WRF ARW Graphical Tools

Cindy Bruyère

outline

- **netCDF Data**
- **NCL**
- **RIP4**
- **ARWpost** ***(old WRF-to-GrADS & WRF-to-VIS5D)***

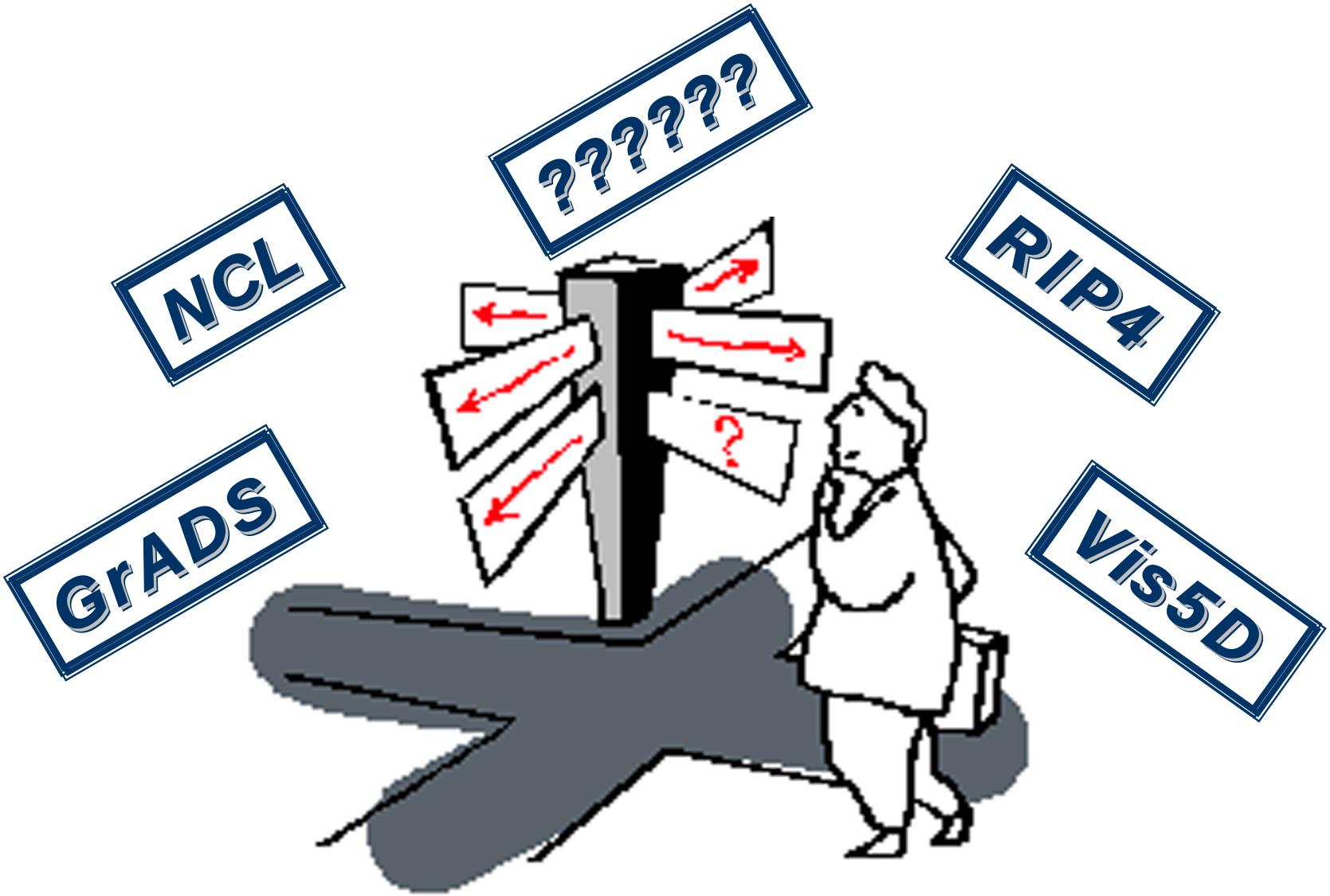
netCDF data

- **netCDF stands for *net*work *C*ommon *D*ata *F*orm**
- **netCDF is one of the current supported data formats chosen for WRF I/O API**
  - WRF I/O supports netCDF *(not CF compliant)*/ binary/GRIB/HDF
  - Most support graphical packages currently only support netCDF file format
- http://www.unidata.ucar.edu/
  - Documentation available at this site

available graphics

- NCL
- RIP4
- GrADS
- Vis5D
- IDL
- Matlab
- GMT
- FX-Net
- Interactive Visualization Environment (*IVE*)

- General Meteorological Package / National Advanced Weather Interactive Processing System (GEMPAK / NAWIPS, e.g. “garp” and “nmap”)
- Integrated Data Viewer *(IDV)*
- Open-source Data Explorer *(OpenDX)*


??????
NCL
RIP4
GrADS
Vis5D

	NCL	RIP4	ARWpost (GrADS / Vis5D)	
Directly ingest WRF model output	✓	✖ converter	✖ / (✓) converter	
Vertical Coordinate	n / p / h	n / p / h	n / p / h	
Model input & output	i / o / wps	i / o / wps	i / o / wps	
Ideal	✓ 3D / 2D	✓ 3D / 2D	✓ 3D / 2D / 1D	✓ 3D / 2D
WRF-Var *(double precision)*	✓	✓	✓	
Input format	netCDF	netCDF	netCDF / GRIB1	
Publish Quality	Good	Good	Fair	Difficult
Easy of use	NCL - need to do some scripting beyond standard plots			
	RIP4 – easy, but need to know available fields			
	GrADS - take some time to learn, but one can get up to speed fast			
	ViS5D - need a large computer to run, as it is memory intensive			

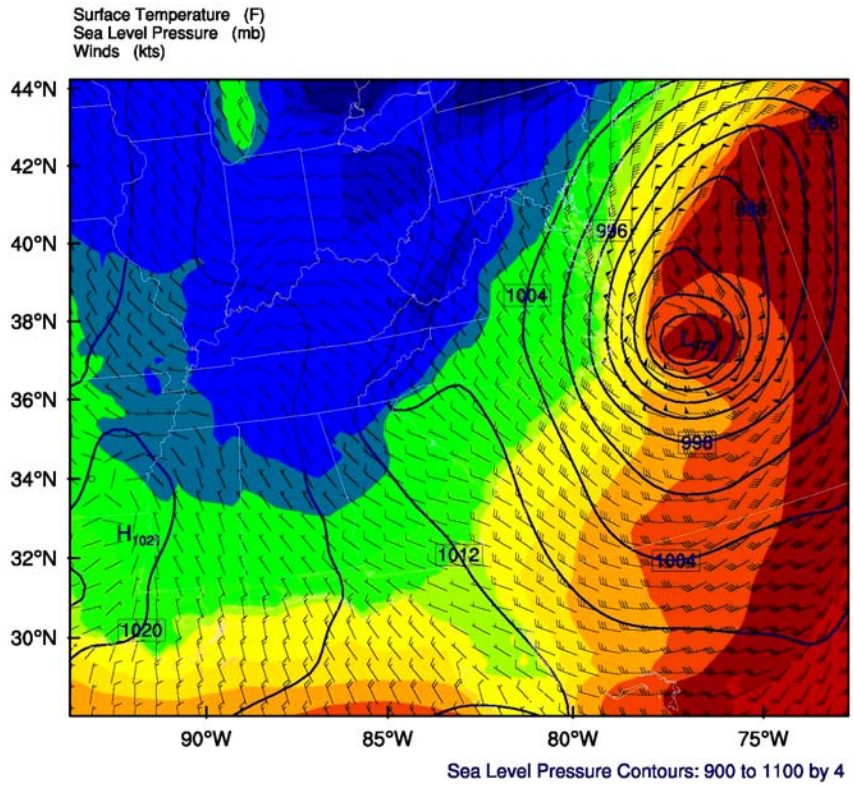
	NCL	RIP4	ARWpost (GrADS / Vis5D)	
Software required *(All binaries are free)*	*High level NCL*	*NCAR Graphics*	*GrADS*	*Vis5D*
Diagnostics	*Some*	*>100*	*Some*	
Adding diagnostics	*Fortran + NCL scripting*	*Fortran*	*Fortran + GrADS scripting*	
Intermediate files	*NO*	*LOTS*	*One large file*	
Interactive?	✓	✗	✓	✓
Can also be used with	*GRIB1+ ASCII*	---	*GRIB1+ Station*	---
Special Overlays	*Station*	✗	*Station*	*2D+3D*
3D Capability	✗	✗	✗	✓
Multiple Input Files	✓	✓	✓	✓
Cross-sections	✓	✓	✓	✓

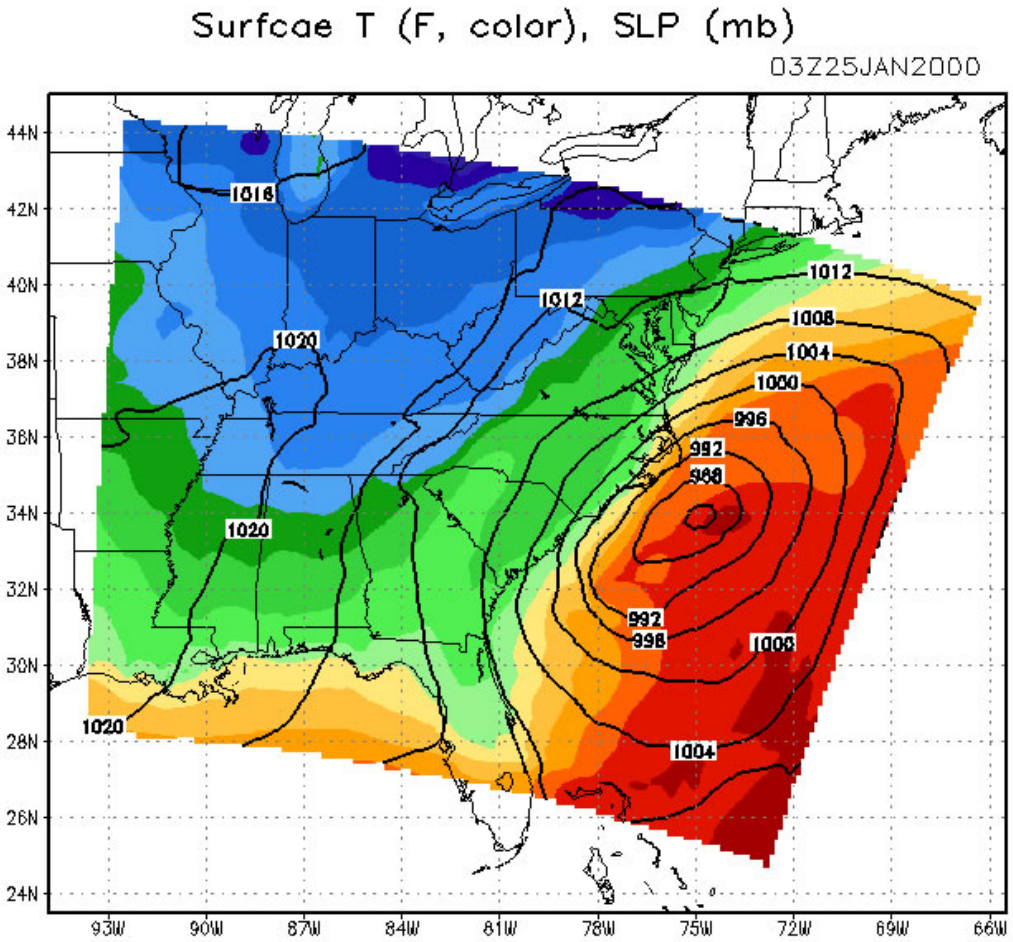
	NCL	RIP4	ARWpost (GrADS / Vis5D)	
Skew-T's	✓	✓	✓	✗
Plot sub-domains *(zooming)*	✓	✓	✓	✗
Animations?	✗	✗	✓	✓
Multiple images per page?	✓	✗	✓	✗
Plotting format	*X-Y*	*X-Y*	*Projected*	*3D*
Output formats	*X11, ps, pdf, cgm*	*cgm*	*X11, gif, png*	*X11*
Create Trajectory data	✗	✓	✗	✗
Input to Vis5D	✗	✓	✓	

