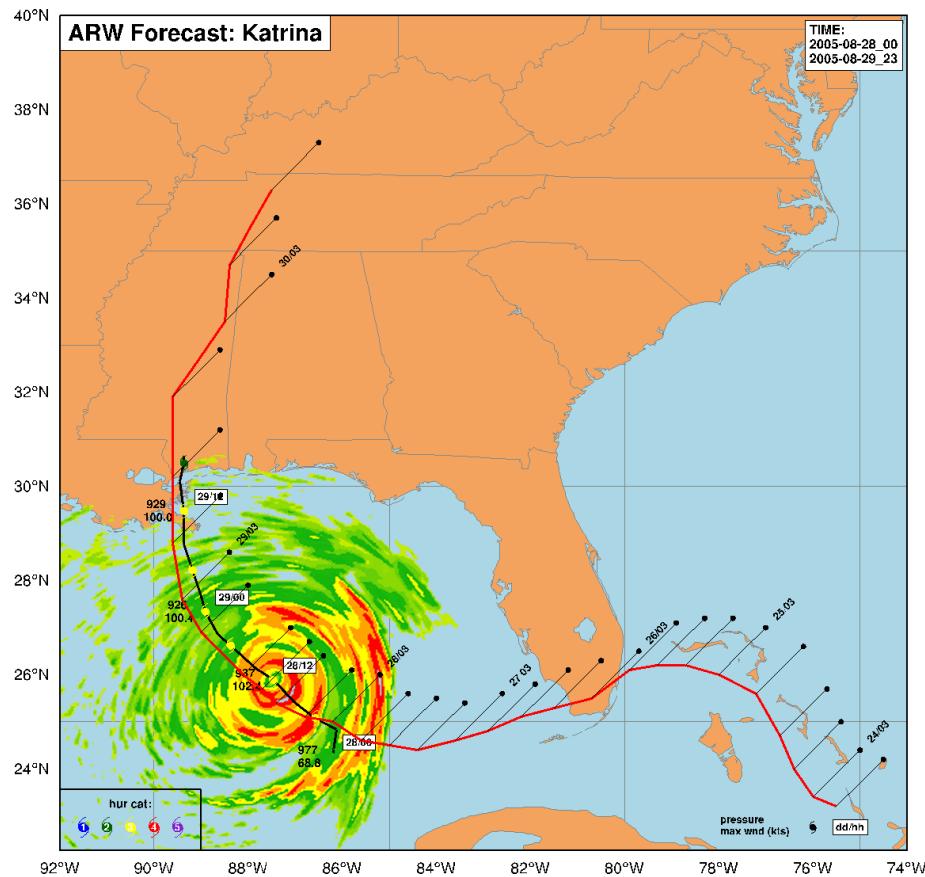


Post-processing Tools: NCL

Abby Jaye

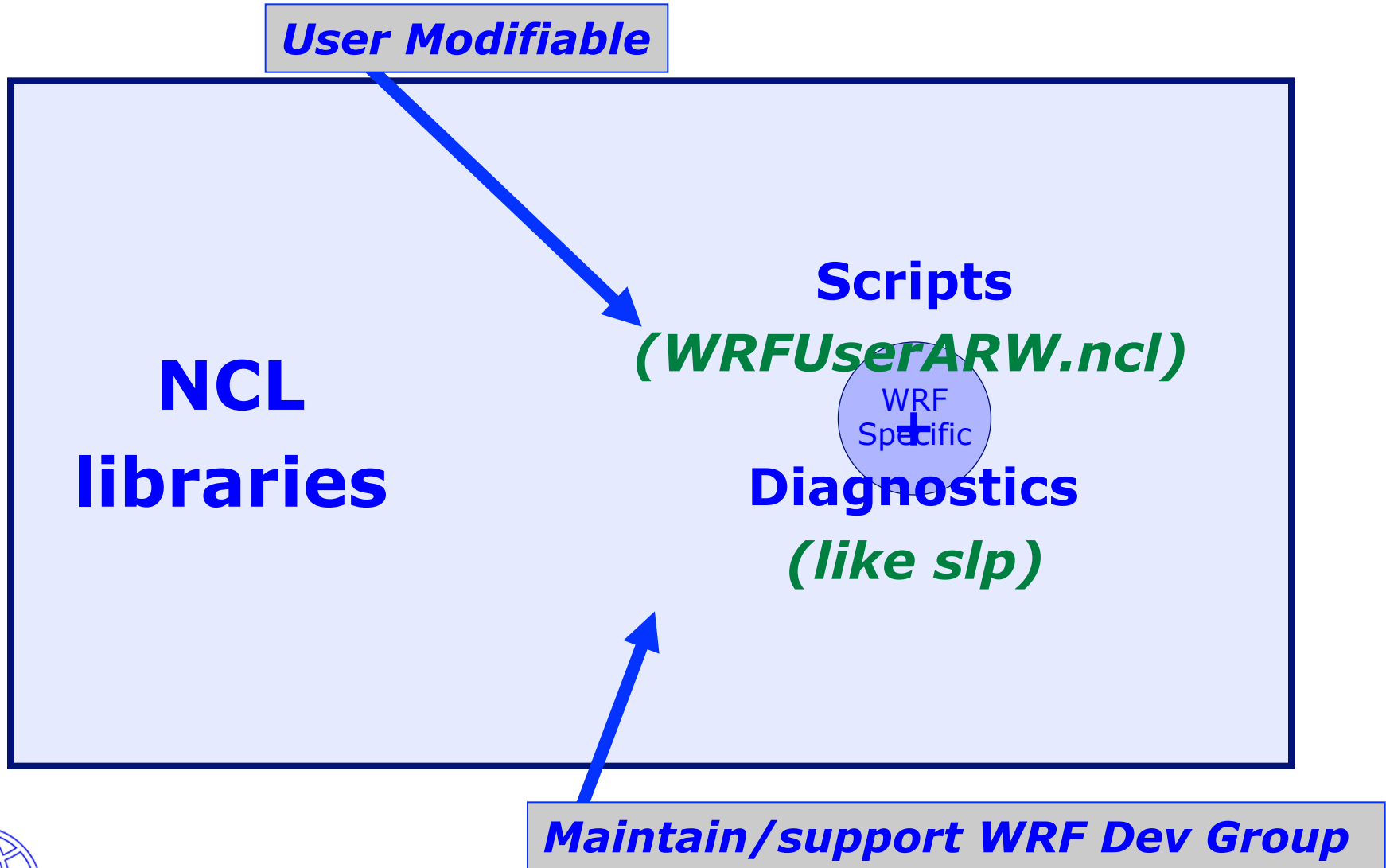


NCL

- 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 6 or *later* (current v6.3.0 – highly recommended)
- NCARG_ROOT environment variable
 - `setenv NCARG_ROOT /usr/local/ncl`



NCL & WRF



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>
- Less important in NCL version 6 and higher

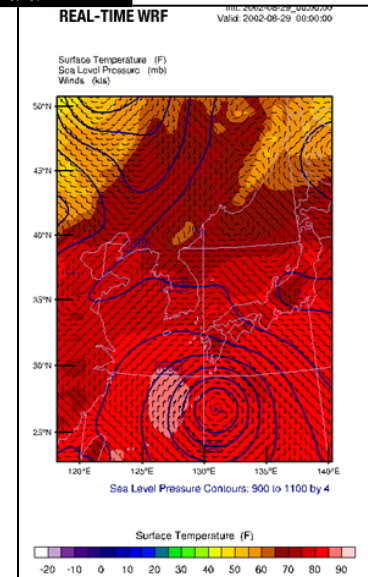
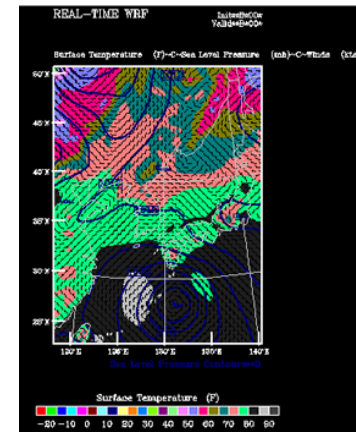


