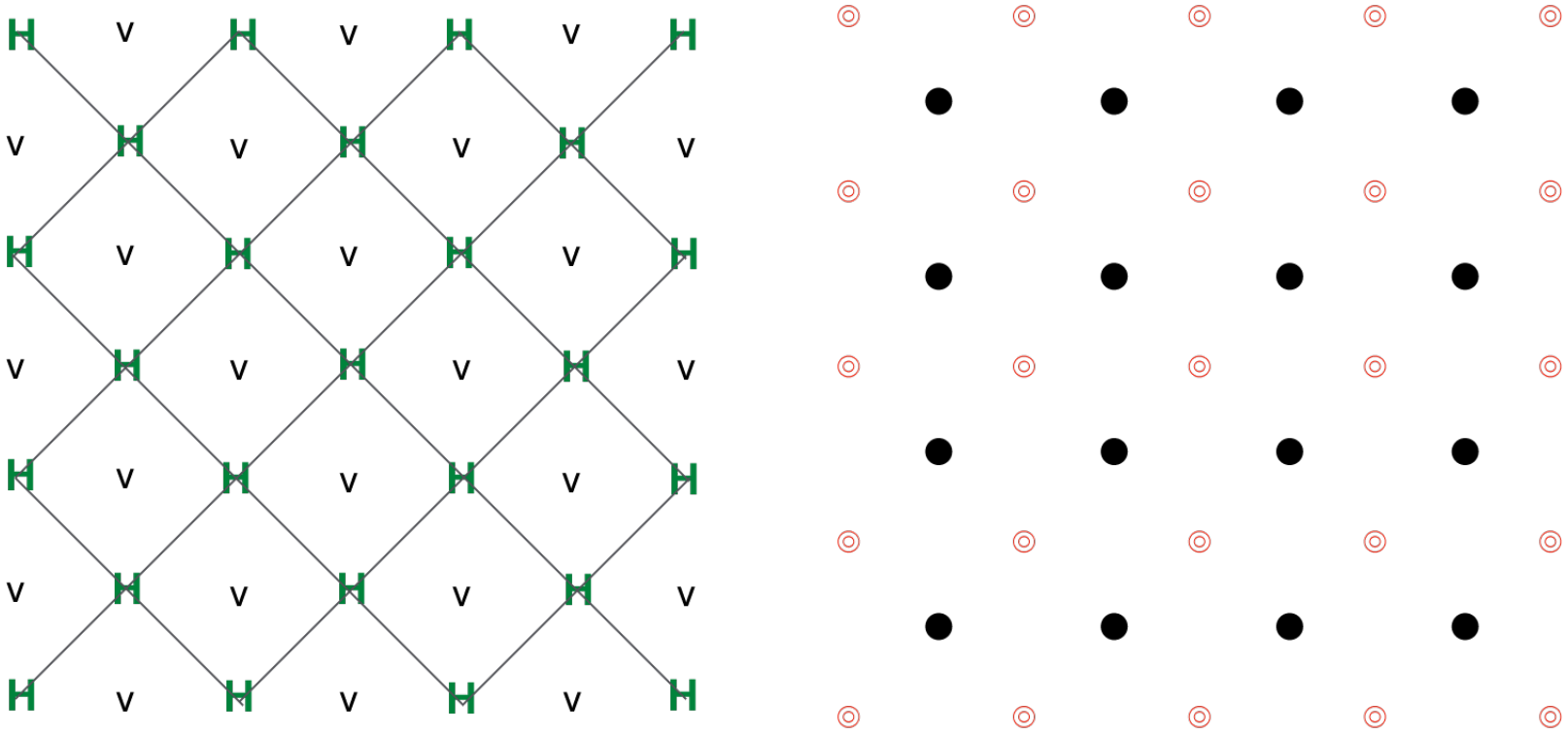


RIP4 - NMM Grid *(iinterp 0)*





RIP4 - NMM Grid *(iinterp 1)*



new projection ; no direct relationship

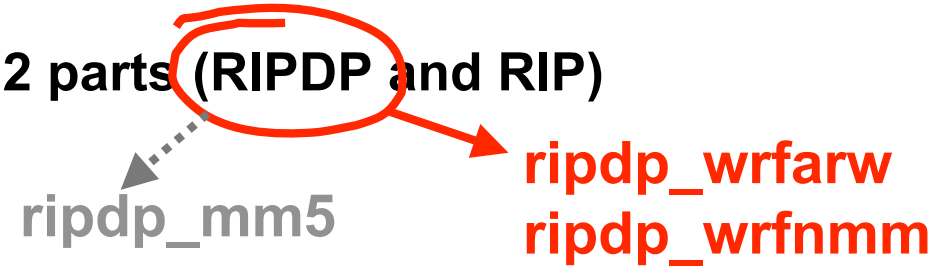
RIP4 on your computer

- **set environment variables**
  setenv RIP_ROOT /usr/$USER/RIP4 ***(rip_root)***
  setenv NCARG_ROOT /usr/local/ncl ***(/usr/local/ncarg)***