examples

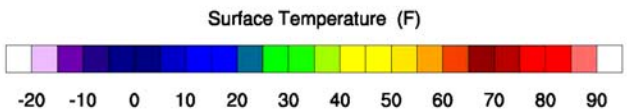
REAL-TIME WRF

Init: 2000-01-24_12:00:00
Valid: 2000-01-25_12:00:00



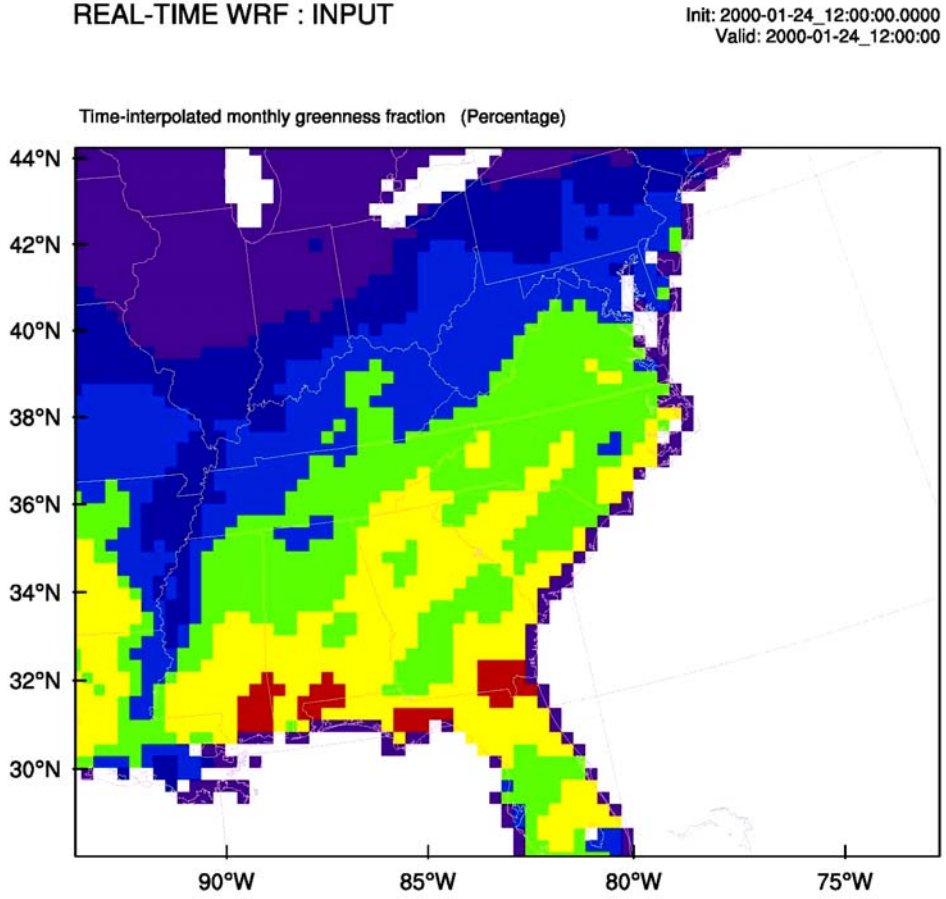


NCL

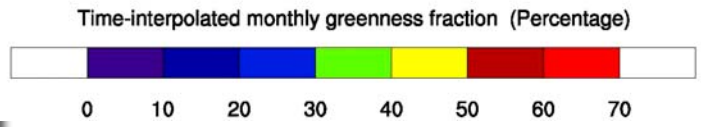


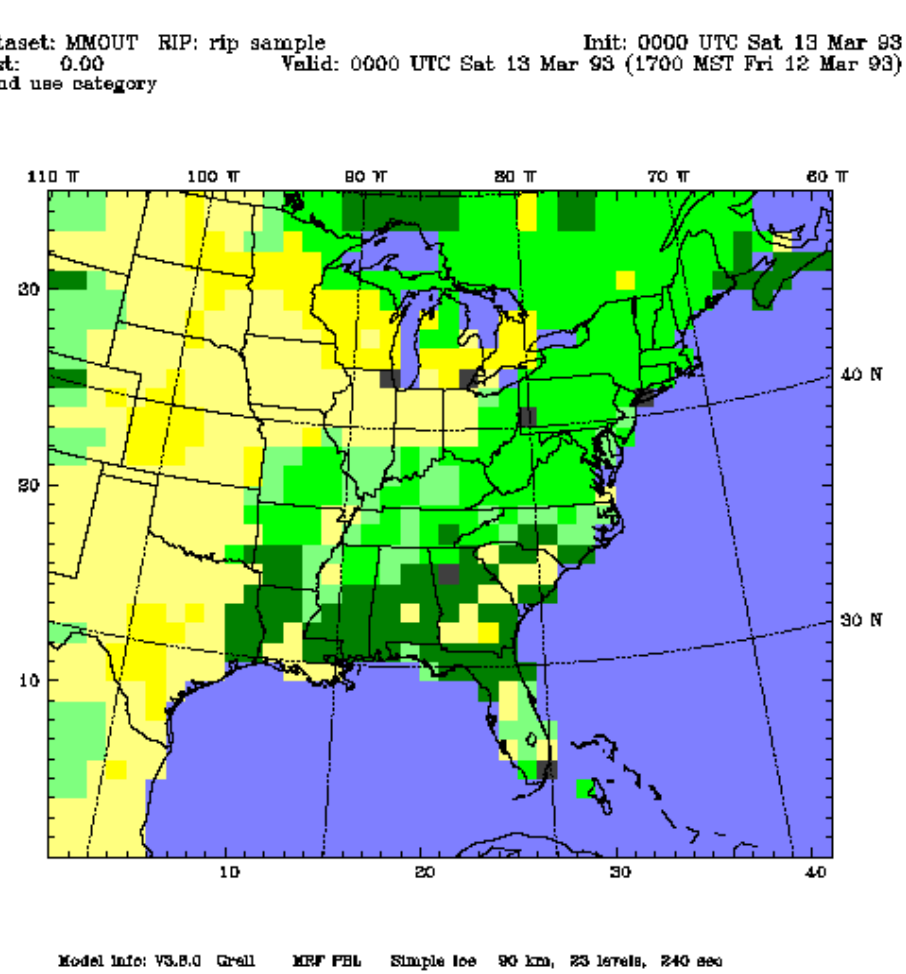
GrADS

REAL-TIME WRF : INPUT

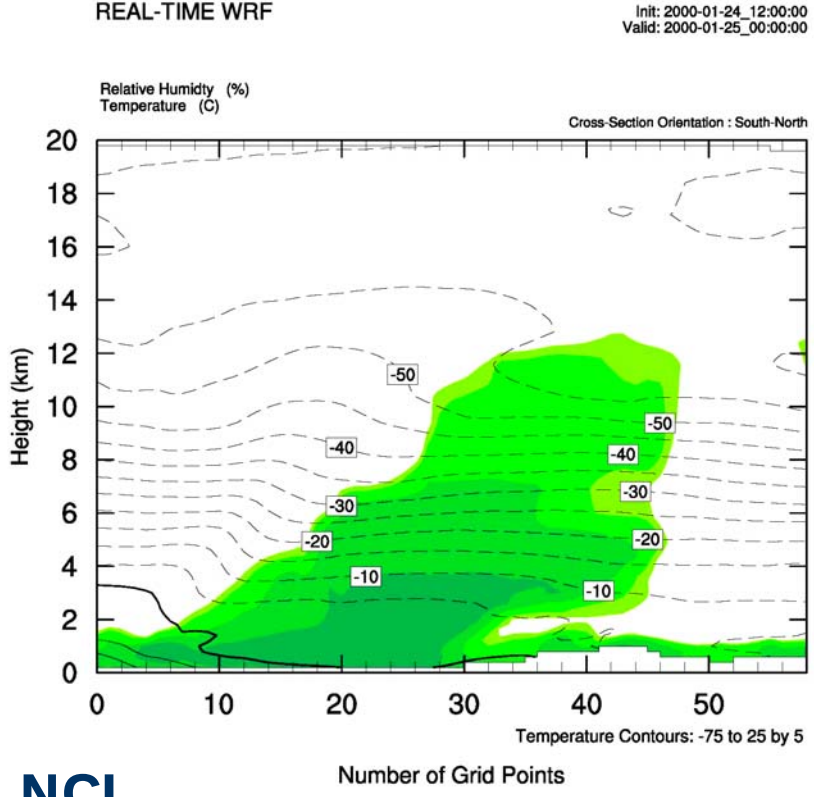


NCL

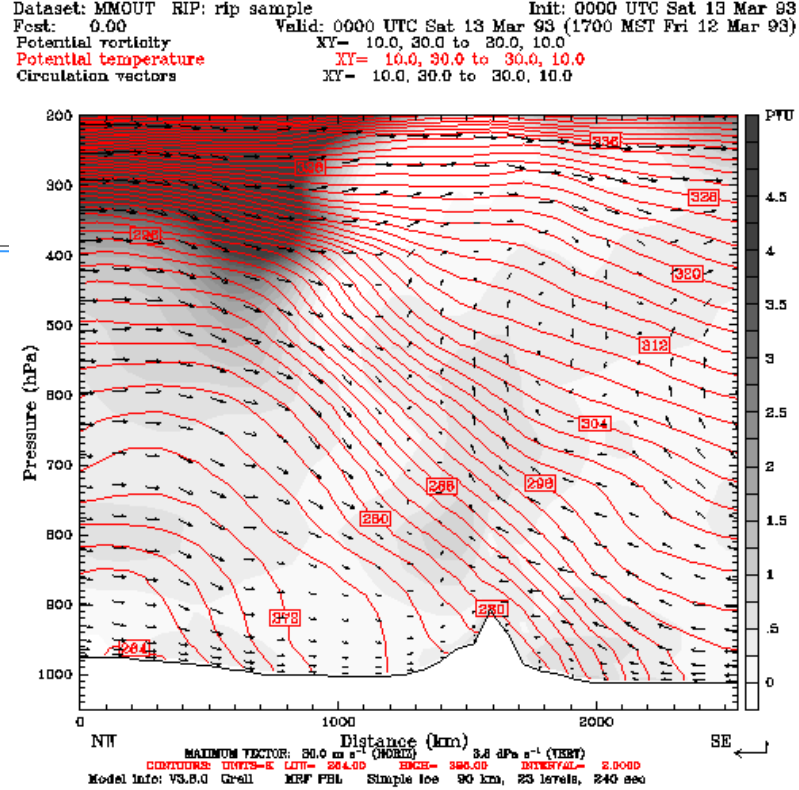




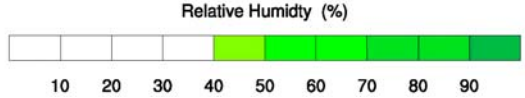
RIP4

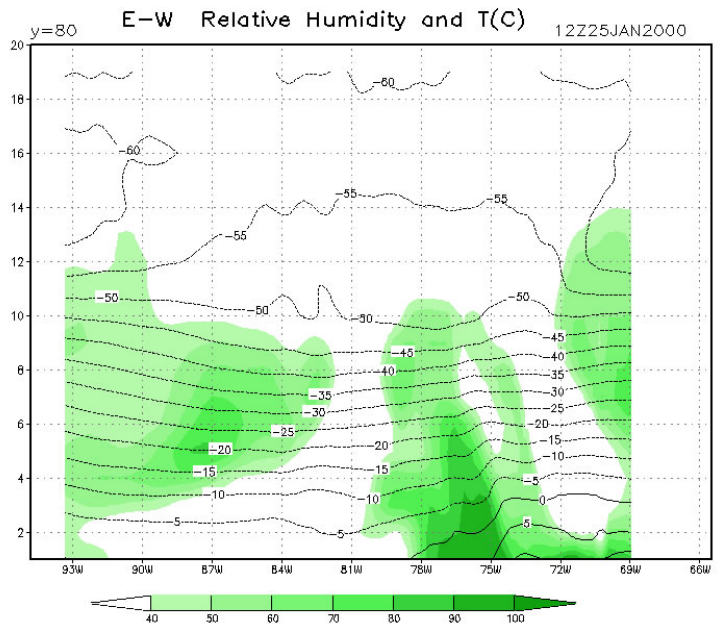


NCL

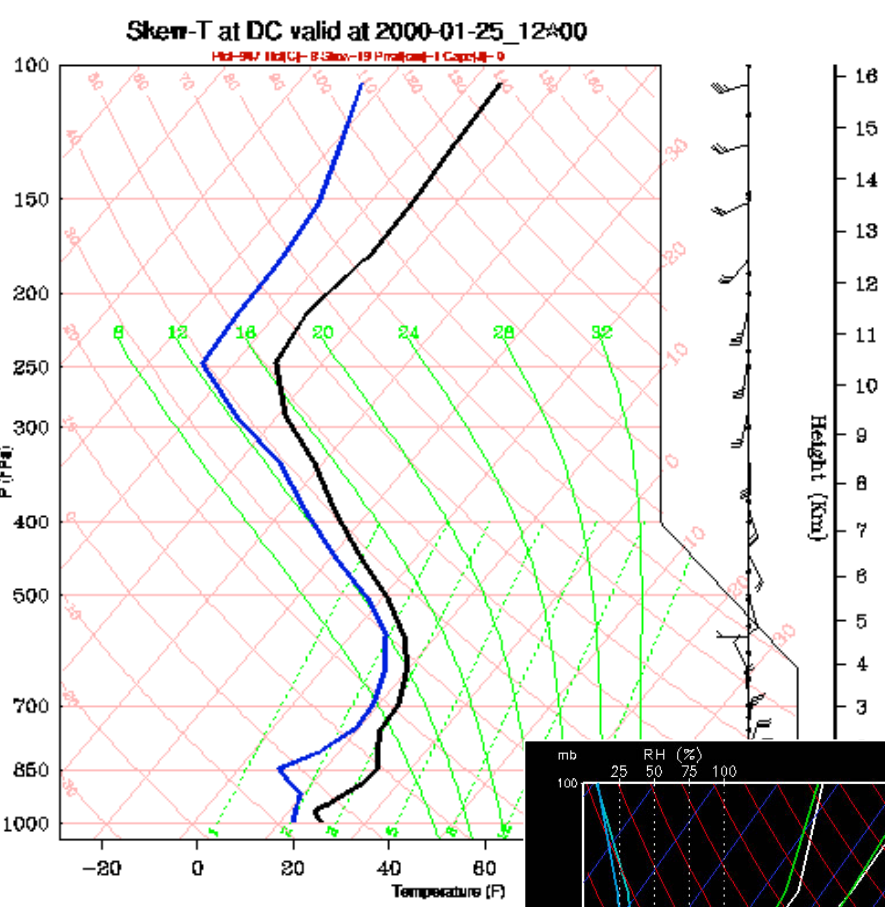


RIP4

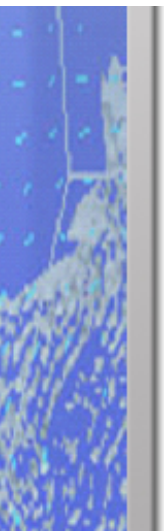


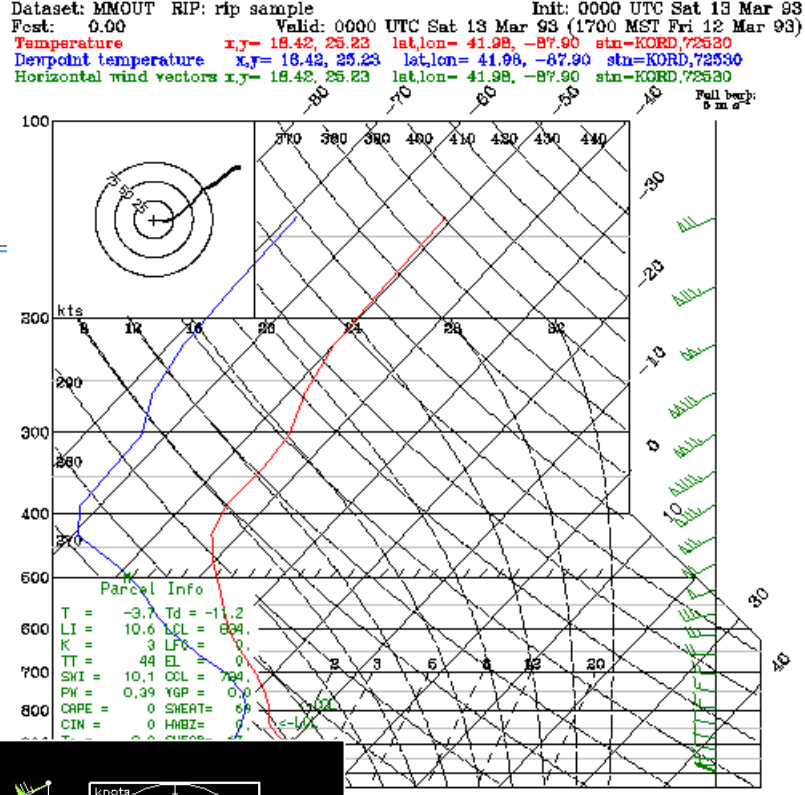


GrADS

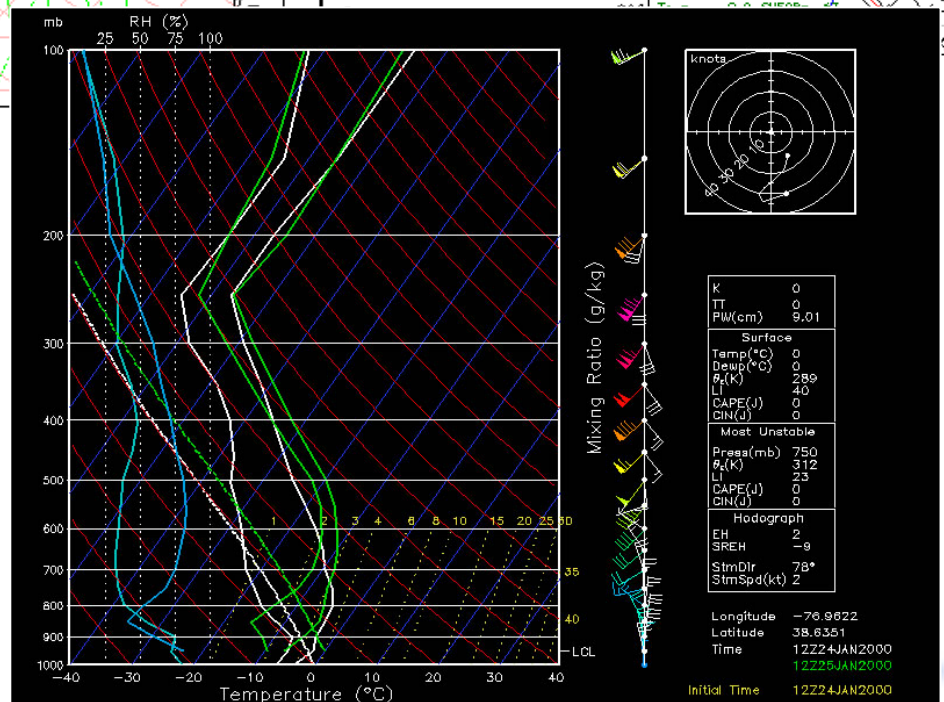


NCL

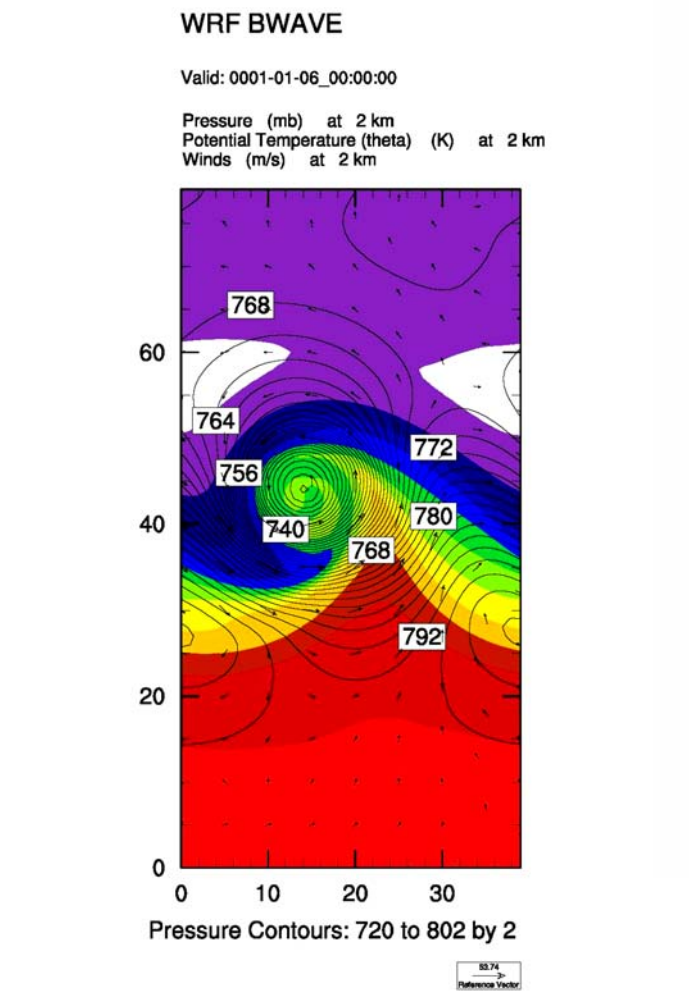




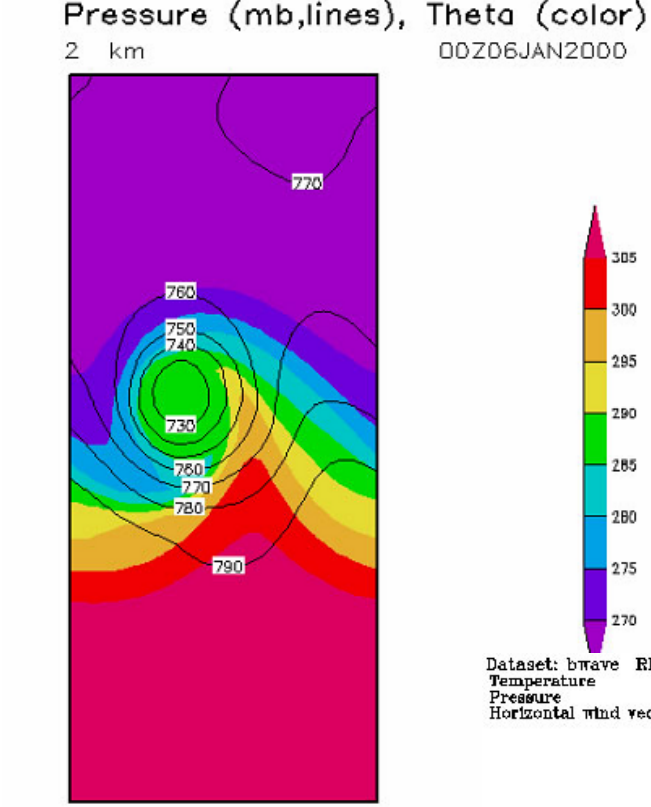
RIP4



GrADS

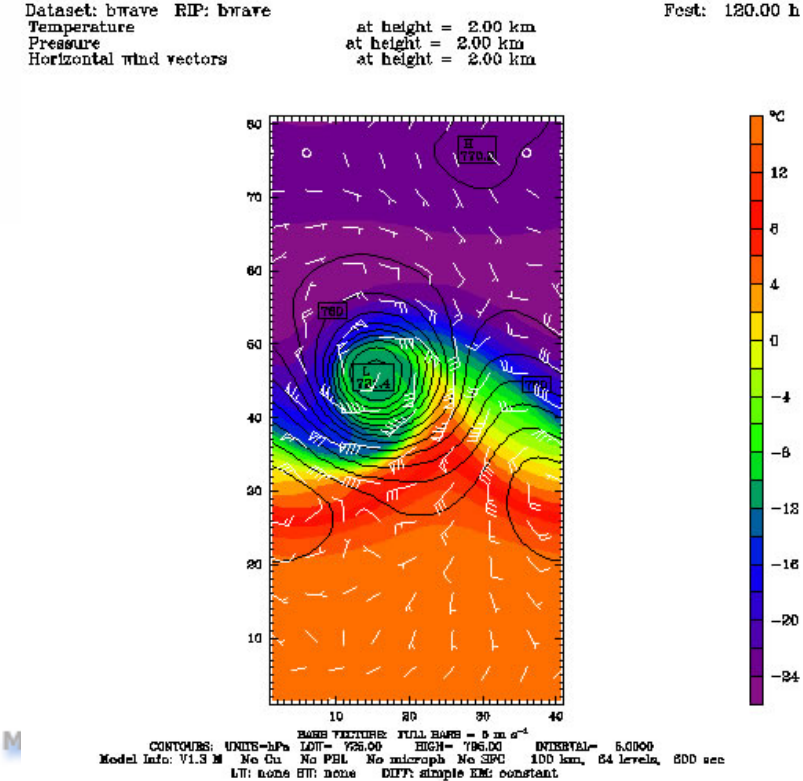


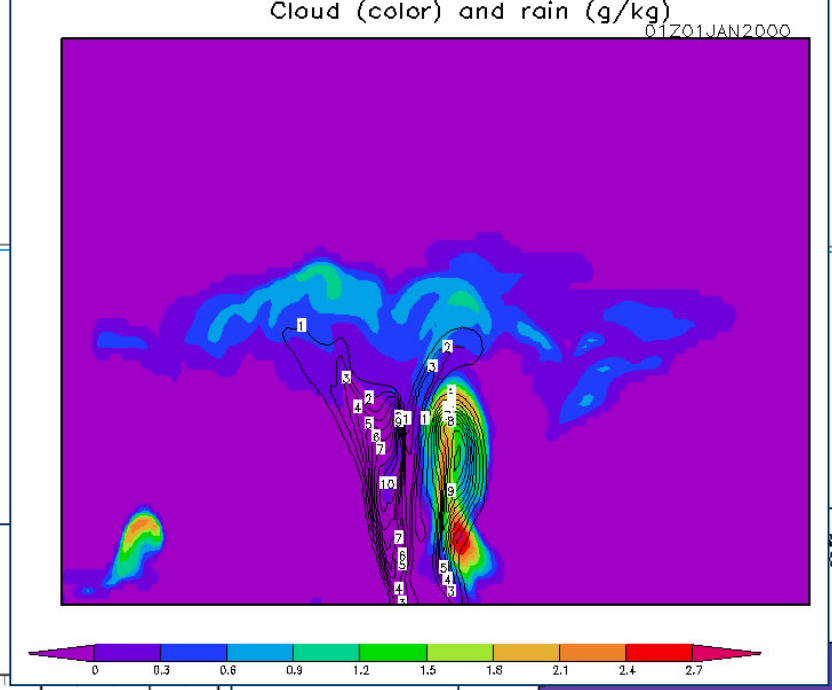
NCL



GrADS

RIP4





GrADS

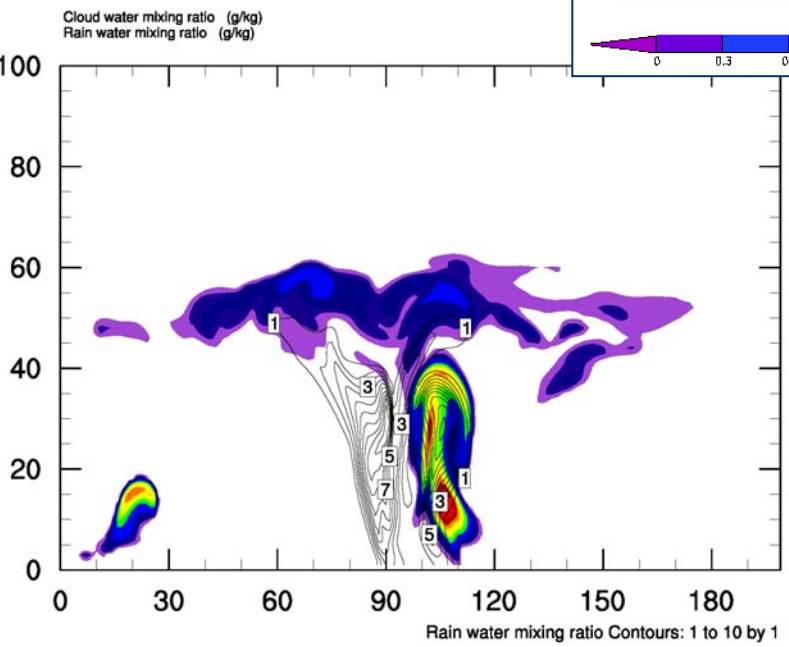
WRF squall2D_x

Fest: 1.00 h

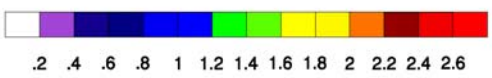
Y= 2.0, 2.0 to 201.0, 2.0

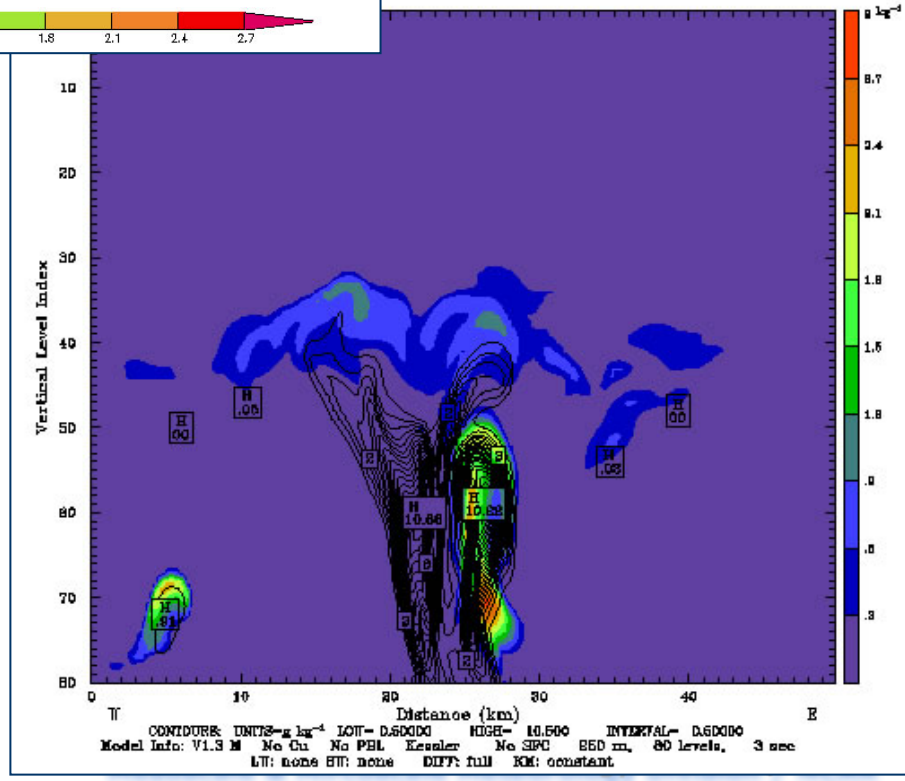
Y= 2.0, 2.0 to 201.0, 2.0

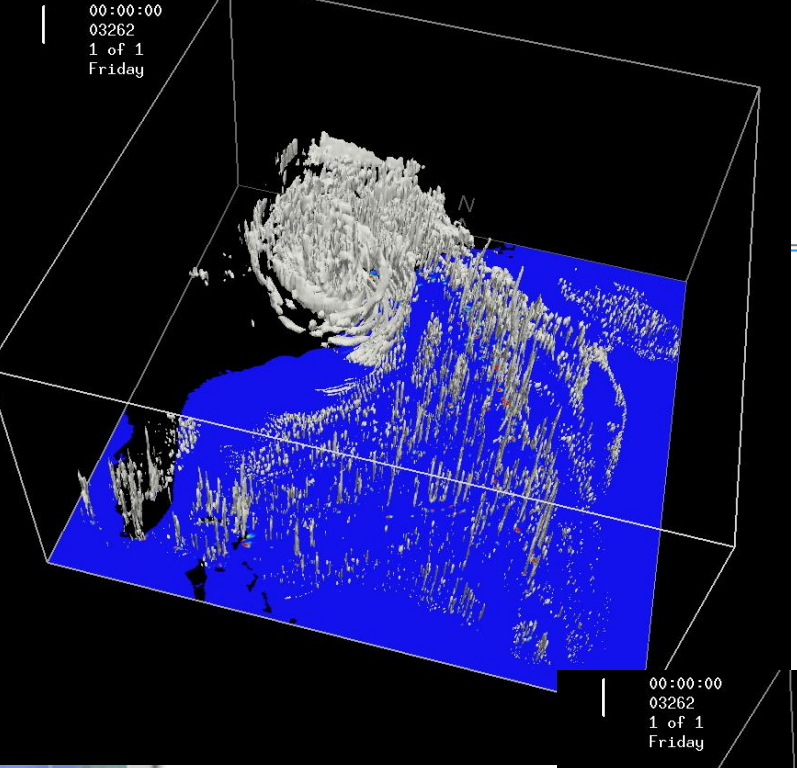
RIP4

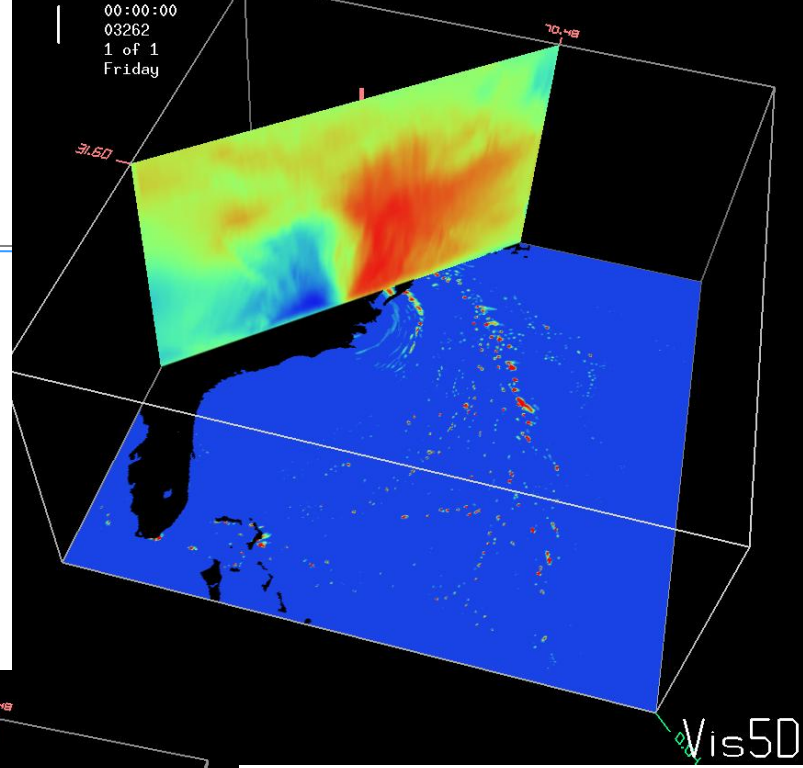


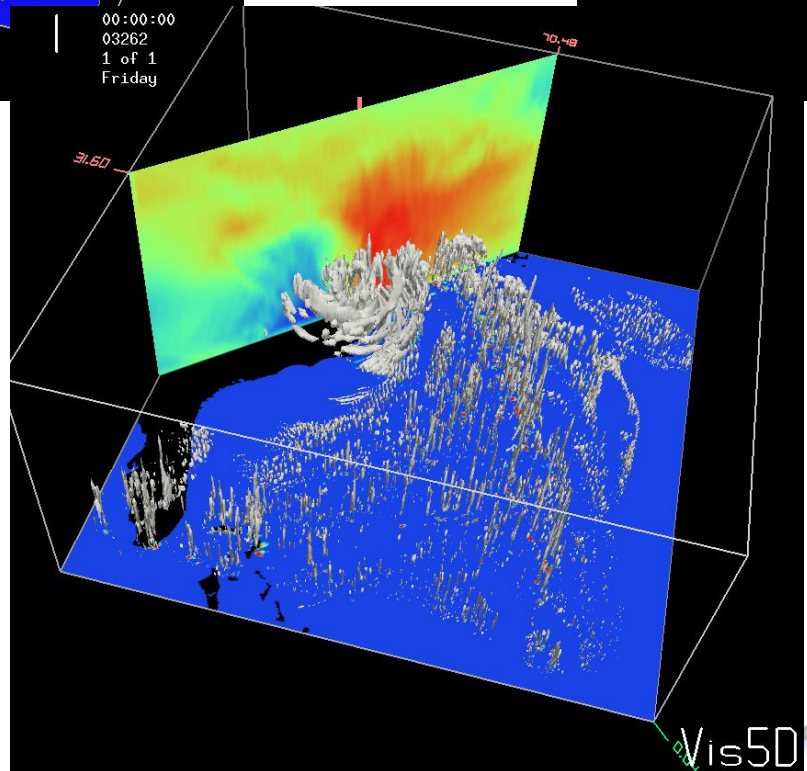
NCL



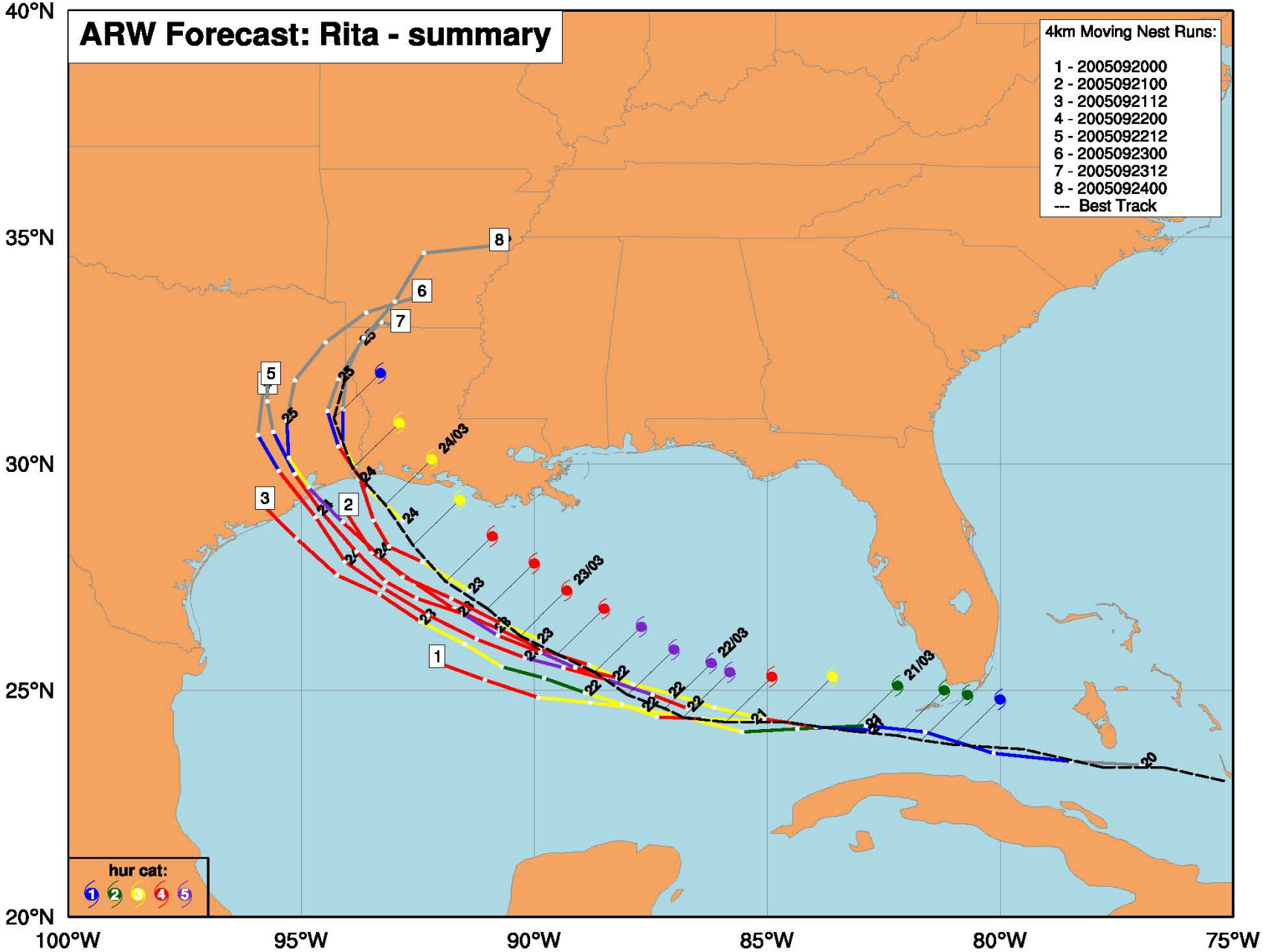








Vis5D


NCL
ARW Forecast: Rita - summary
4km Moving Nest Runs:
1 - 2005092000
2 - 2005092100
3 - 2005092112
4 - 2005092200
5 - 2005092212
6 - 2005092300
7 - 2005092312
8 - 2005092400
--- Best Track
hur cat:
1 2 3 4 5
40°N
35°N
30°N
25°N
20°N
100°W
95°W
90°W
85°W
80°W
75°W
24/03
23/03
22/03
21/03

NCL

NCL stands for *NCAR Command Language*

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