~/.hluresfile

```
*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](http://www.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/.hluresfile)



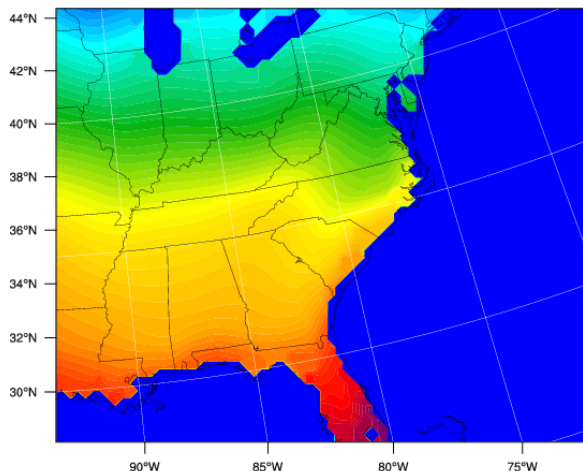
Generate Plots with NCL

- 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)
Most of the WRF script routines are called from
`"$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"`
Feel free to add or change this script
- Run NCL script
 - `ncl wrf_real.ncl`

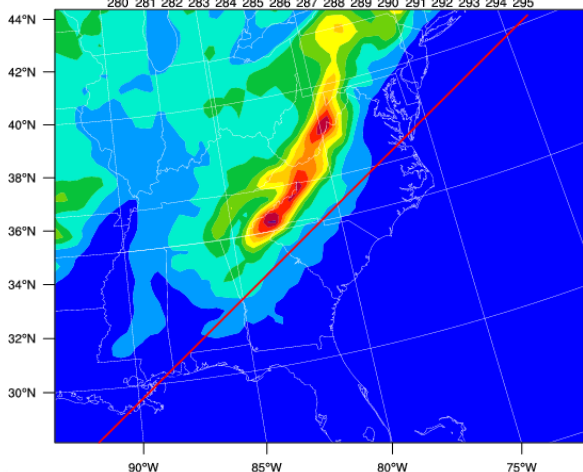


Examples

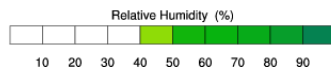
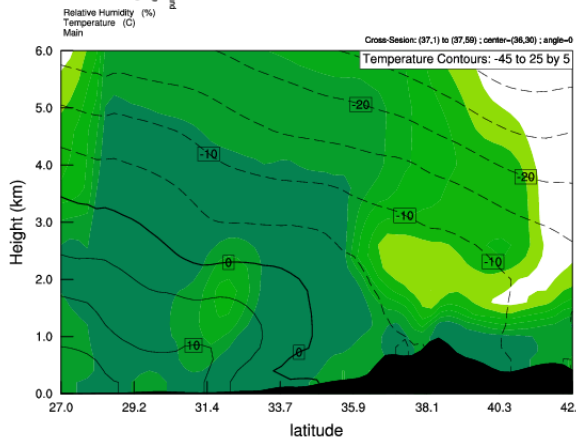
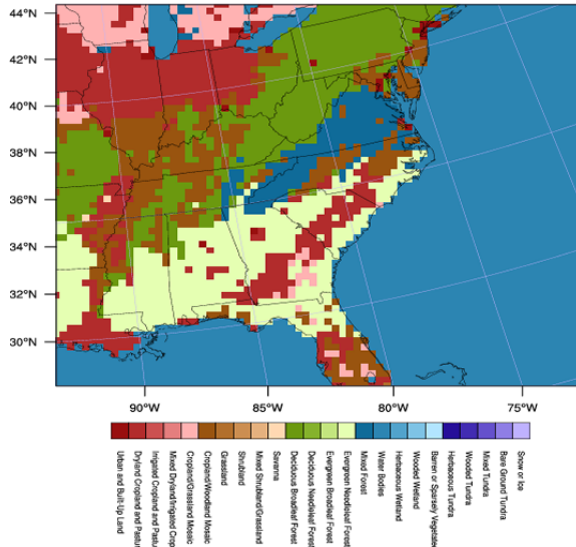
SOILTEMP : Annual mean deep soil temperature (Kelvin)



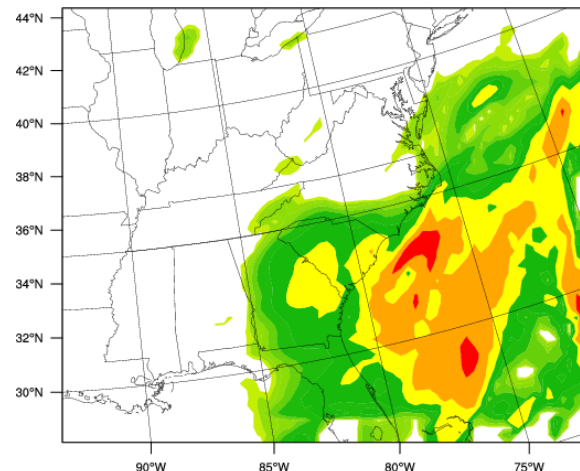
SOILTEMP : Annual mean deep soil temperature (Kelvin)



LU_INDEX : Dominant category (category)



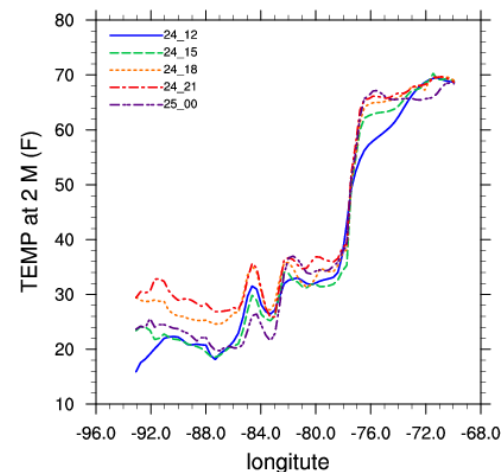
Total Precipitation (mm)



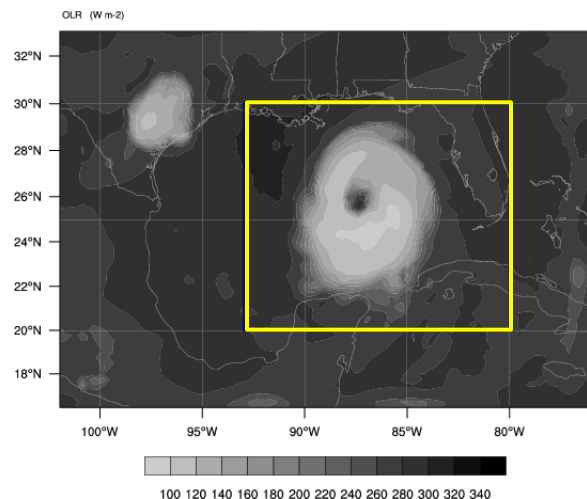
Total Precipitation (mm)



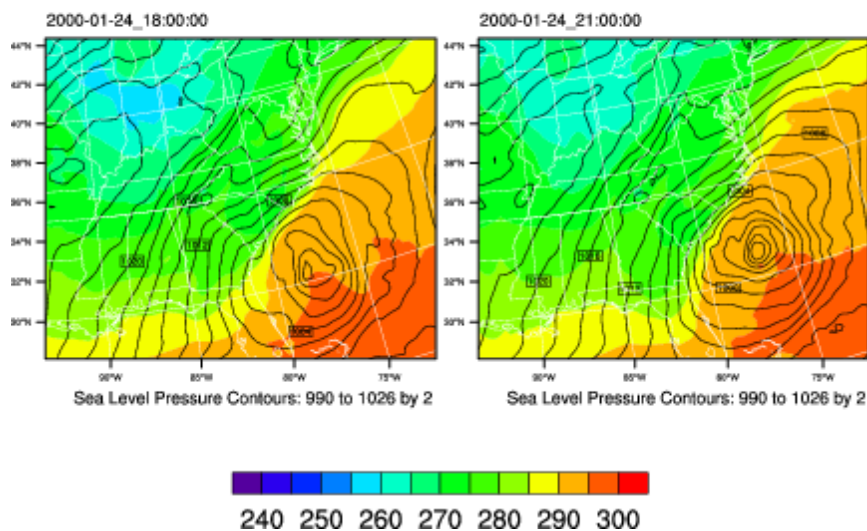
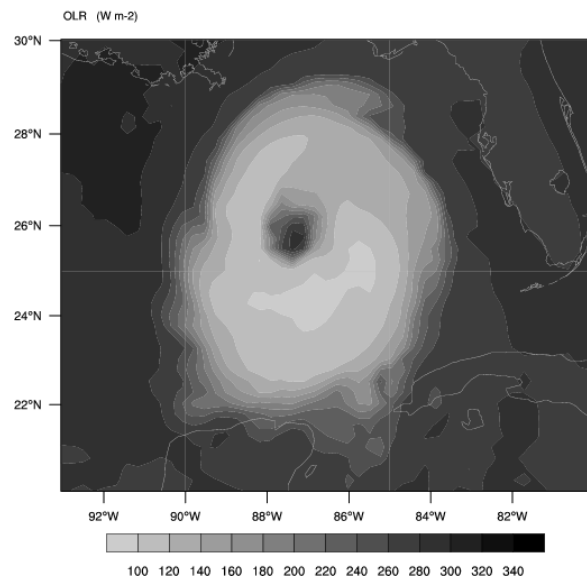
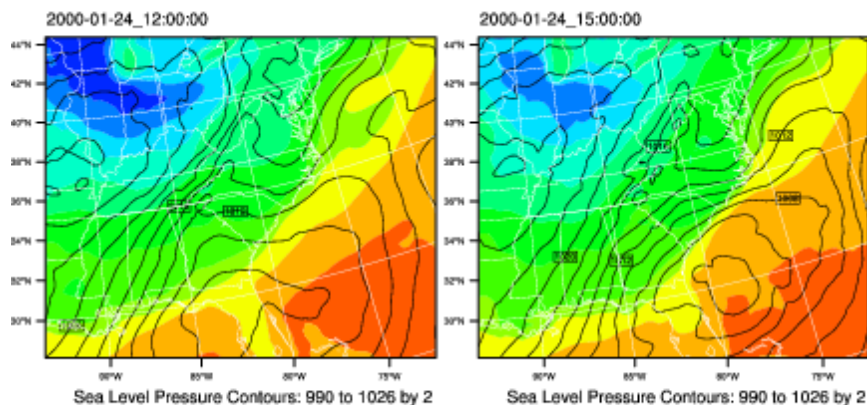
Surface temperature along a cross section



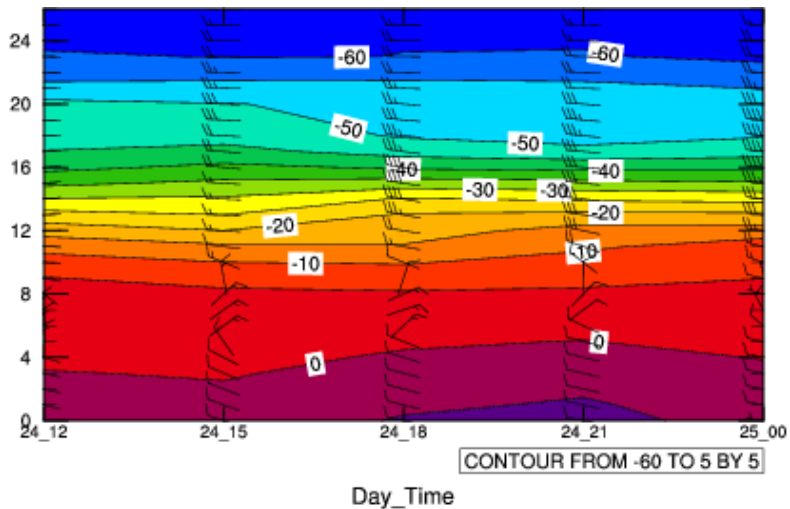
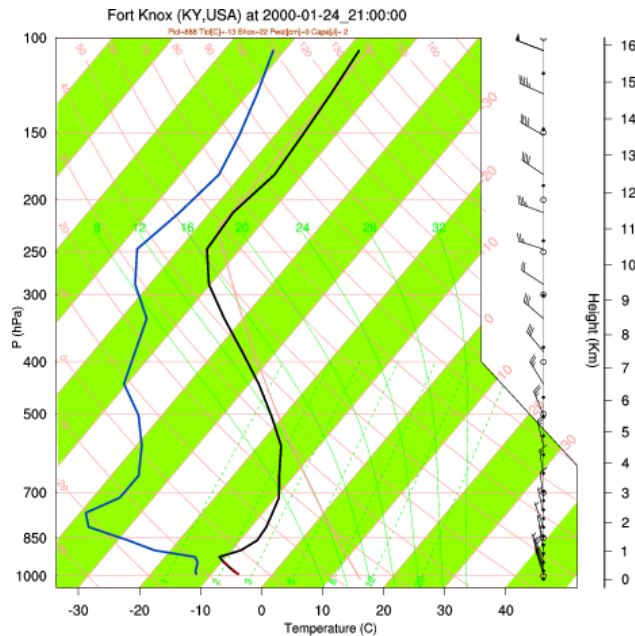
Examples



TEMP at 2 M (K)

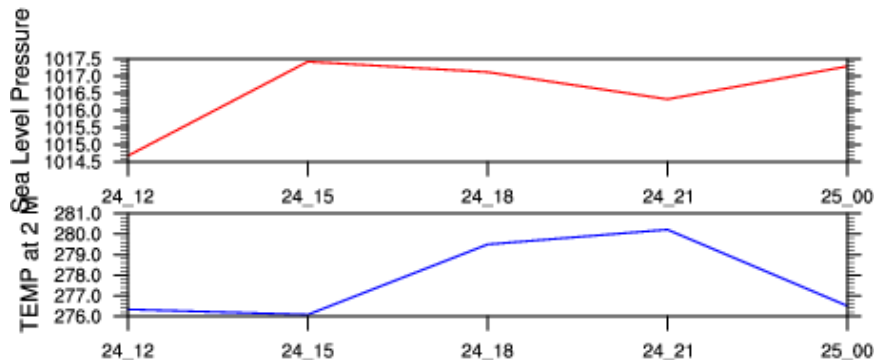
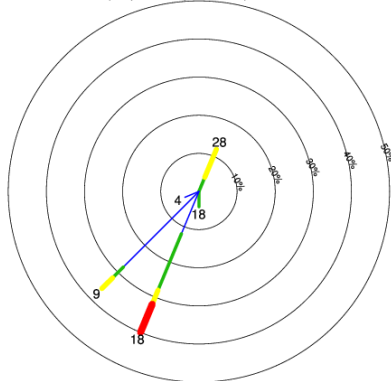


Examples

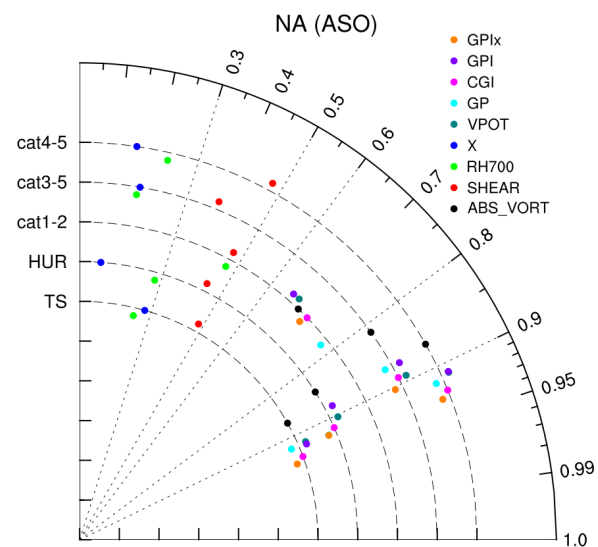
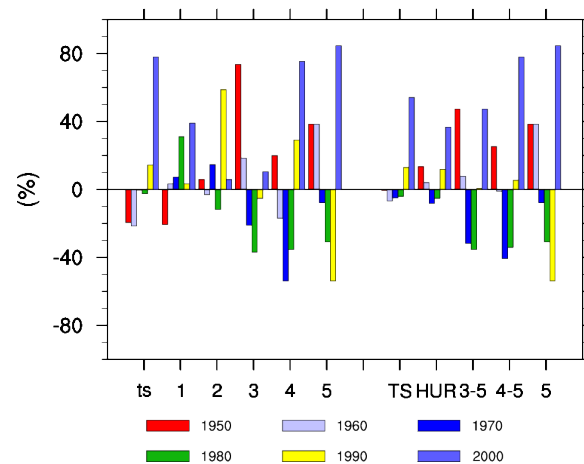
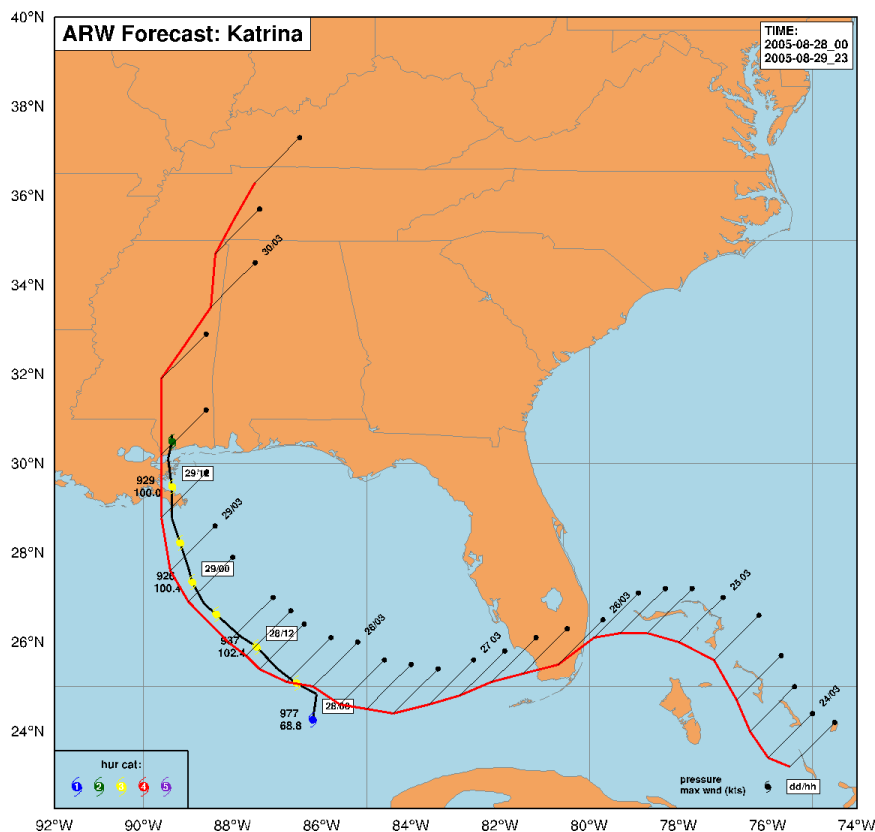


WRF: All Times: grid point [25.65 , -87.37]

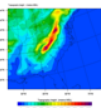
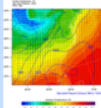
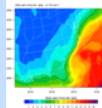
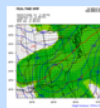
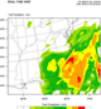
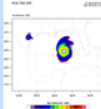
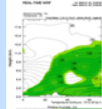
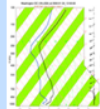
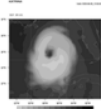
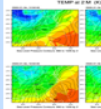
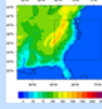
SpdAve=16 SpdStd=11 DirAve=216 No Calm Reports Nwnd=25
Frequency dots every 10%. Mean speed indicated.



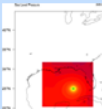
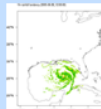
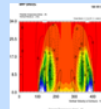
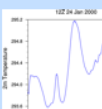
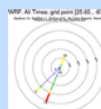
Examples



Generate Plots: *A good start - OnLine Tutorial*

 Basic Plots Basic Plot Setup (This series of examples takes users through same basic steps in generating plotting scripts.) Get and plot a single field Multiple input files Plot all fields in a file	 Basic Surface Plots Surface 1 Surface 3 Surface 2 Surface with multiple input files	 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 PW Vorticity helicity updraft_helicity (More diagnostics 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 Smooth terrain	 Skew_T Plots Skew_T
 Overlay and Zoom Overlay Zoom Overlay & Zoom	 Panel Plots Panel 1 Panel 2	 Overlay Domains Overlay 2 domains	

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

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



Creating a Plot : NCL script