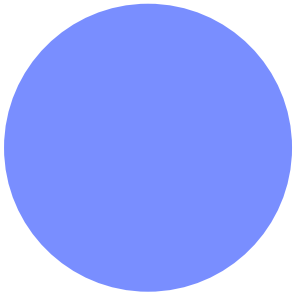
- **Configure**
  ./configure
  (check configure.rip to ensure netCDF paths are correct)

- **Compile**
  ./compile

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



Compiling with ncl



ncl myscript.ncl

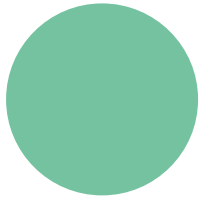
NCARG_ROOT

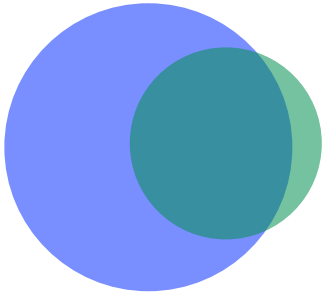
/usr/local/ncl

compile <util>

NCARG_ROOT

/usr/local/ncarg





ncl myscript.ncl

compile <util>

NCARG_ROOT

/usr/local/ncl

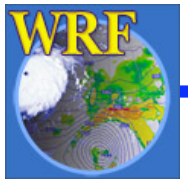
Compiling with ncl

```
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'
```

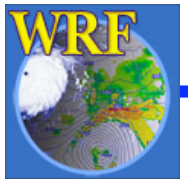
-L<path_to_gfortran_lib> -lgfortran



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

- **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

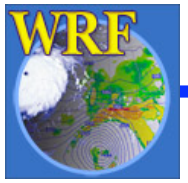
Example:

ripdp_wrfarw RIPDP/test all wrfout*

rip [-f] RIPDP/test rip_sample.in

output

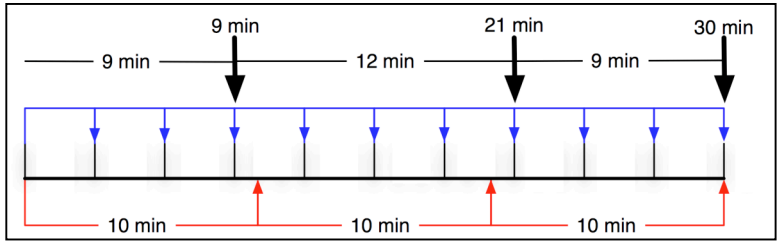
[rip_sample.out]
rip_sample.TYPE



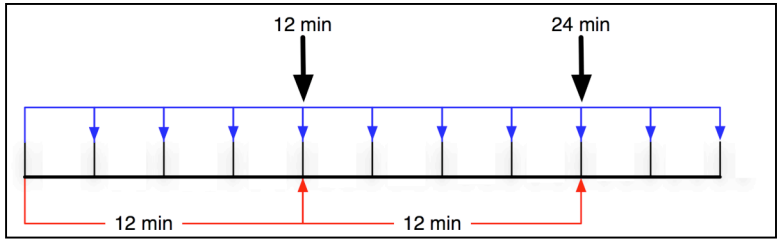
ripdp namelist

- **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)]

`history_interval=10`



`history_interval=12`



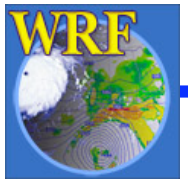
- **discard**: fields if '**all**' is selected on the command line
- **retain**: fields if '**basic**' is selected on the command line

NMM only

ripdp namelist

- **iinterp = 1:** interpolate to a new B-grid
- **dskmcib:** grid spacing, in km, of the coarse domain on which the new B-grid will be based
- **miycorsib, mjxcorsib:** number of grid points in the y and x directions of new B-grid
- **nprojib:** map projection number (0: none/ideal, 1: LC, 2: PS, 3: ME, 4: SRCE) of new B-grid
- **xlatcib, xloncib:** central latitude and longitude of new B-grid
- **truelat1ib, truelat2ib:** two true latitudes of new B-grid

- **miyib, mjxib:** number of grid points in the y and x directions, of the fine domain
- **yicornib, xjcornib:** coarse domain y and x locations of the lower left corner point of the fine domain
- **dskmib:** grid spacing, in km, of the fine domain



rip UIF

&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= .....

Frame specification group (FSG)

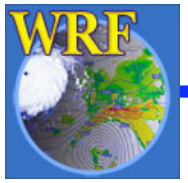
==========

feld= .....