downloading NCL

- **Go to:**
  - **http://www.ncl.ucar.edu/Download/**
    Read and agree to GPL license
    Fill out short registration form
    Download binaries
- **Set NCARG_ROOT environment variable:**
  - *setenv NCARG_ROOT /usr/local/ncarg*
- ***Recommended to install NCAR Graphics first and then NCL on top of NCAR Graphics*** *(most people don't - setenv NCARG_ROOT /usr/local/ncl)*
- ***WRF_NCL (Sep 2005 version) needs NCL version 4.2.0.a032, or higher***

WRF_NCL

- **Current release - Version 2.0**
- **Release date September 2005**
- **Re-write of functions and procedures**
- **Cannot combine code from version 1 and version 2**
  - *due to the re-write of WRF functions and procedures*
  - *and upgrades made to newer versions of NCL*

download NCL for WRF

- From wrf-model.org web site (*wrf_ncl_tar.gz*):

  http://www.mmm.ucar.edu/wrf/users/download/get_source.html

- "How to" and examples:

  http://www.mmm.ucar.edu/wrf/users/graphics/WRF_NCL/NCL.htm

NCL for WRF

README_FIRST
README_NCL

DOThluresfile

gsn_code.ncl
SkewTFunc.ncl
WRFOptions.ncl
WRFPlot.ncl
WRFUserARW.ncl

wrf_user_fortran_util_0.f

make_ncl_fortran
make_ncl_fortran.alpha
make_ncl_fortran.ibm
make_ncl_fortran.ibm64
make_ncl_fortran.linux
make_ncl_fortran.sun

wrf_bwave.ncl
wrf_hill2d.ncl
wrf_grav2d.ncl
wrf_qss.ncl
wrf_squall_2d_x.ncl
wrf_squall_2d_y.ncl
wrf_real_input.ncl
wrf_real.ncl
wrf_cloud.ncl

NCL functions and procedures

gsn_code.ncl
High level NCL scripts

SkewTFunc.ncl
Skew-T plots

WRFOptions.ncl
Basic plotting options, like headers/footers

WRFPlot.ncl
maps, contour plots, shaded plots, vectors

WRFUserARW.ncl