```
load ncl library scripts
```

```
begin
```

```
  ; Open input file(s)
```

```
  ; Open graphical output
```

```
  ; Read variables
```

```
  ; Set up plot resources & Create plots
```

```
  ; Output graphics
```

```
end
```



Creating a Plot : NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"  
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"  
  
begin  
  
; Open input file(s)  
; Open graphical output  
  
; Read variables  
  
; Set up plot resources & Create plots  
; Output graphics  
  
end
```

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



Creating a Plot : NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"  
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"  
  
begin  
  
; Open input file(s)  
; Open graphical output  
  
; Read variables  
  
; Set up plot resources & Create plots  
; Output graphics  
  
end
```



Creating a Plot : NCL script

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

begin

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

; Read variables

; Set up plot resources & Create plots
; Output graphics

end
```

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

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

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



Creating a Plot : NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"  
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"  
  
begin  
  
  a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")  
  wks = gsn_open_wks("X11","plt_Surface")  
  
  ; Read variables  
  
  ; Set up plot resources & Create plots  
  ; Output graphics  
  
end
```

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



Creating a Plot : NCL script

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

begin

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

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

  ; Set up plot resources & Create plots
  ; Output graphics

end
```

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

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



Special WRF NCL Functions

wrf_user_getvar

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



Creating a Plot : NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"  
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"  
  
begin  
  
  a = addfile("./wrfout_d01_2012-09-28_00:00:00.nc","r")  
  wks = gsn_open_wks("X11","plt_Surface")  
  
  T2 = wrf_user_getvar(a,"T2",0)  
  
  ; Set up plot resources & Create plots  
  ; Output graphics  
  
end
```



Creating a Plot : NCL script

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

begin

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

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

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

end
```

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

opts: Resources associated with each
individual plot



Creating a Plot : NCL script

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

begin

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

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

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



Creating

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

```
begin
```

```
a = addfile("./wrfout  
wks = gsn_open_wks("\}
```

```
T2 = wrf_user_getvar(
```

```
pltres = True
```

```
mpres = True
```