Plot specification line (PSL)

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

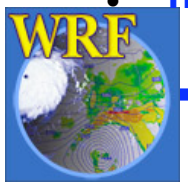
Plot Specification Table (PST)



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**

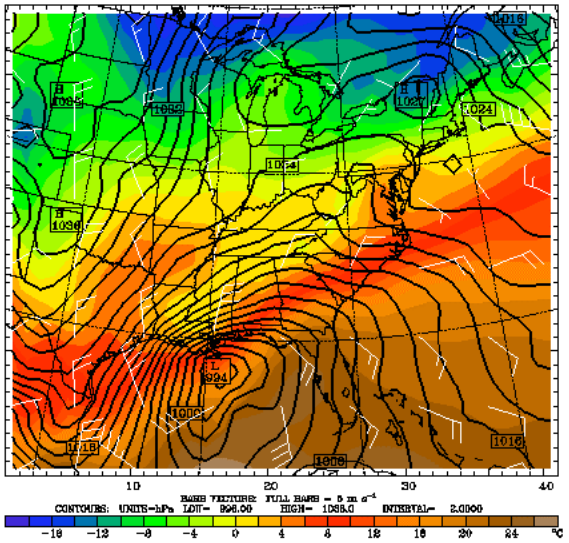


Creating a Plot

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**

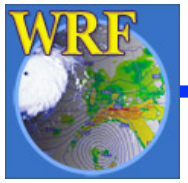
```
==========================================
feld=tmc; ptyp=hc; vcor=s; levs=1fb; >
   cint=2; cmth=fill; >
   cosq=32,light.violet,-16,blue, >
   0,yellow,16,orange,32,light.gray
feld=slp; ptyp=hc; cint=2; linw=2
feld=uuu,vvv; ptyp=hv; vcmx=1; >
    colr=white;intv=5
feld=map; ptyp=hb
feld=tic; ptyp=hb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



Common Error Message

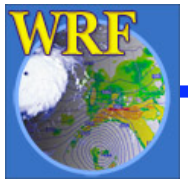
- **Most often this is NOT a graphics error.**
- **More often this is an error with the times you are asking RIP to process**
  - Check the ptimes in your .in file
  - Check the xtimes files created by RIPDP

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



ARWpost

- **Converter**
  - Requires GrADS / vis5d to display data
- **GrADS software only needed to display data**
- **If vis5d output is required, vis5sd libraries are needed to compile the code** *(not recommended)*
- **Generate a number of graphical plots**
  - Horizontal, cross-section, skewT, meteogram, panel



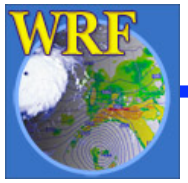
ARWpost - *converter*

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

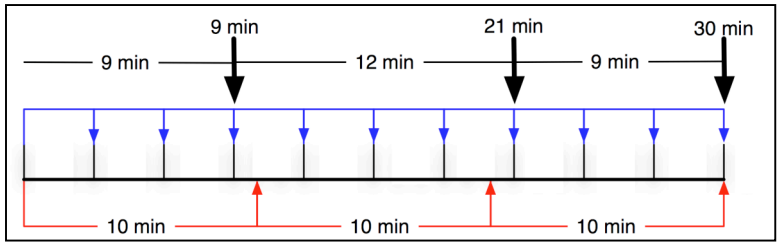
- **MUST have WRF compiled *(similar to WPS)***
  ./configure & ./compile

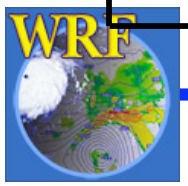
- **For GrADS output**
  - GrADS libraries only needed to display data *(freely available)*
  - http://grads.iges.org/grads/grads.html

- **For vis5d output**
  - vis5d libraries needed for compilation *(freely available)*
  - http://www.ssec.wisc.edu/~billh/vis5d.html



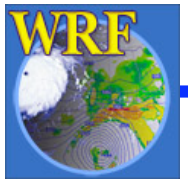
namelist.ARWpost

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



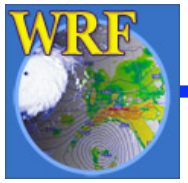
namelist.ARWpost

io_form_input	2=netCDF, *5=GRIB1*
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**, OR *output_root_name.**v5d***
output_type	Options are 'grads' (*default*) or *'v5d'*
mercator_defs	Set to true if mercator plots are distorted



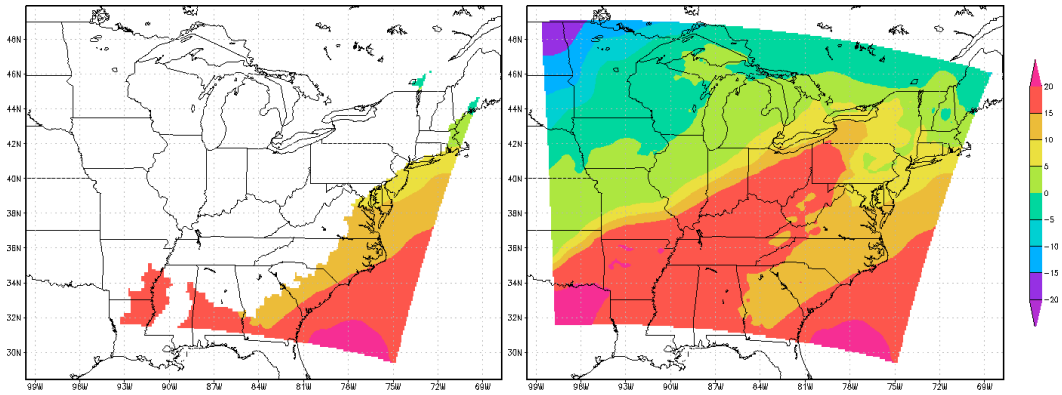
namelist.ARWpost

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

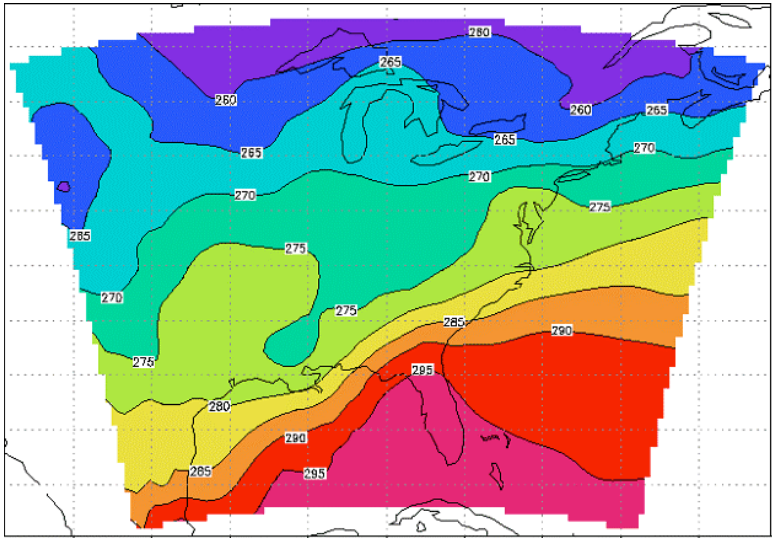
interp_method	0 = sigma levels, -1 = code defined "nice" height levels, 1 = user defined height or pressure levels
interp_levels	Only used if interp_method=1 Supply levels to interpolate to, in hPa (*pressure*) or km (*height above sea level*) Supply levels bottom to top
extrapolate	Extrapolate below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

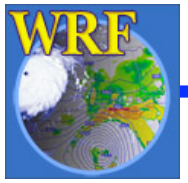


GrADS - .ctl file