Obtain native model variables
Basic diagnostics
Interface to FORTRAN code (wrf_fortran_user_util_0.f)

Must be loaded at top of NCL script, i.e.
load "WRFPlot.ncl"

NCL for WRF

README_FIRST
README_NCL

DOThluresfile

gsn_code.ncl
SkewTFunc.ncl
WRFOptions.ncl
WRFPlot.ncl
WRFUserARW.ncl

wrf_user_fortran_util_0.f

make_ncl_fortran
make_ncl_fortran.alpha
make_ncl_fortran.ibm
make_ncl_fortran.ibm64
make_ncl_fortran.linux
make_ncl_fortran.sun

wrf_bwave.ncl
wrf_hill2d.ncl
wrf_grav2d.ncl
wrf_qss.ncl
wrf_squall_2d_x.ncl
wrf_squall_2d_y.ncl
wrf_real_input.ncl
wrf_real.ncl
wrf_cloud.ncl

.hluresfile

- **Required by NCL**
- **Must be in your ~/ directory**
- **Control**
  - color table
  - font
  - white/black background
  - size of plot
  - control characters
- ***http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml***

- **cp DOThluresfile ~/.hluresfile** ← NB

NCL for WRF

README_FIRST
README_NCL

DOThluresfile

gsn_code.ncl
SkewTFunc.ncl
WRFOptions.ncl
WRFPlot.ncl
WRFUserARW.ncl

wrf_user_fortran_util_0.f

make_ncl_fortran
make_ncl_fortran.alpha
make_ncl_fortran.ibm
make_ncl_fortran.ibm64
make_ncl_fortran.linux
make_ncl_fortran.sun

wrf_bwave.ncl
wrf_hill2d.ncl
wrf_grav2d.ncl
wrf_qss.ncl
wrf_squall_2d_x.ncl
wrf_squall_2d_y.ncl
wrf_real_input.ncl
wrf_real.ncl
wrf_cloud.ncl

wrf_user_fortran_util_0.f

- Contains FORTRAN routines for diagnostic calculations
- Code must be in F77
- Code must contain special NCL START and END markers
- Must create a shared object library from these routines in order for NCL to recognize the code.

```
subroutine compute_tk ( tk, pressure, theta, nx, ny, nz )
implicit none
integer nx,ny,nz