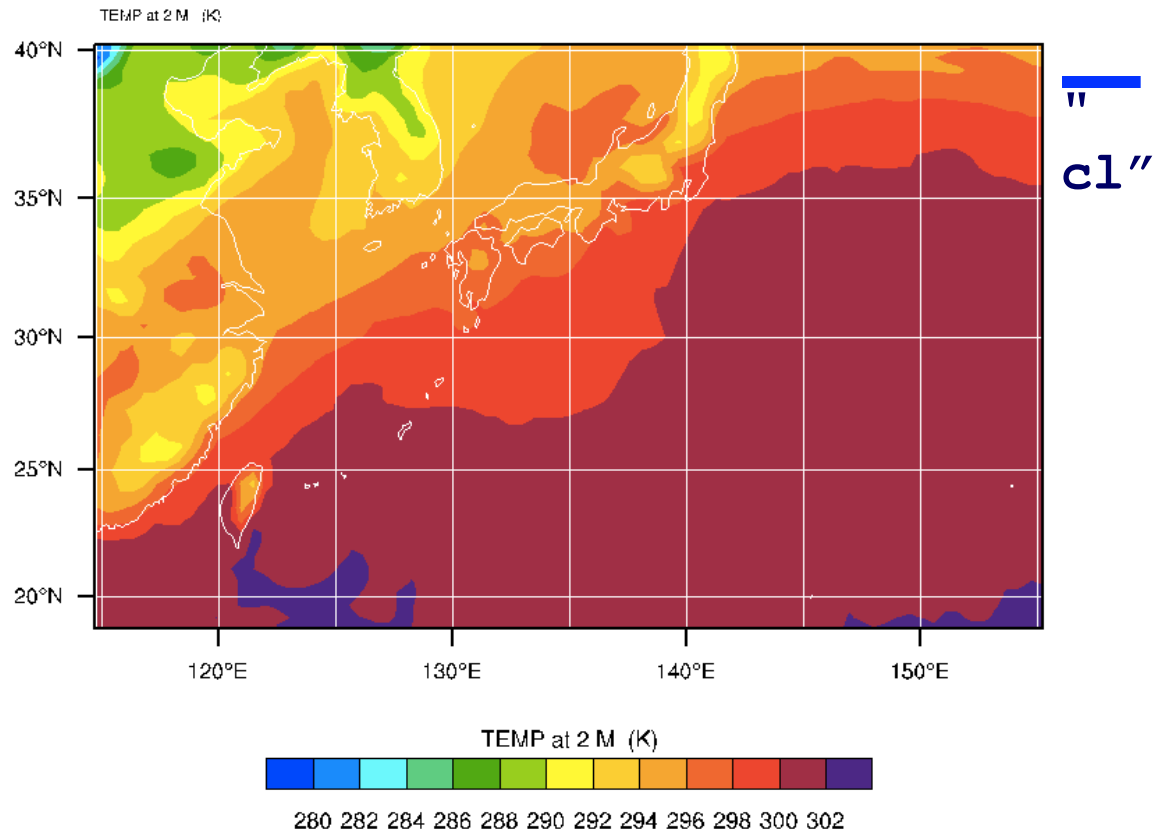
```
opts = True
```

```
opts@cnFillOn = True
```

```
contour_t2 = wrf_contour(a,wks,T2,opts)
```

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

```
end
```



"
c1"



Creating a Plot : NCL script

```
T2 = wrf_user_getvar(a,"T2",0)
slp = wrf_user_getvar(a,"slp",0)
pltres = True
mpres = True
```

```
opts = True
opts@cnFillOn = True
contour_t2 = wrf_contour(a,wks,T2,opts)
delete(opts)
opts = True
opts@cnLineColor = "Blue"
contour_slp = wrf_contour(a,wks,slp,opts)
delete(opts)
```

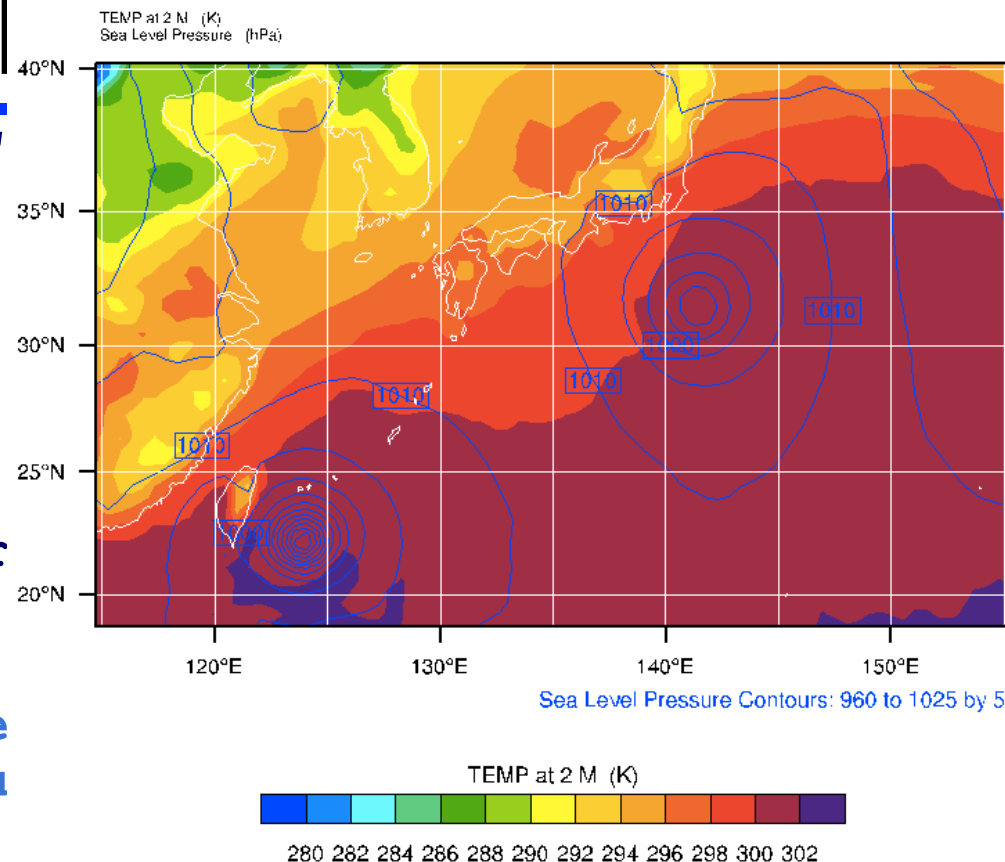
```
plot = wrf_map_overlays(a,wks, (/contour_t2,contour_slp/),
pltres,mpres)
```



Creating a

```
T2 = wrf_user_getvar(a,"
slp = wrf_user_getvar(a,
pltres = True
mpres = True
```

```
opts = True
opts@cnFillOn = True
contour_t2 = wrf_contour
delete(opts)
opts = True
opts@cnLineColor = "Blue
contour_slp = wrf_contou
delete(opts)
```

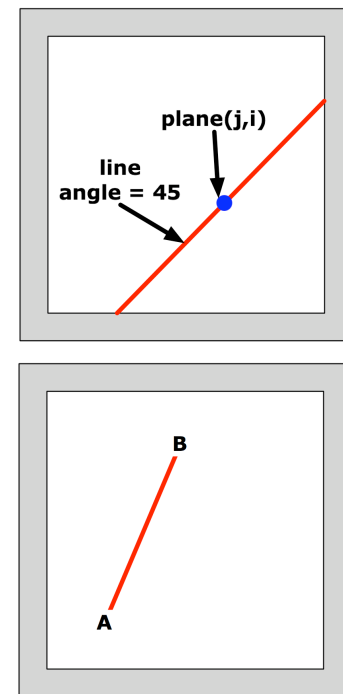


```
plot = wrf_map_overlays(a,wks, (/contour_t2,contour_slp/),
pltres,mpres)
```



Special WRF NCL Functions

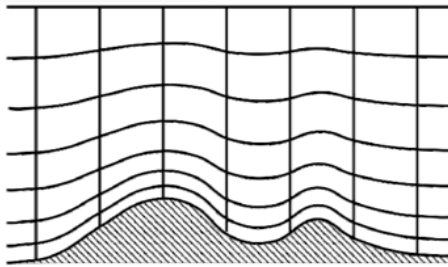
- `wrf_user_getvar`
Get native and diagnostic variables
- `wrf_contour` / `wrf_vector`
Create line/shaded & vector plots
- `wrf_map_overlays` / `wrf_overlays`
Overlay plots created with `wrf_contour` and `wrf_vector`
- `wrf_user_intrp3d` / `wrf_user_intrp2d`
*Interpolate horizontally to a given pressure/height
(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_unstagger`
Unstagger an array
- `wrf_user_vert_interp`



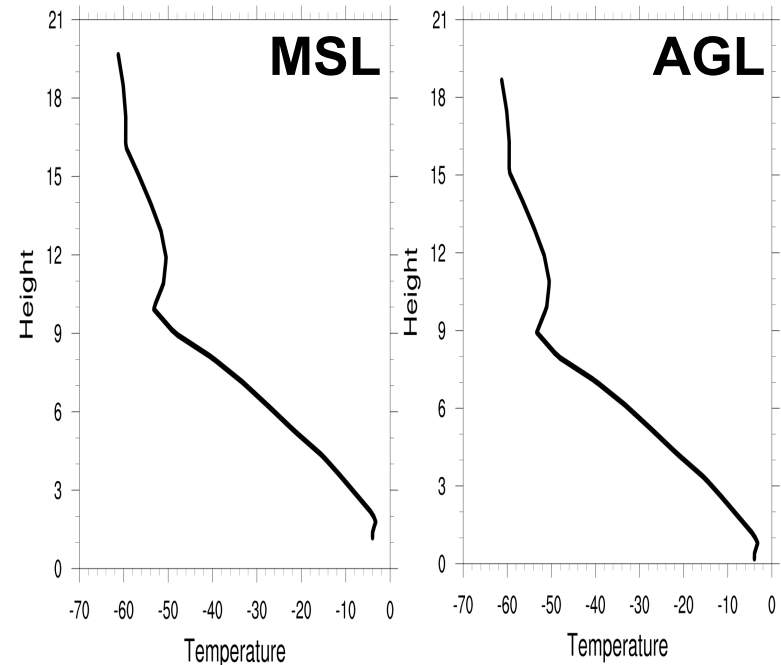
• `wrf_wps_read_int` / `wrf_wps_write_int`

wrf_user_intrp3d

- `tc_plane = wrf_user_intrp3d(tc,z,"v",plane,angle,opts)`
 - Interpolate “tc” to “z”, which by definition is Height Above Mean Sea Level



```
z_AGL = z
dims = dimsizes(z)
do k=0,dims(0)-1
  z_AGL(k,:,:) = z_AGL(k,:,:) - ter(:,:)
end do
```



wrf_user_vert_interp

- Interpolate to:
 - "pressure", "pres" – pressure [hPa]
 - "ght_msl" – grid point height msl [km]
 - "ght_agl" – grid point height agl [km]
 - "theta" – potential temperature [K]
 - "theta-e" – equivalent potential temperature [K]
- Extrapolate below the ground
 - Resource – `opts@extrapolate`