```
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
```

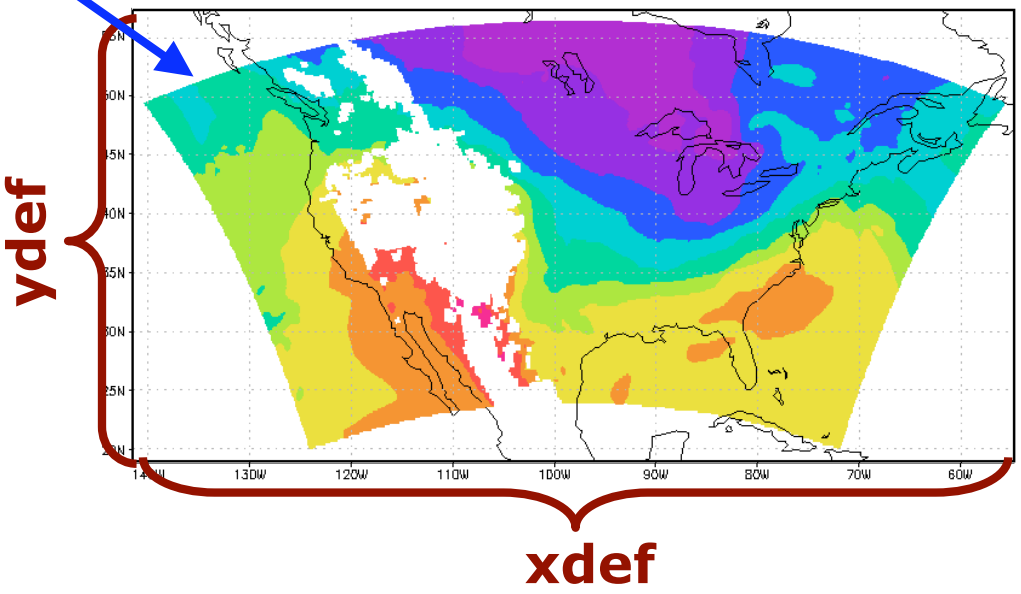
options byteswapped

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



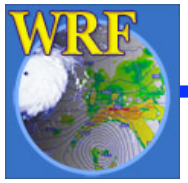
GrADS - .ctl file