real   pi
real   pressure(nx,ny,nz)
real   theta(nx,ny,nz)
real   tk(nx,ny,nz)


 integer i,j,k
real p1000mb, r_d, cp
parameter (p1000=100000., r_d=287., cp=7.*r_d/2.)

  do k=1,nz
  do j=1,ny
  do i=1,nx
   pi=(pressure(i,j,k)/p1000)**(r_d/cp)
   tk(i,j,k) = pi*theta(i,j,k)
  enddo
  enddo
  enddo

end
```

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


      integer i,j,k
      real p1000mb, r_d, cp
      parameter (p1000=100000., r_d=287., cp=7.*r_d/2.)

        do k=1,nz
        do j=1,ny
        do i=1,nx
         pi=(pressure(i,j,k)/p1000)**(r_d/cp)
         tk(i,j,k) = pi*theta(i,j,k)
        enddo
        enddo
        enddo

      end
```

wrf_user_fortran_util_0.so

WRAPIT wrf_user_fortran_util_0.f

setenv NCARG_ROOT */usr/local/ncl*

/usr/local/ncl/bin/WRAPIT
wrf_user_fortran_util_0.f

USE in NCL script

wrf_user_fortran_util_0 ::
compute_tk(tk, pressure, theta, nx, ny, nz)

(WRFUserARW.ncl)

NCL for WRF

README_FIRST
README_NCL

DOThluresfile

gsn_code.ncl
SkewTFunc.ncl
WRFOptions.ncl
WRFPlot.ncl
WRFUserARW.ncl

wrf_user_fortran_util_0.f

make_ncl_fortran
make_ncl_fortran.alpha
make_ncl_fortran.ibm
make_ncl_fortran.ibm64
make_ncl_fortran.linux
make_ncl_fortran.sun

wrf_bwave.ncl
wrf_hill2d.ncl
wrf_grav2d.ncl
wrf_qss.ncl
wrf_squall_2d_x.ncl
wrf_squall_2d_y.ncl
wrf_real_input.ncl
wrf_real.ncl
wrf_cloud.ncl

wrf_user_fortran_util_0.so

IF WRAPIT DOES NOT WORK

Try one of the makefiles to create shared object library

make_ncl_fortran wrf_user_fortran_util_0

Potential Problem

wrapit77 is part of NCAR Graphics, and the path to this function needs to be correctly specified (.cshrc file)

a script