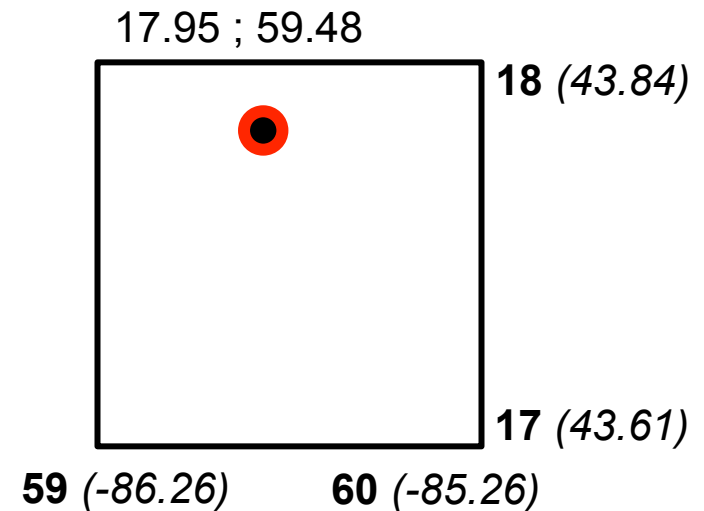
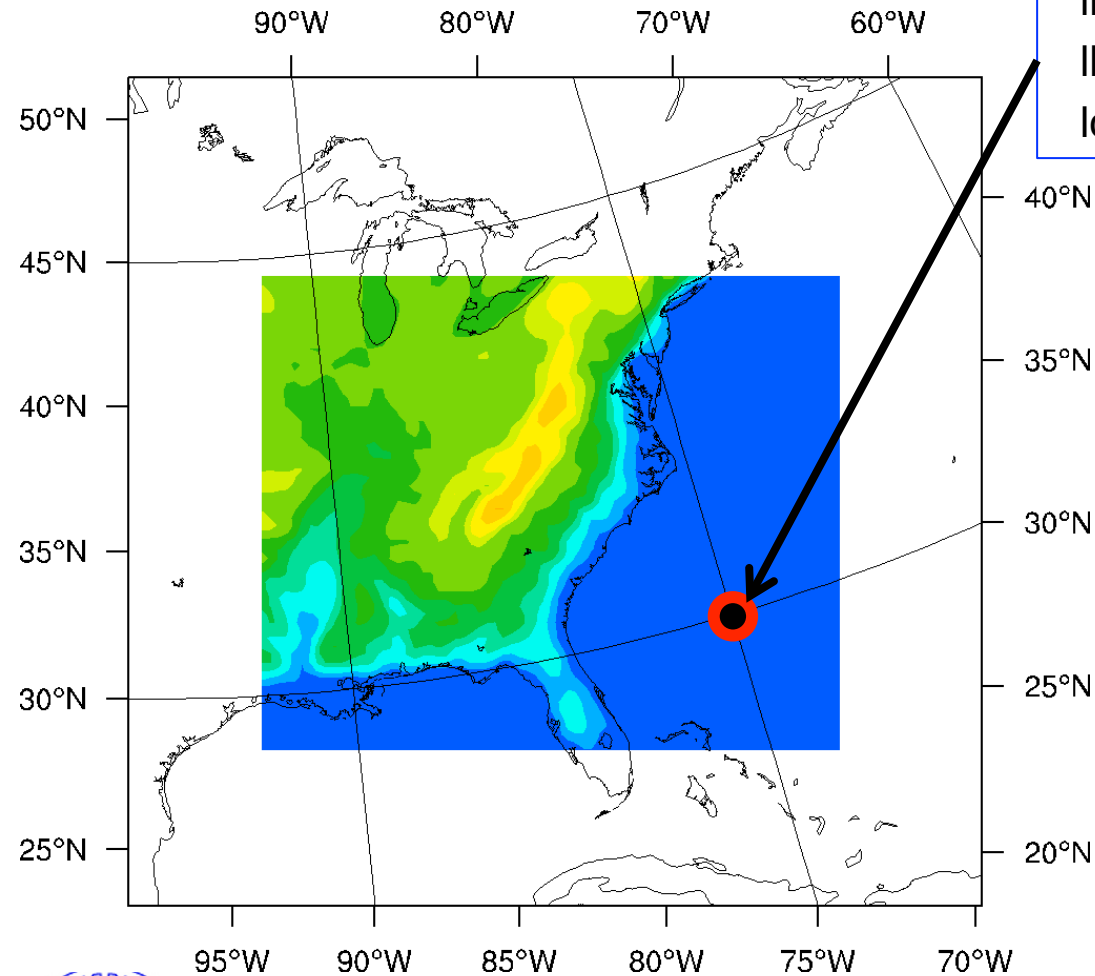


wrf_user_ll_to_ij

llres = True

llres@returnInt = **False**

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

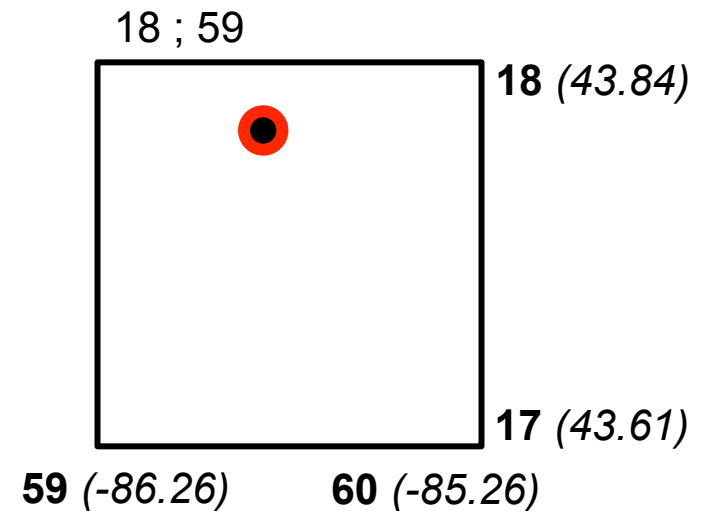
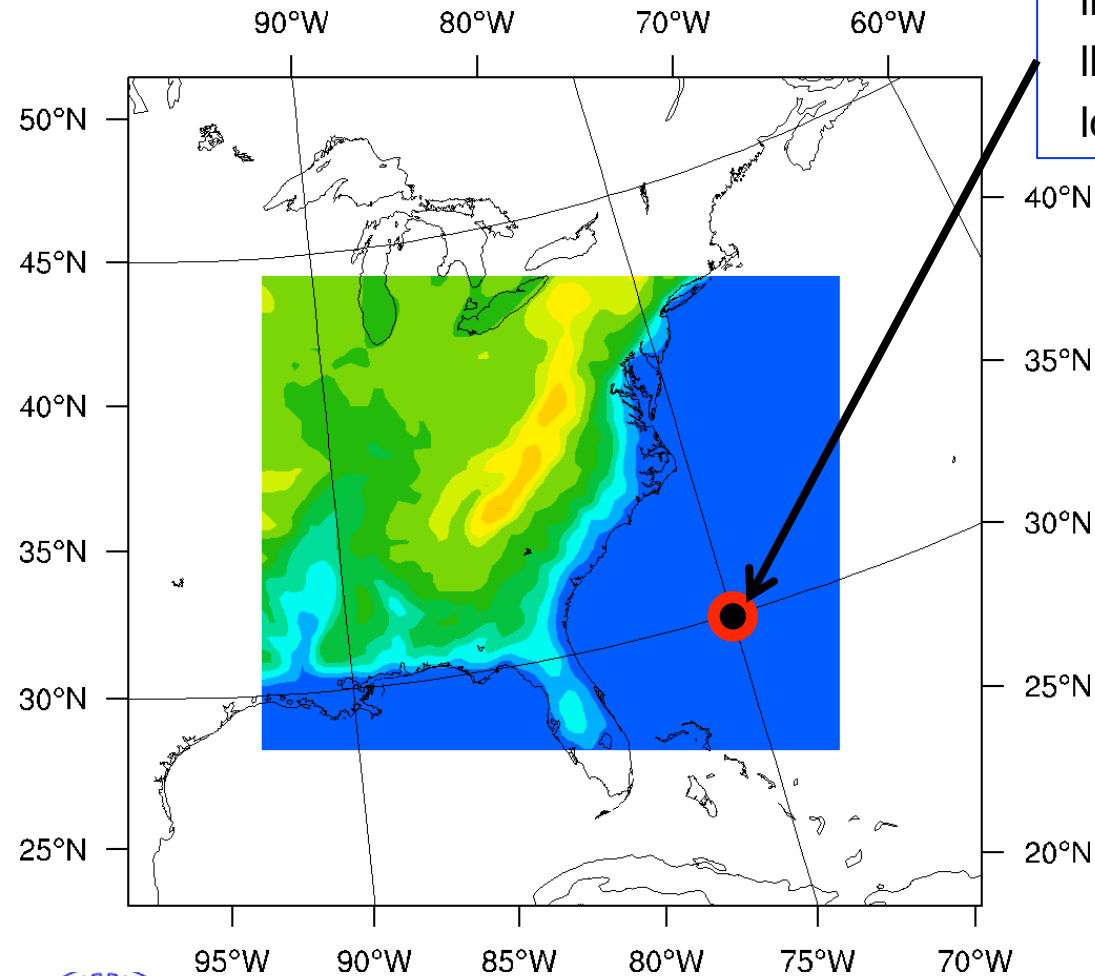