```
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
```



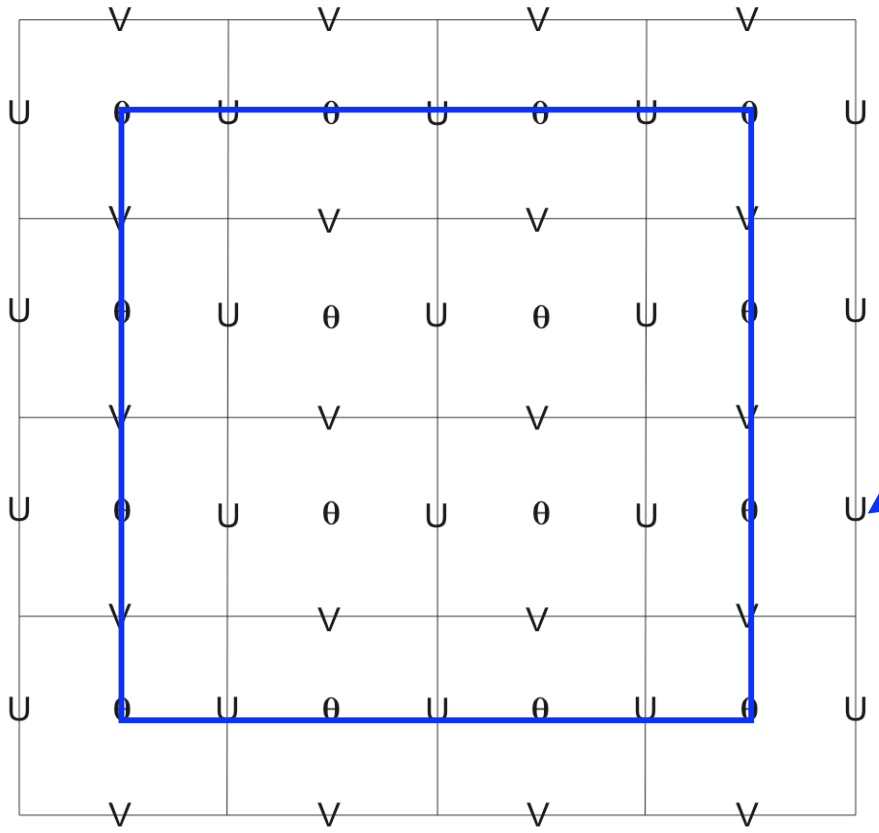
GrADS conversion - question

- **Why is a converter needed if GrADS can display netCDF files?**
  - Can only display model surface coordinates
  - Cannot interpolate to height or pressure levels
  - All diagnostics must be added via GrADS script files
  - GRIB1 model output can also be read directly by GrADS, but above issues are still valid