```
load "WRFOptions.ncl“  ; set up options res
load "gsn_code.ncl"
load "WRFPlot.ncl"
load "WRFUserARW.ncl"

begin
 ; Open a file
  a = addfile("wrfout_d01_2000-01-24_12:00:00.nc","r")
  type = "pdf"        ; Will create pdf output files
 ; open workstation + set output
  wks = gsn_open_wks(type,"wrfout")
  res@MainTitle    = "ARW Real Data"  ; Set plot title

  times  = wrf_user_list_times(a)  ; get times in the file
  ntimes = dimsizes(times)     ; number of times in the file
```

Not part of file name, but needs to be added inside the ncl script

a script

```
do it = 0,ntimes-1 ; loop through file for all times

   res@TimeLabel  = times(it)    ; Set Valid time info
   map = wrf_map(wks,a,res)      ; Create a map background
   ter = wrf_user_getvar(a,"HGT",time)   ; get terrain
   t2  = wrf_user_getvar(a,"T2",time)    ; get T2

  ; Set some options for terrain plot
   opts_ter = res
   opts_ter@cnFillOn             = True
   contour_ter  = wrf_contour(a,wks,ter,opts_ter)
  ; Set some options for t2 plot
   opts_t2 = res
   contour_t2  = wrf_contour(a,wks,t2,opts_t2)

 wrf_map_overlay(wks,map,(/contour_ter,contour_t2/),False)
end do

end
```

One image with map, shaded terrain and contour t2

a script - *alternative*

```
do it = 0,ntimes-1 ; loop through file for all times

   res@TimeLabel  = times(it)    ; Set Valid time info
   map = wrf_map(wks,a,res)      ; Create a map background
   ter = wrf_user_getvar(a,"HGT",time)   ; get terrain
   t2  = wrf_user_getvar(a,"T2",time)    ; get T2

   opts_ter = res
   opts_ter@cnFillOn             = True
   contour_ter  = wrf_contour(a,wks,ter,opts_ter)
   opts_t2 = res
   contour_t2  = wrf_contour(a,wks,t2,opts_t2)

 wrf_map_overlay(wks,map,(/contour_ter/),False)
 wrf_map_overlay(wks,map,(/contour_t2/),False)
 wrf_map_overlay(wks,map,(/contour_ter,contour_t2/),False)
end do

end
```

a script – *ideal case*

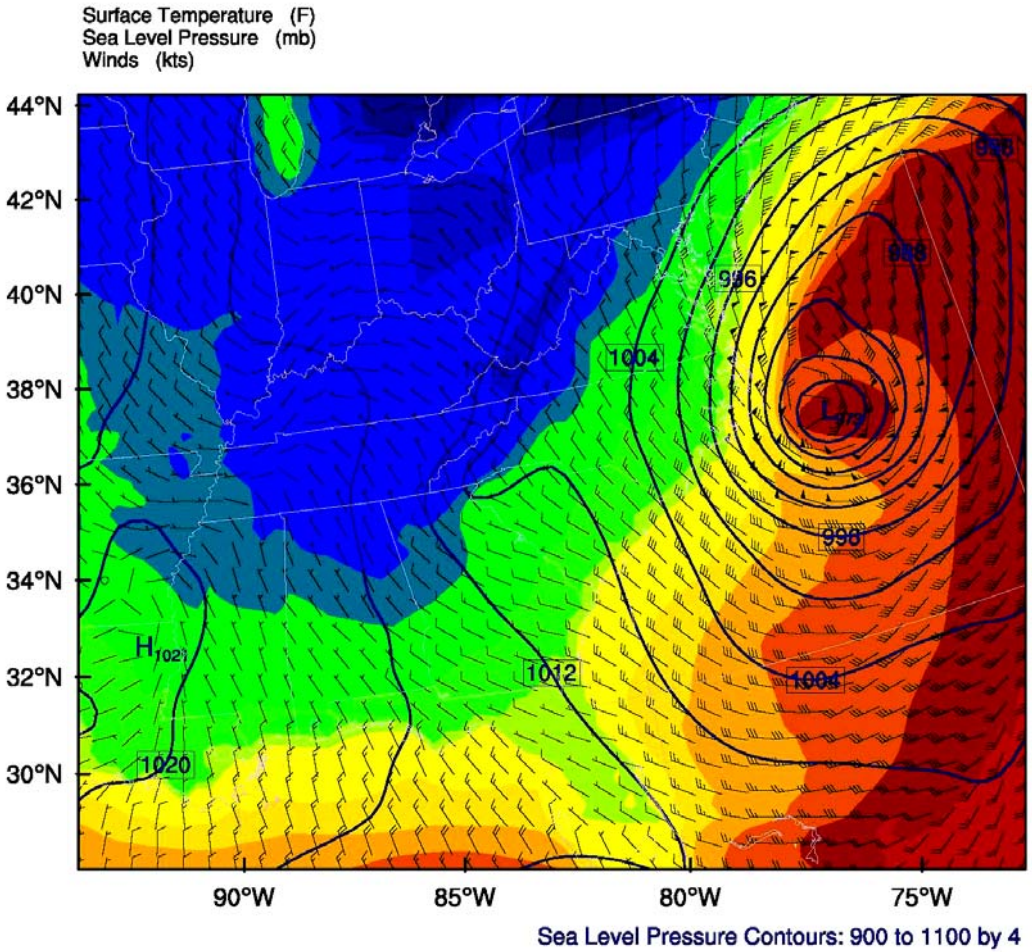
```
do it = 0,ntimes-1 ; loop through file for all times

   res@TimeLabel  = times(it)    ; Set Valid time info
   map = wrf_map(wks,a,res)      ; Create a map background
   ter = wrf_user_getvar(a,"HGT",time)   ; get terrain
   t2  = wrf_user_getvar(a,"T2",time)    ; get T2

   opts_ter = res
   opts_ter@cnFillOn             = True
   contour_ter  = wrf_contour(a,wks,ter,opts_ter)
   opts_t2 = res
   contour_t2  = wrf_contour(a,wks,t2,opts_t2)

   wrf_map_overlay(wks,map,(/contour_ter,contour_t2/),False)
   wrf_overlay(wks,/contour_ter,contour_t2/),False)
end do

end
```

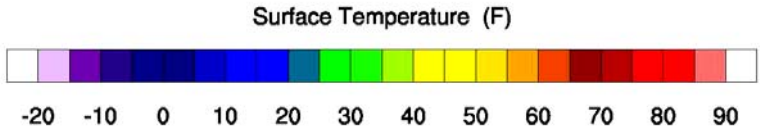
example

```
tc_plane = 1.8*tc2+32.
opts_tc = res
opts_tc@FieldTitle = "Surface Temperature"
opts_tc@UnitLabel          = "F"
opts_tc@ContourParameters  = (/-20.,90.,5./)
opts_tc@cnFillOn           = True
contour_tc=wrf_contour(a,wks,tc_plane,opts_tc)

opts_psl = res
opts_psl@ContourParameters = (/900.,1100.,4./)
opts_psl@cnLineColor       = "NavyBlue"
opts_psl@cnHighLabelsOn    = True
opts_psl@cnLowLabelsOn     = True
contour_psl=wrf_contour(a,wks,slvl,opts_psl)

u_plane = u10*1.94386
v_plane = v10*1.94386
opts_vct = res
opts_vct@FieldTitle        = "Winds"
opts_vct@UnitLabel         = "kts"
opts_vct@NumVectors        = 47
vector=wrf_vector(a,wks,u_plane,v_plane,opts_vct)

wrf_map_overlay(wks,map,(/contour_tc,contour_psl,vector/),False)
```

REAL-TIME WRF

Init: 2000-01-24_12:00:00
Valid: 2000-01-25_12:00:00





how to run NCL?

- **cp DOThluresfile ~/.hluresfile**
  (change if needed)
- **Make external function**
  WRAPIT wrf_user_fortran_util_0.f
- **Edit WRFOptions.ncl**
  (set header and footer options)
- **Edit NCL script**
  (set input and output files ; set Main Title)
- **Run NCL script**
  ncl *wrf_real.ncl*

News Flash

News Flash

README_FIRST
README_NCL

DOThluresfile

gsn_code.ncl
SkewTFunc.ncl
WRFOptions.ncl
WRFPlot.ncl
WRFUserARW.ncl

~~wrf_user_fortran_util_0.f~~

~~make_ncl_fortran~~
~~make_ncl_fortran.alpha~~
~~make_ncl_fortran.ibm~~
~~make_ncl_fortran.ibm64~~
~~make_ncl_fortran.linux~~
~~make_ncl_fortran.sun~~

wrf_bwave.ncl
wrf_hill2d.ncl
wrf_grav2d.ncl
wrf_qss.ncl
wrf_squall_2d_x.ncl
wrf_squall_2d_y.ncl
wrf_real_input.ncl
wrf_real.ncl
wrf_cloud.ncl

RIP4

Read / Interpolate / Plot (Version 4)

Originally written for MM5 input and output, and recently rewritten and generalized to include WRF

Written by Mark Stoelinga
UW / NCAR

general

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

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

download RIP4 for WRF

- From wrf-model.org web site (*rip4.tar.gz*):

  http://www.mmm.ucar.edu/wrf/users/download/get_source.html

- "How to" and examples:

  http://www.mmm.ucar.edu/wrf/users/graphics/RIP4/RIP4.htm

RIP4 on your computer

- **set RIP_ROOT environment variable**

  setenv RIP_ROOT /usr/$USER/RIP4

- **Edit *Makefile* to define paths to netCDF library and include file on your computer:**

  NETCDFLIB and *NETCDFINC*

- **make <machine type>** (*it'll make suggestions*)

  make linux (*example*)

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

ripdp_wrf

- **ripdp_wrf is *RIP Data Preparation for WRF***

- **RIP does not read WRF data directly**
- ***ripdp_wrf*** converts WRF netCDF data into RIP input format *(format described in the document)*
- **RIP puts each variable at each time into a separate file – LOTS of files**

mkdir storm_case

running ripdp_wrf

Optional
ripdp_sample.in

- ripdp_wrf [-n namelist-file] \
  <model_data_name> [basic/all] \
  <input_file1 input_file2>

Example

ripdp_wrf storm_case/test basic wrfout

use directory as part of the model_data_name

ripdp_wrf namelist