wrf_user_ll_to_ij

llres = True

llres@returnInt = **True**

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



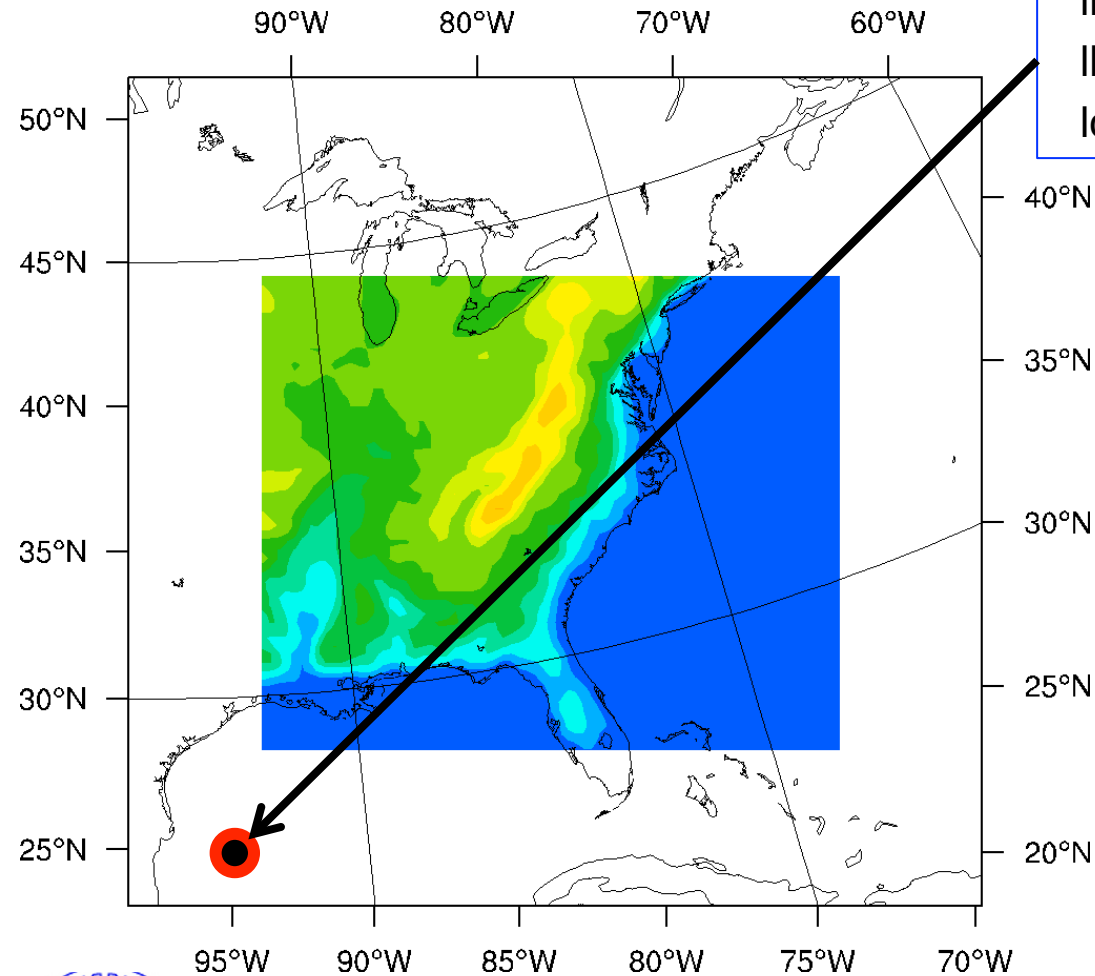
wrf_user_ll_to_ij

llres = True

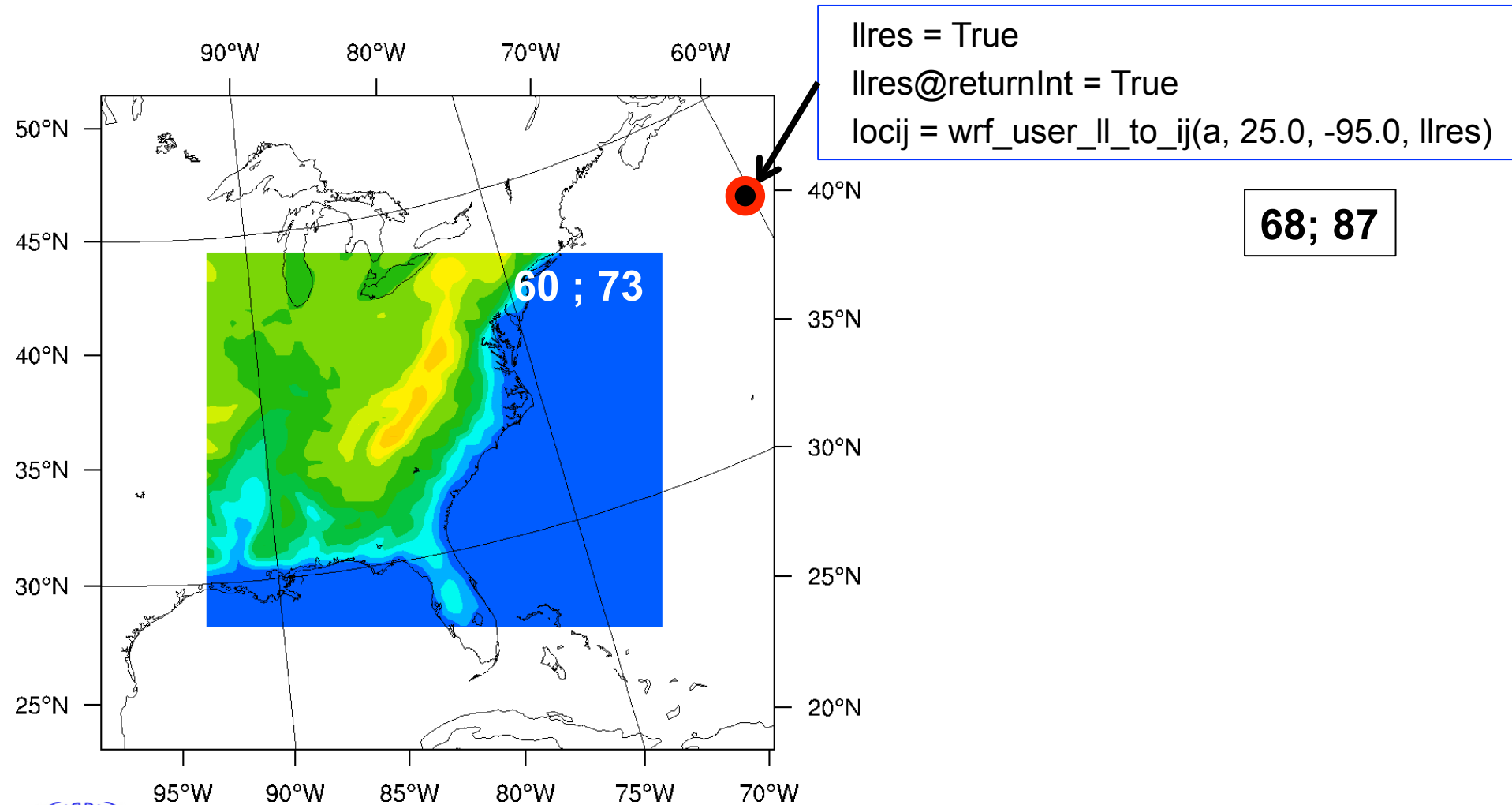
llres@returnInt = True

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

-10 ; -2



wrf_user_ll_to_ij

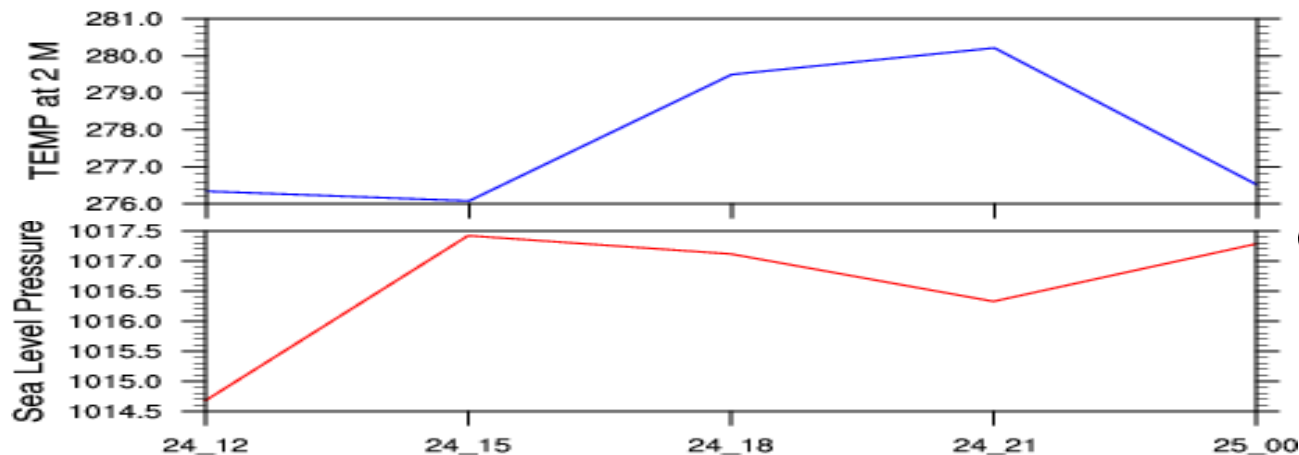


Time Series

```
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

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

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

```
mpres@mpGeophysicalLineColor  
mpres@mpGridLineColor  
mpres@mpNationalLineColor  
mpres@mpUSStateLineColor
```

```
mpres@mpOutlineBoundarySets  
  "NoBoundaries" ; "Geophysical"  
  "National" ; "USStates"  
  "GeophysicalAndUSStates"  
  "AllBoundaries"
```