  - For GRIB1, there is also a stagger problem



GrADS conversion - question

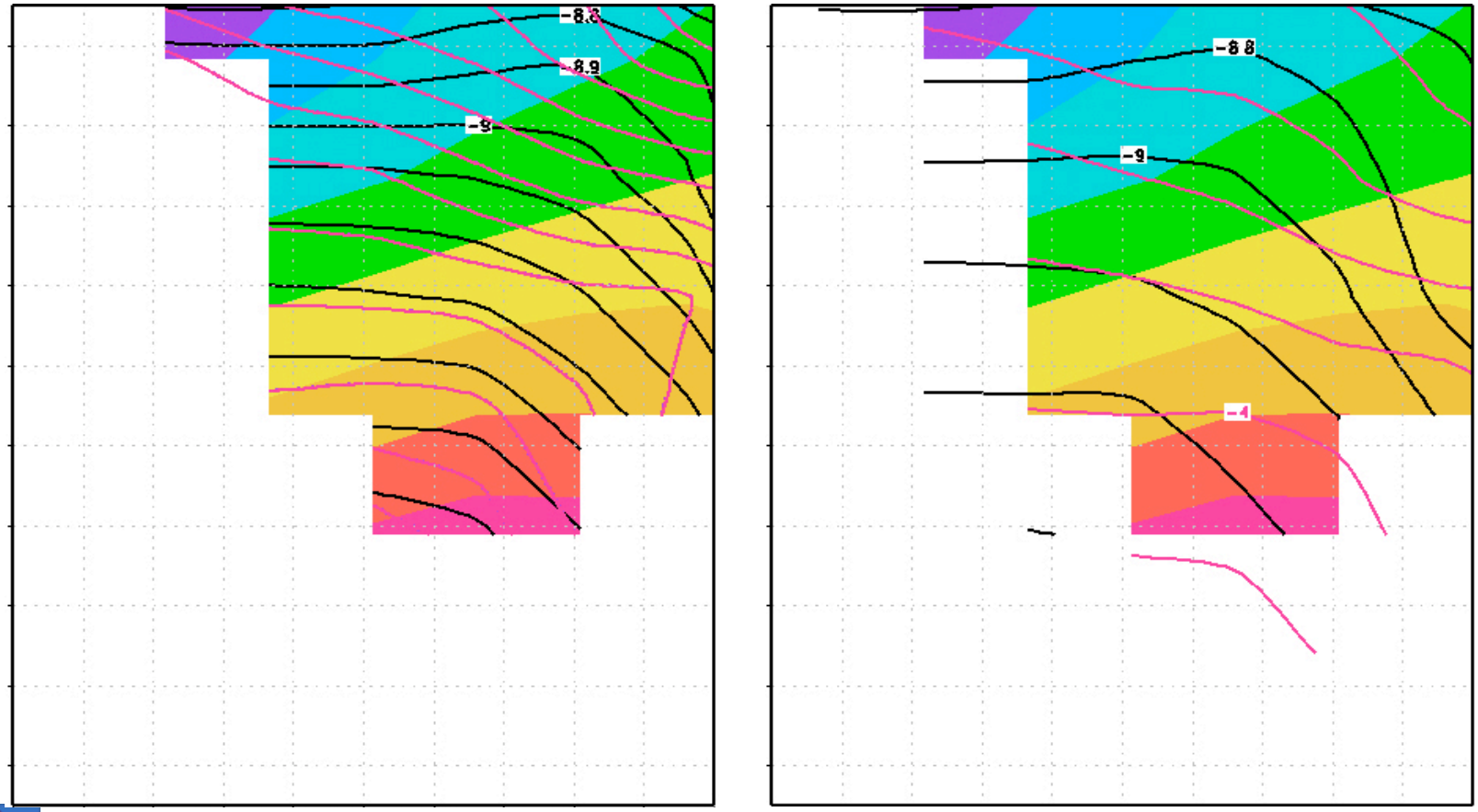
- **Why is a converter needed if GrADS can display netCDF files?**



WRF staggered grid

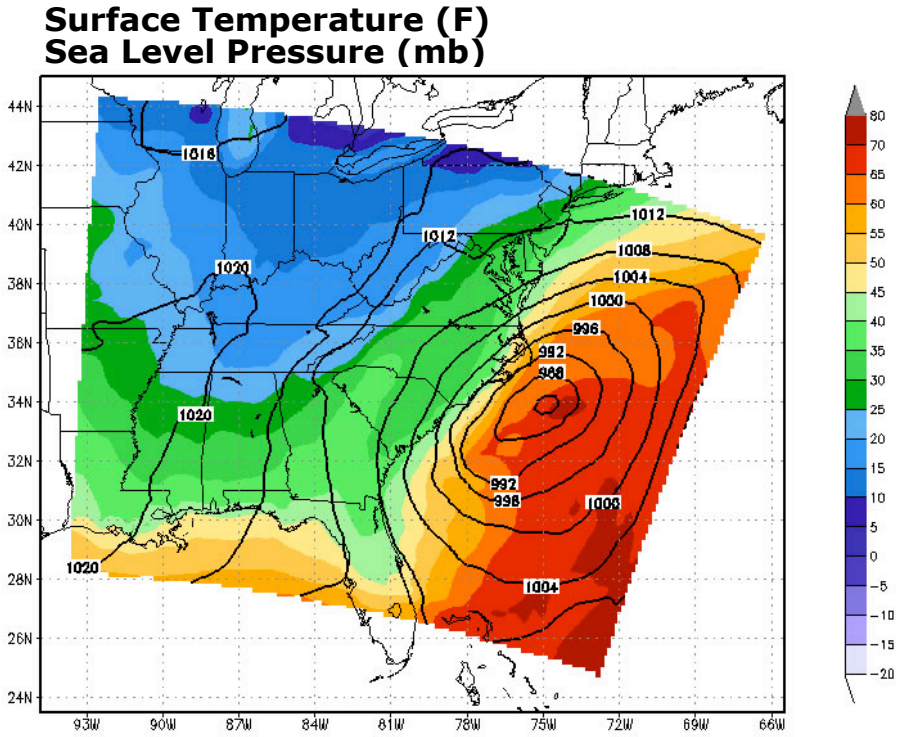
Staggering

shaded=T ; black=U ; red=V



Creating a Plot

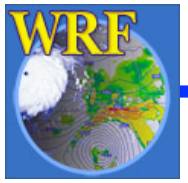
```
open em_real.ctl
set mpdset hires
set display color white