```
&userin
   ptimes=0,-72,1,ptimeunites='h',tacc=90.,
              discard='LANDMASK',H2SO4',
   iexpandedout=0
&end
```

- **Use namelist to add control**
  - ptimes – times for *ripdp_wrf* to process
    - *0,1,2*
    - *0,-72,1*
    - *0, 3,-24,3, 48*
  - discard fields if '*all*' is selected on the command line

rip

- **read the output generated by *ripdp_wrf***
- **read User Input File (UIF) *(rip_sample.in)***
  - **First** section is a list of general parameters *(namelist format)*
  - **Second** section is a series of plots in the Plot Specification Table (PST)
- **generate meta file**

running rip

- Edit the User Input File (UIF)
- setenv NCARG_ROOT /usr/local/ncarg
- setenv RIP_ROOT *your-rip-directory*

running rip

created by
ripdp_wrf

- **rip [-f] model-data-set-name **
  rip-execution-name

User Input File (UIF)

Example

rip -f storm_case/test rip_sample.in

use directory as part of the model_data_set_name

output ; metacode

rip_sample.out
rip_sample.ncgm

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**

rip namelist - userin

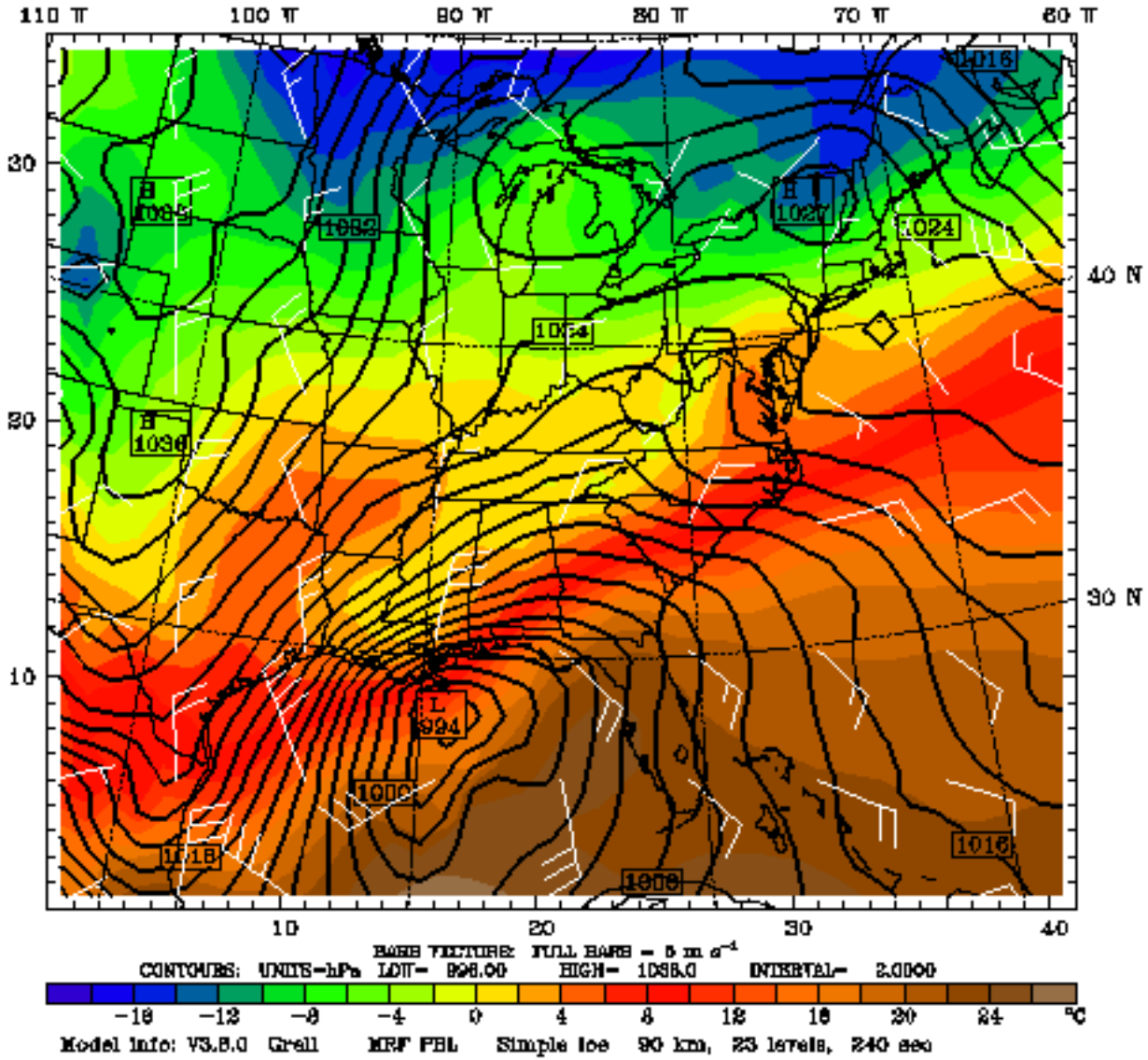
- idotitle – first part of first title line
- titlecolor – color of title lines
- ptimes, ptimeunits – times to process
- tacc – tolerance for processing data
- timezone –display of local time
- iusedaylightrule – 1 applied, 0 not applied
- iinittime – plotting of initial time
- ivalidtime – plotting of valid time
- inearsth – plot times as 2 / 4 digits
- flmin, frmax, fbmin, ftmax – frame size
- ntextq – text quality

rip namelist - userin

- ntextcd – text font
- fcoffset – 12 means hour 12 of the MM5 forecast is considered hour 0 by you
- idotser – generate time series output
- idescriptive – more descriptive titles
- icgmsplit – split metacode into several files
- maxfld – reserve memory for RIP (10-15)
- itrajcalc – 0, 1 ONLY when doing trajectory calculations *(use also namelist trajcalc)*
- imakev5d – 0, 1 generate Vis5D data

example

```
=========================================
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, creates

GrADS ***(.dat)*** and ***(.ctl)*** files, or
Vis5D input files

download ARWpost

- From wrf-model.org web site (*ARWpost.tar*):

  http://www.mmm.ucar.edu/wrf/users/download/get_source.html

- "How to" and examples:

  http://www.mmm.ucar.edu/wrf/users/graphics/ARWpost/ARWpost.htm

general

- **MUST have WRFV2 compiled**

- **IF Vis5D files are being created,**
  - Vis5D libraries needed for compilation
  - Vis5D is free and can be downloaded from: http://www.ssec.wisc.edu/~billh/vis5d.html

code

README
arch/
clean*
compile*
configure*
fields.plt
gribinfo.txt
gribmap.txt
myLIST
namelist.ARWpost
scripts/
src/
util/

compilation

./configure

```
Will use NETCDF in dir: /usr/local/netcdf-pgi
-------------------------------------------------
Please select from among the following supported
platforms.
1. PC Linux i486 i586 i686, PGI compiler (no
vis5d)
2. PC Linux i486 i586 i686, PGI compiler (vis5d)
3. PC Linux i486 i586 i686, Intel compiler (no
vis5d)
4. PC Linux i486 i586 i686, Intel compiler
(vis5d)

Enter selection [1-4] :
```

compilation

- **configure.arwp,** will be created
- If your WRFV2 code is not compiled under **../WRFV2,** edit **configure.arwp,** and set **“WRF_DIR“** to the correct location of your WRFV2 code
- **./compile**

  This will create **ARWpost.exe**

namelist.ARWpost

- **set times to be processed** *(&datetime)*
- **set input and output file names and variables to be processed** *(&io)*
- **set levels to interpolate too** *(&interp)*

namelist.ARWpost *(&io)*

io_form_input	2=netCDF, 5=GRIB1
input_root_name	Path and root name of files to use as input. Do not use wild characters in the input_root_name.
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 *(&io)*

plot	Which fields to process. Options are: **all, basic, list, file, basic_file, basic_list, list_file, all_file, all_list, basic_list_file, all_list_file** Order has no effect, i.e., “all_list” and “list_all” are similar. "**list**" - a list of variables must be supplied under "**fields**" "**file**" - a list of variables must be added to a file, and the filename supplied under "**fields_file**”
fields	Fields to plot. Only used is **list** was used in the “**plot**” variable.
fields_file	File name that contains list of fields to plot. Only used is **file** was used in the “**plot**” variable.

***Available diagnostics*:** height ,theta ,tc, tk, td, rh, umet, vmet, pressure, dbz, max_dbz, u10m, v10m, slp, mcape, mcin, lcl, lfc, cape, cin

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) Supply levels bottom to top

run

- ./ARWpost.exe
- Will create either,

output_root_name.dat & *output_root_name*.ctl

OR

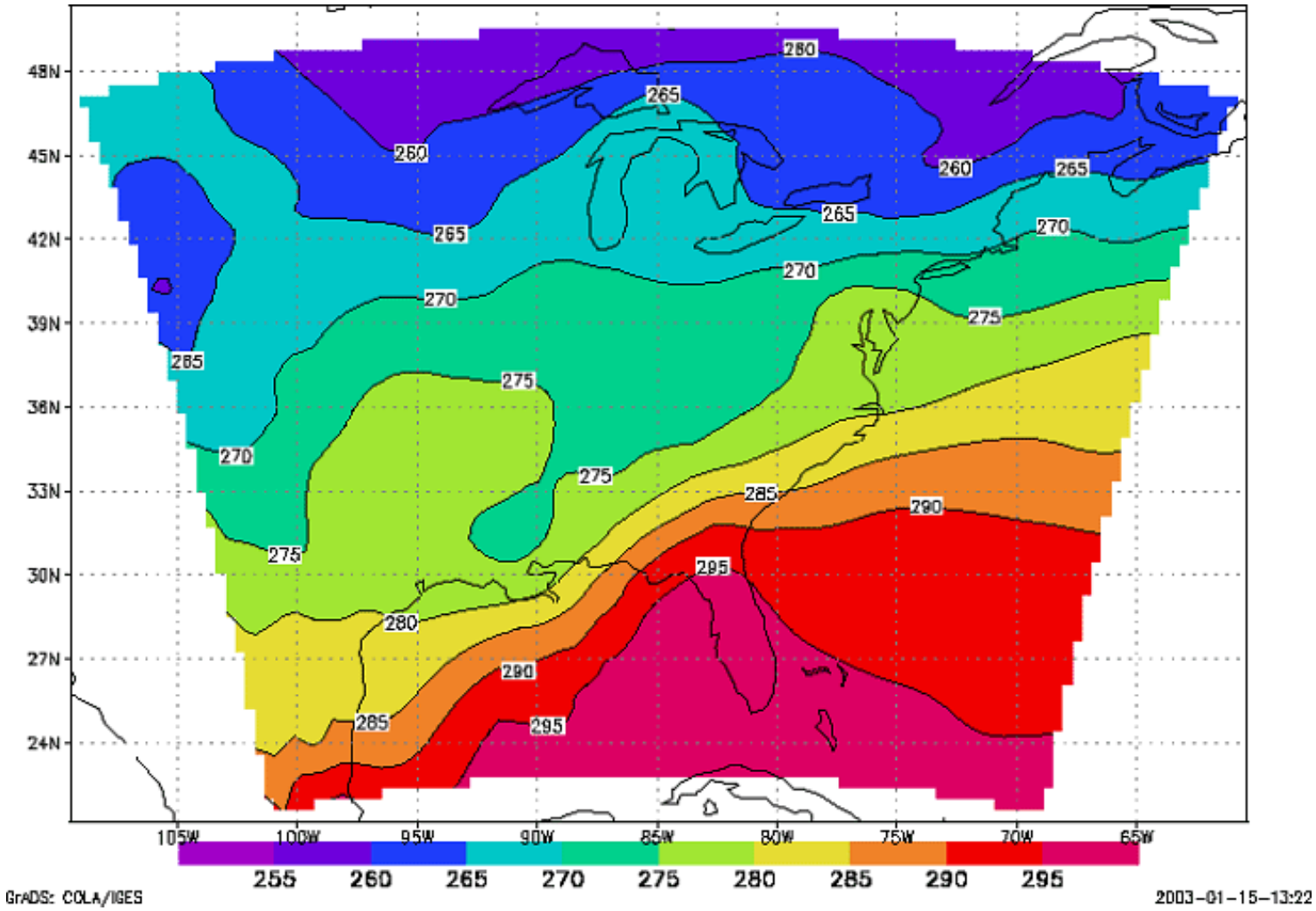
output_root_name.v5d

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

GrADS - projections



GrADS conversion - question

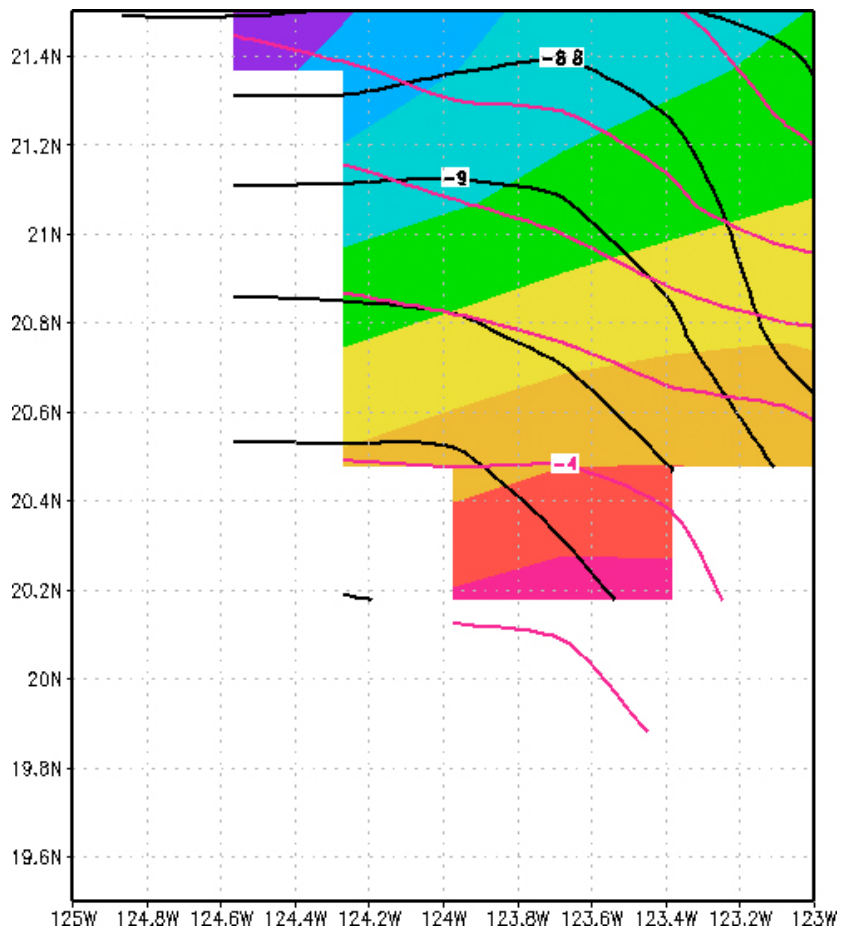
- **Why is a converter needed if GrADS can display netCDF files?**
  - Can only display model surface coordinates
  - Cannot interpolate to height or pressure levels
  - All diagnostics must be added via GrADS script files
  - *GRIB1 model output can also be read directly by GrADS, but above issues are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