```
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)
```

```
t2 = wrf_user_getvar(a,"T2",  
  (/0,10,2/))
```

```
t2 = wrf_user_getvar(a,"T2",  
  (/1,2,3,4,5/))
```



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)

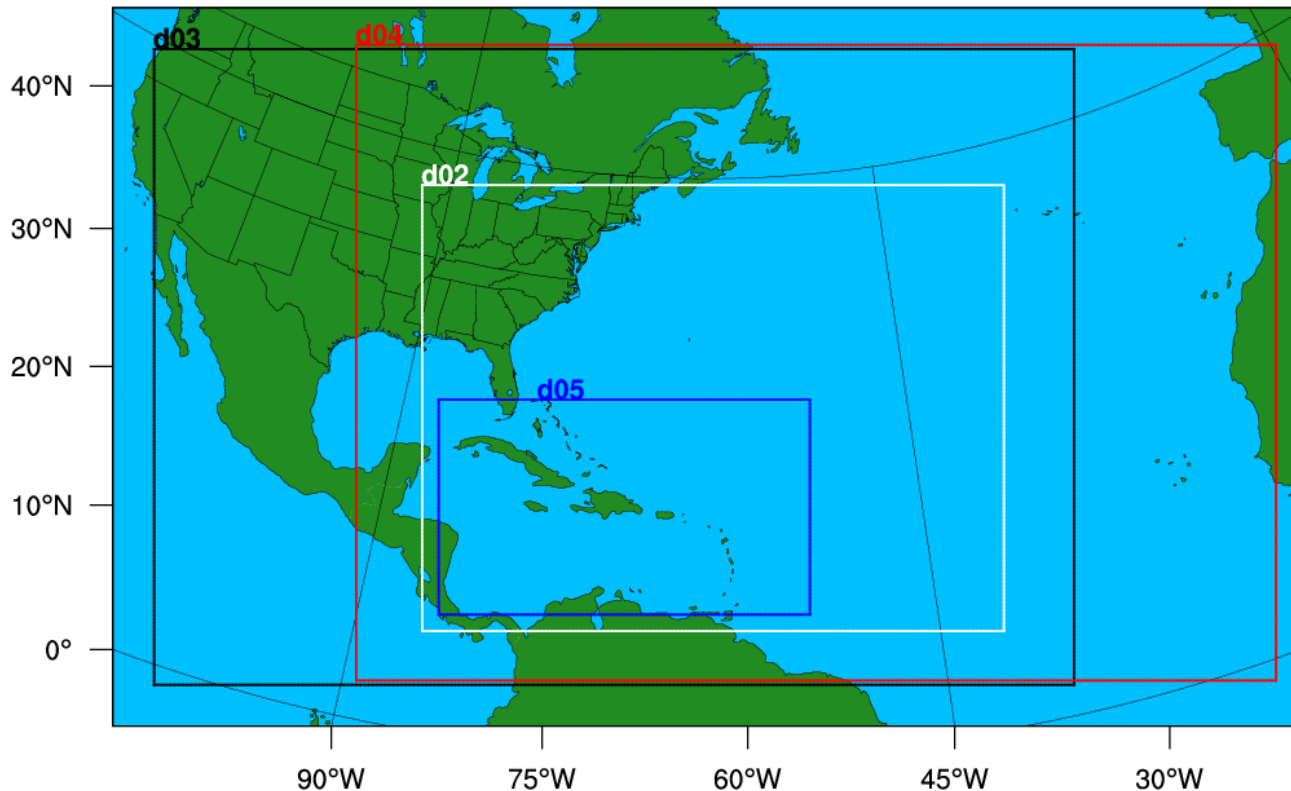


Domain Design

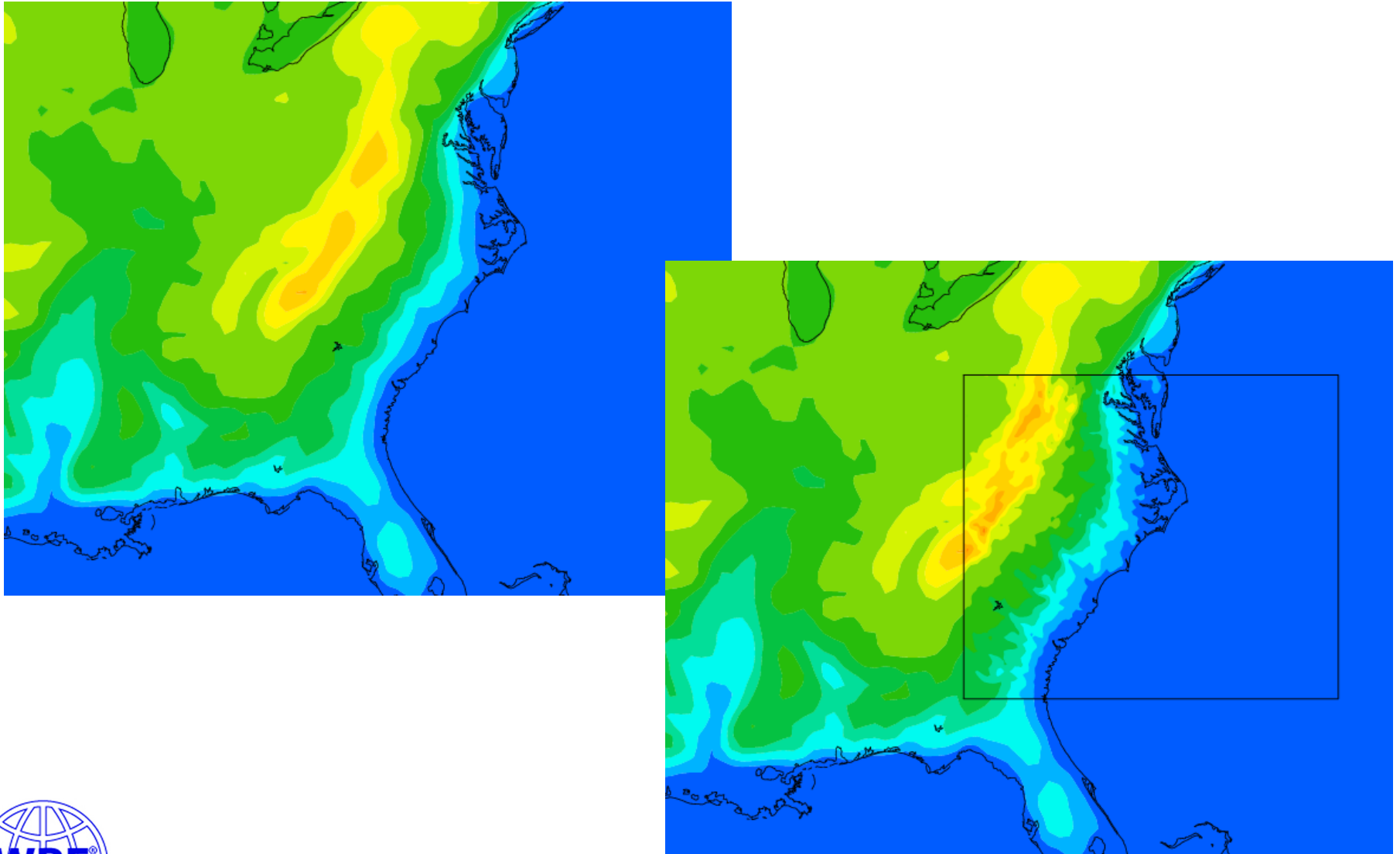
`mp = wrf_wps_dom (wks, mpres, lnres, txres)`

WPS/util/plotgrids.ncl

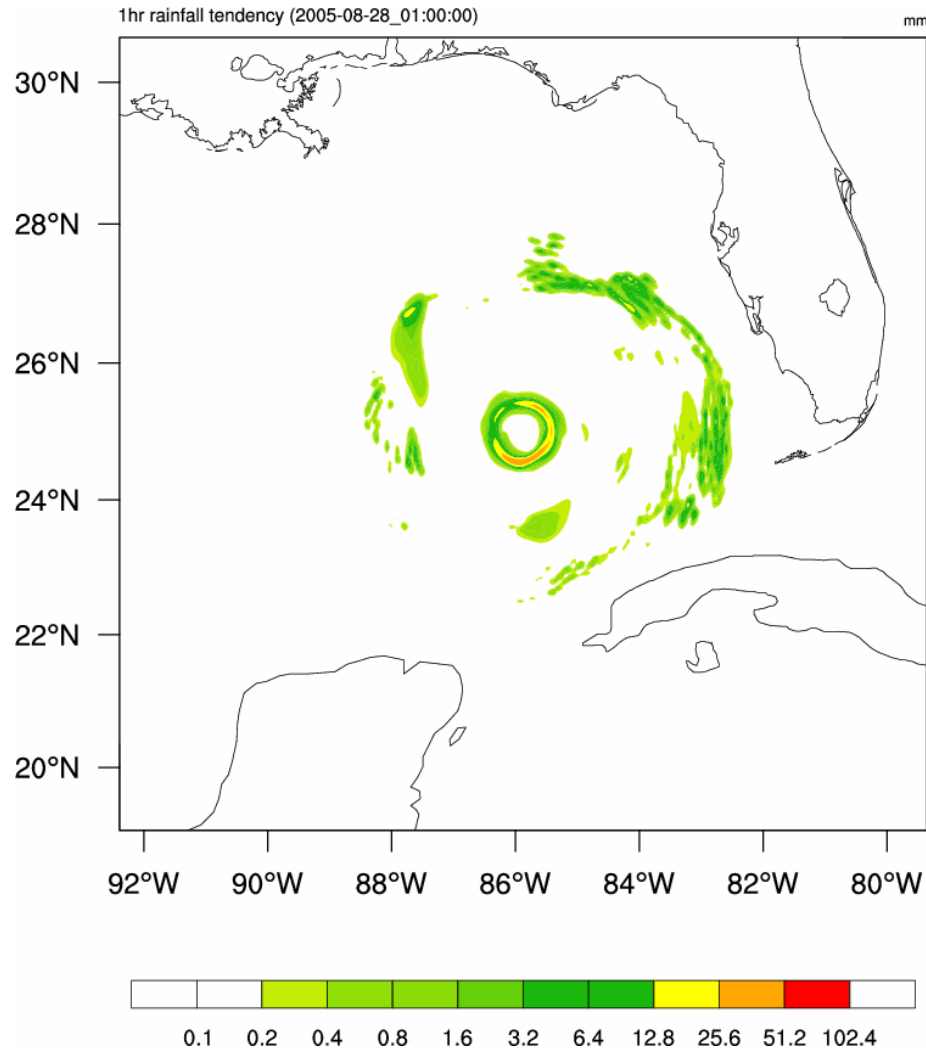
Test Domain



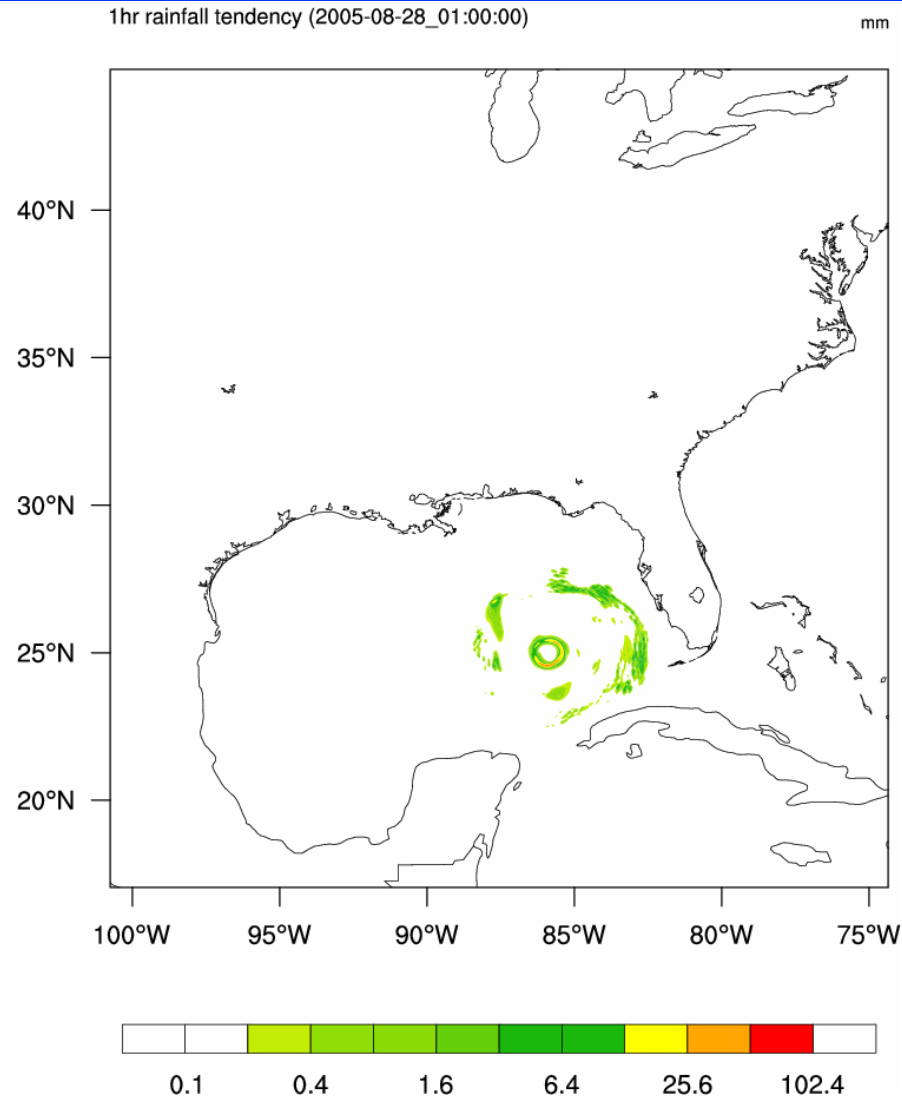
Over Domains



Moving Nests



Moving Nests



Change fields in netCDF file

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