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
```



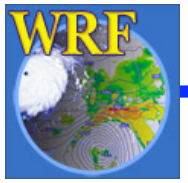
How to add diagnostics

- **RIP4**
  - Create a subroutine (note RIP4 expects the code to be in “j/l/-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

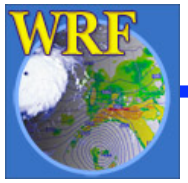


VAPOR

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



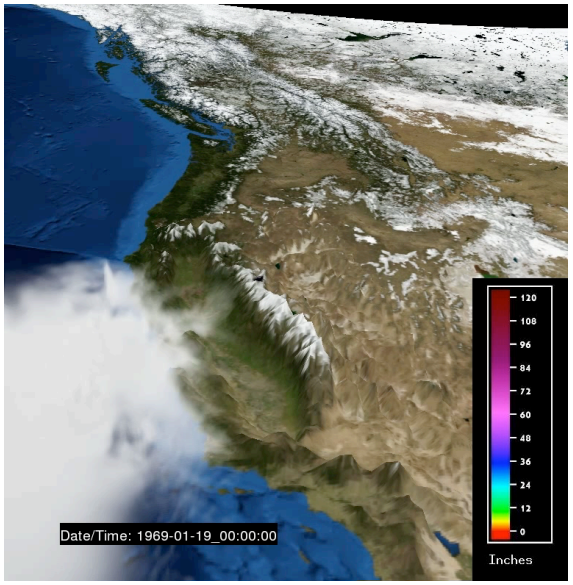
WRF in VAPOR

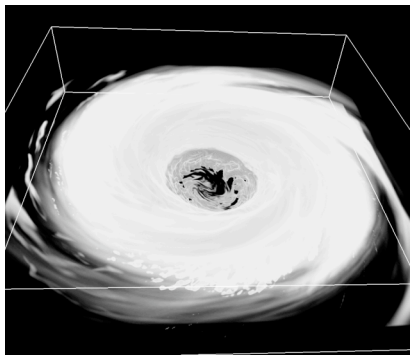
Computational and Information Systems Laboratory
National Center for Atmospheric Research

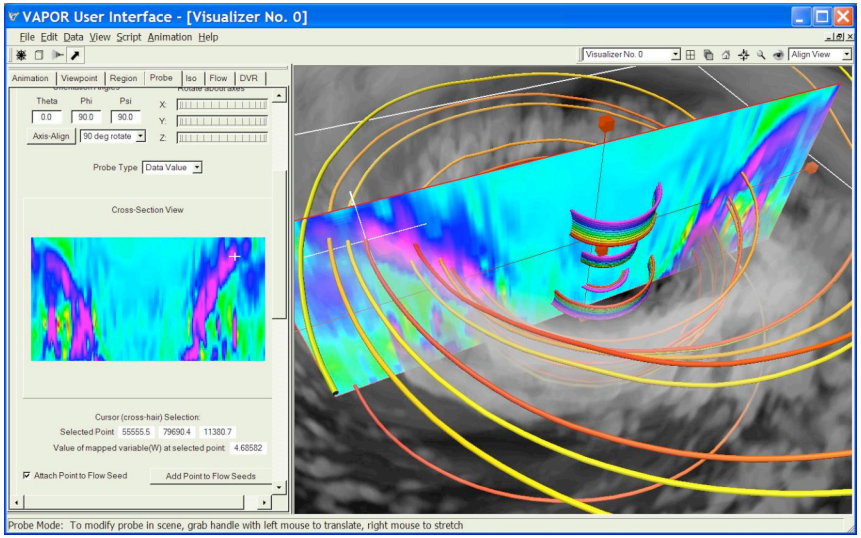
- **Interactive 3D visualization of WRF-ARW data *(wrfout files only)***
- **Available free on Linux, Windows, Mac**
- **Interactive rendering and animation (using GPU acceleration)**