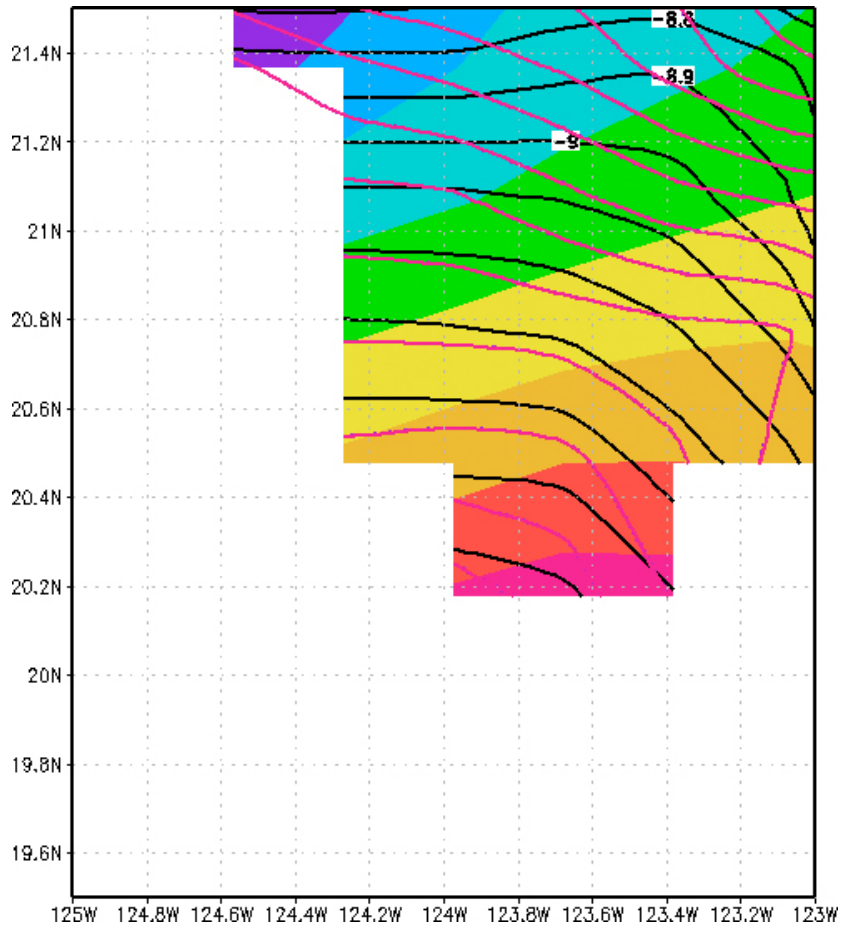
GrADS conversion - question

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

staggering

shaded=T ; black=U ; red=V



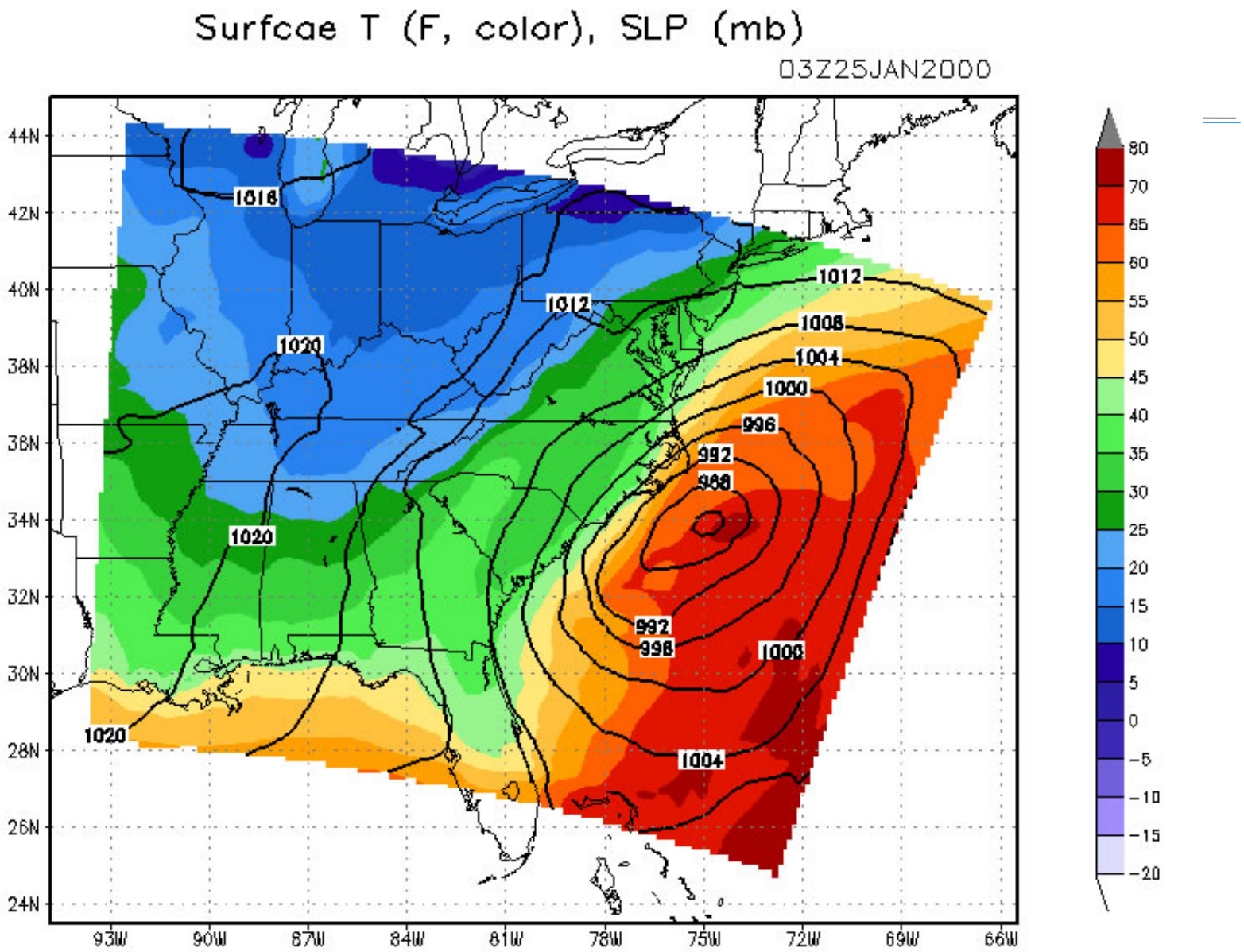


staggering

- **Since GrADS version 1.9**
  - a new **gradsnc** interface is available *created by developers for WRF*
- **To USE**
  - must create 4 .ctl files (M ; U ; V ; W)
  - must open the all at once
- **Utility**
  - ARWpost/util/WRFnc2ctl.f

GrADS script example

```
'open real.ctl‘
'set mpdset hires'
'set display color white'
'run rgbset.gs'

'define tf=1.8*tc + 32‘
'set gxout shaded'
'set clevs -20 -15 -10 -5  0  5 10 15 20 25 30 35 40
  45 50 55 60 65 70 75 80'
'set ccols   0  51  53 55 57 58 59 49 47 45 39 37 36
  34 22 23 24 25 27 29 29'
'set z 1'
'd tf‘
'run cbar.gs‘

'set gxout contour'
'set ccolor 1'
'set cint 4'
'd slvl‘
'draw title Surface T(F,color), SLP(mb)'
```


Surfcae T (F, color), SLP (mb)
03Z25JAN2000
44N
42N
40N
38N
36N
34N
32N
30N
28N
26N
24N
93W
90W
87W
84W
81W
78W
75W
72W
69W
66W
1016
1012
1020
1008
1004
1000
996
992
988
1020
992
998
1008
1020
1004
80
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
-5
-10
-15
-20

Vis5D Specific Notes

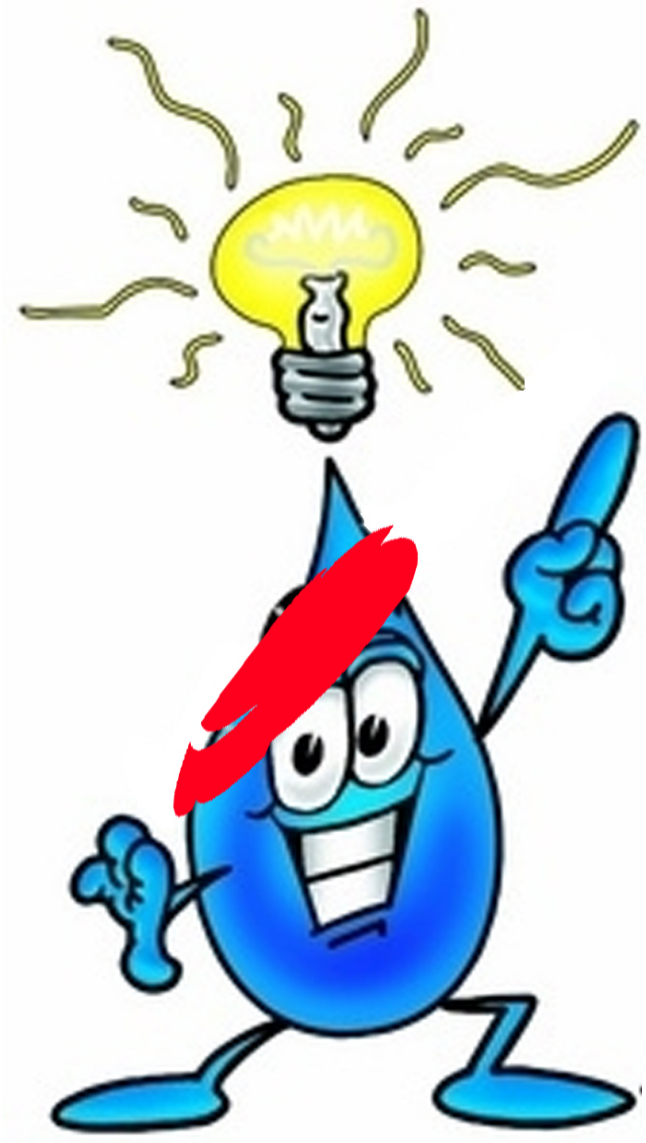
- **Vis5D is a three-dimensional visualization software**
- **Vis5D is free and can be downloaded from: http://www.ssec.wisc.edu/~billh/vis5d.html**
- **Run**

  vis5d *output_root_name*.v5d

- **Graphical Interface**

NCL

RIP4



GrADS

Vis5D