```
begin
```

```
  a = addfile("./met_em.d01.2000-01-24_12:00:00.nc","w")
```

```
  sst = a->SST      ; read a field
```

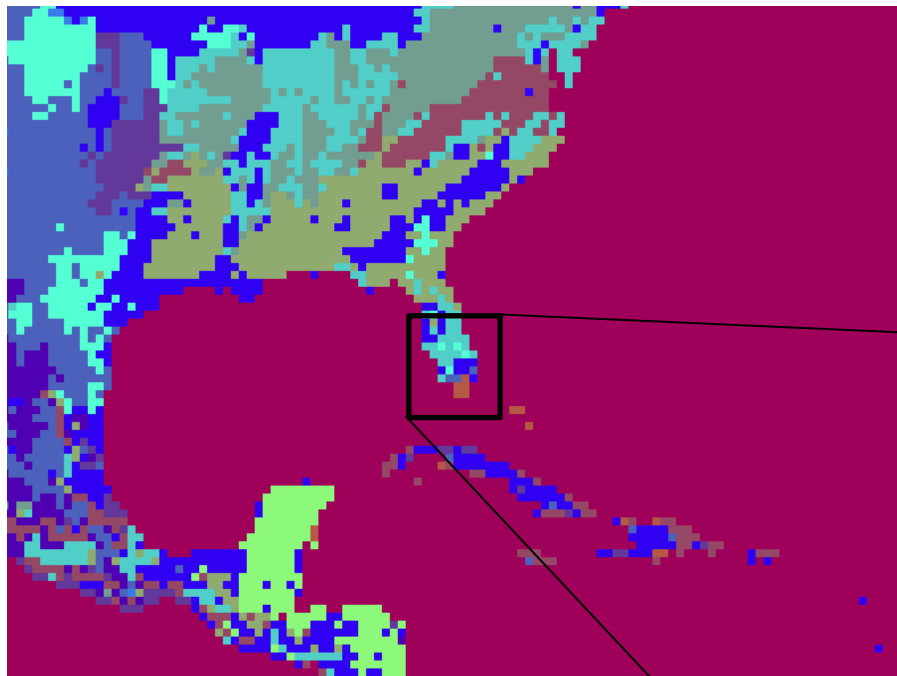
```
  sst = sst + 1     ; change the field
```

```
  a->SST = sst       ; write the field
```

```
end
```



Change fields in netCDF file

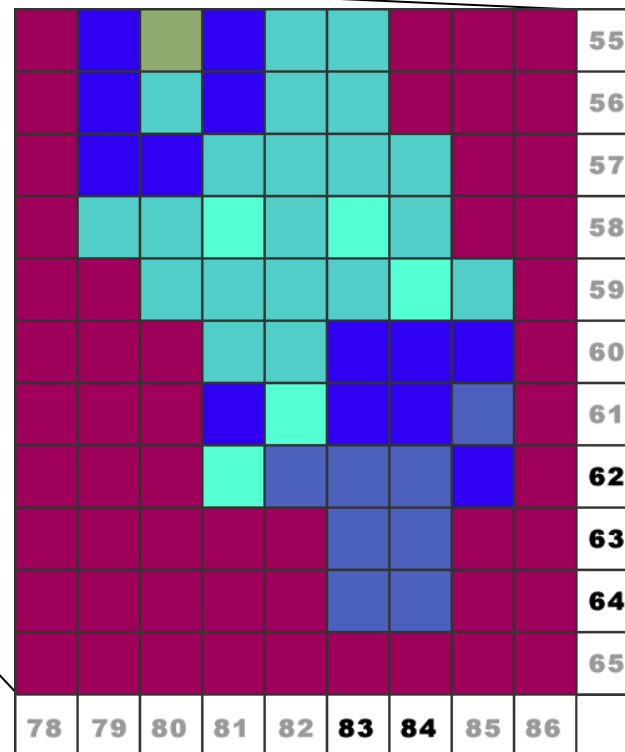


```
a = addfile("./geo_em.d01.nc","w")  
var= a->LANDUSE
```

```
var(:,63:64,83:84) = 7
```

```
var(:,62,84) = 7
```

```
a->LANDUSE = var
```



Reading ASCII Data

```
begin
```

```
fname = "/mydir/ascii.txt"
```

```
foo = asciiread(fname, (/21,6/), "integer")
```

```
end
```

```
Variable: foo  
Dimensions and sizes:  
[Data|21] x [Columns|6]
```

```
(0,0)    1950  
(0,1)    13  
(0,2)    11  
(0,3)    3  
(0,4)    8  
(0,5)    3  
(1,0)    1951  
(1,1)    10
```

/mydir/ascii.txt

1950	13	11	3	8	3
1951	10	8	3	5	2
1952	7	6	3	3	1
1953	14	6	2	4	1
1954	11	8	6	2	1
1955	12	9	3	6	2
1956	8	4	2	2	1
1957	8	3	1	2	2
1958	10	7	2	5	3
1959	11	7	5	2	1
1960	7	4	2	2	2
1961	11	8	1	7	4
1962	5	3	2	1	0
1963	9	7	5	2	1
1964	12	6	0	6	4
1965	5	4	3	1	1
1966	11	7	4	3	1
1967	8	6	5	1	1
1968	8	5	5	0	0
1969	18	12	7	5	1
1970	10	5	3	2	0



Writing ASCII Data

begin

```
fname = "/mydir/ascii.txt"
```

```
foo = asciiread(fname, (/21,6/), "integer")
```

```
fmtx      = "i4,3x,i3"
```

```
opt       = True
```

```
opt@fout = "aaa.txt"
```

```
write_matrix (foo(:,0:1), fmtx, opt)
```

end

1950	13
1951	10
1952	7
1953	14
1954	11
1955	12
1956	8
1957	8
1958	10
1959	11
1960	7

/mydir/ascii.txt

1950	13	11	3	8	3
1951	10	8	3	5	2
1952	7	6	3	3	1
1953	14	6	2	4	1
1954	11	8	6	2	1
1955	12	9	3	6	2
1956	8	4	2	2	1
1957	8	3	1	2	2
1958	10	7	2	5	3
1959	11	7	5	2	1
1960	7	4	2	2	2
1961	11	8	1	7	4
1962	5	3	2	1	0
1963	9	7	5	2	1
1964	12	6	0	6	4
1965	5	4	3	1	1
1966	11	7	4	3	1
1967	8	6	5	1	1
1968	8	5	5	0	0
1969	18	12	7	5	1
1970	10	5	3	2	0



Shapefiles

```
wrf_filename = "wrfout_d01_2000-07-17_00:00:00.nc"
a = addfile(wrf_filename,"r")
var          = a->OLR(0,::)
var@lat2d    = a->XLAT(0,::)
var@lon2d    = a->XLONG(0,::)

shp_filename = "climate_regions_gen.shp"

opt          = True
opt@shape_var = "REGION"
opt@shape_names = ("/"West","South"/)
var_mask     = shapefile_mask_data(var,shp_filename,opt)

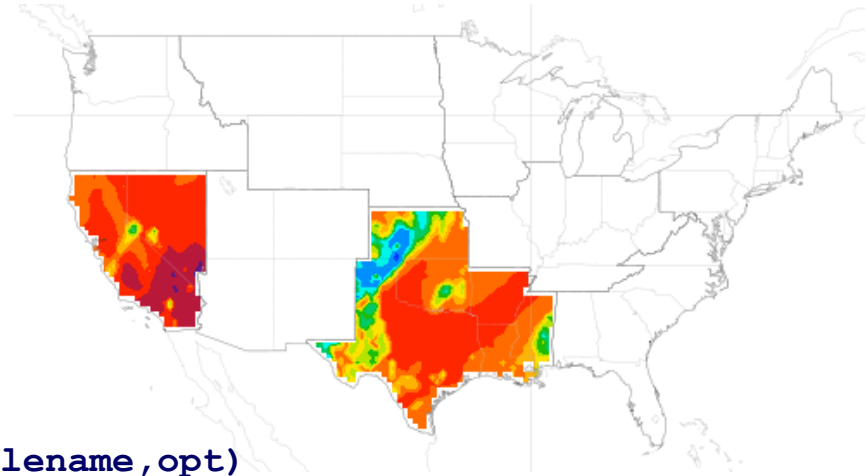
contour_mask = wrf_contour(a,wks,var_mask,True)

pltres       = True
pltres@PanelPlot = True
mpres        = True

plot_mask = wrf_map_overlays(a,wks,contour_mask,pltres,mpres)

id_mask = gsn_add_shapefile_polylines(wks,plot_mask,shp_filename,True)

draw(plot_mask)
frame(wks)
```



Geo Referenced Graphics



```

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

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

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

  mpres = True
  pltres = True

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

```

```

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

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

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

  mpres = True
  pltres = True
  pltres@gsnFrame = False

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

```

NCL and Fortran/C Code

- Link Fortran/C Code to NCL scripts
 - Create a library from your Fortran/C code
 - Link to NCL script
- Link Low Level NCL (NCAR Graphics) to Fortran Code
 - Add calls to code inside Fortran code
 - Compile Fortran Code with NCL libraries
 - Example: WPS/utils/plotfmt.exe
 - *Older way or creating plots – not recommended*



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.



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
```