- **Simple 2-step data conversion from WRF output to VAPOR**
  - wrfvdfcreate & wrf2vdf
- **Steady and unsteady flow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

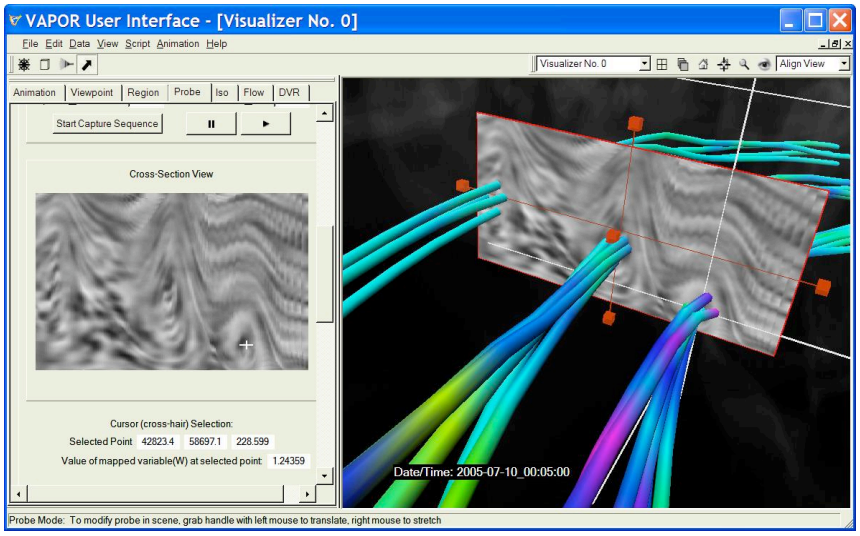
WRF in VAPOR

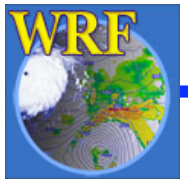
Computational and Information Systems Laboratory
National Center for Atmospheric Research











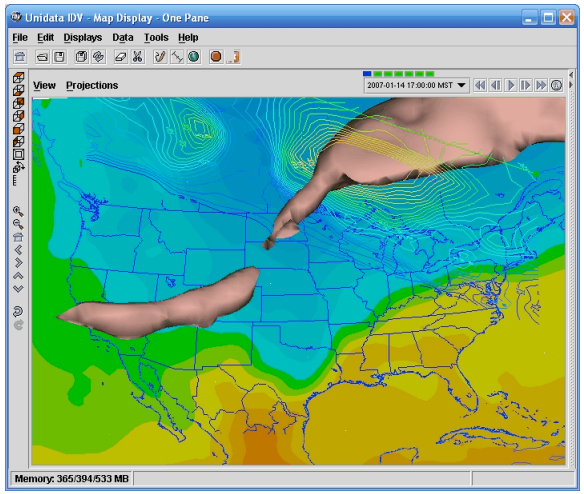
IDV

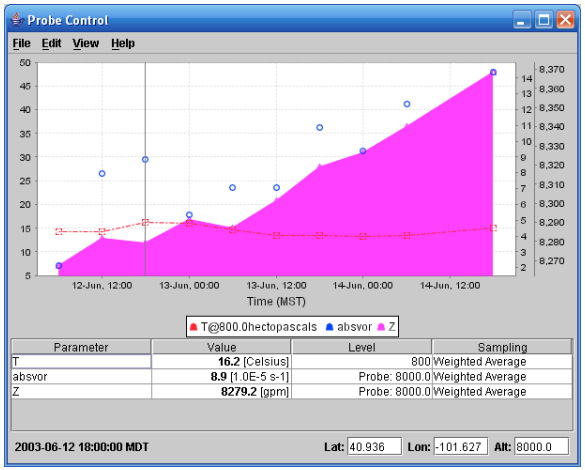
Integrated Data Viewer

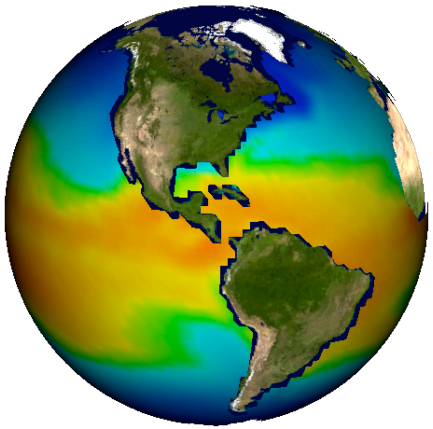
Don Murray and Jeff McWhirter
Unidata Program Center/UCAR

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**

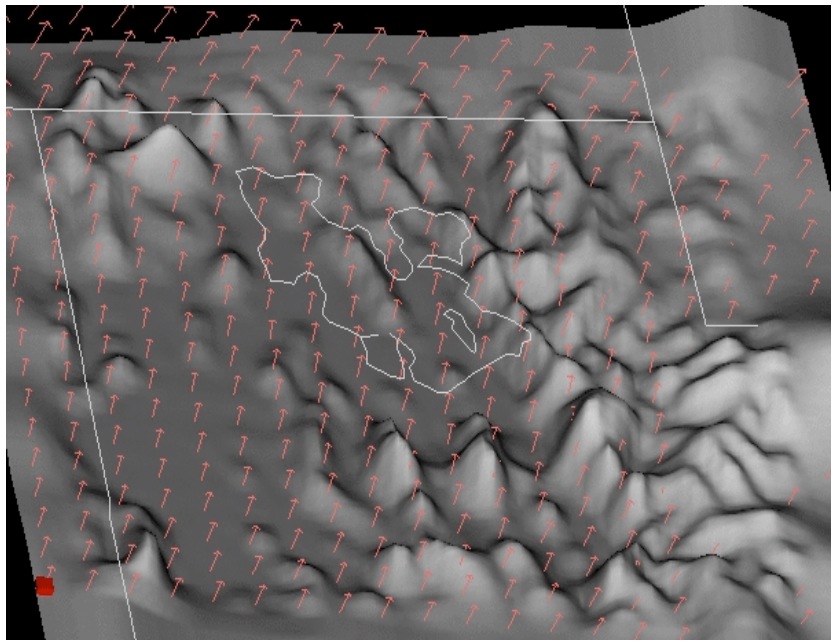






IDV Strengths

- **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