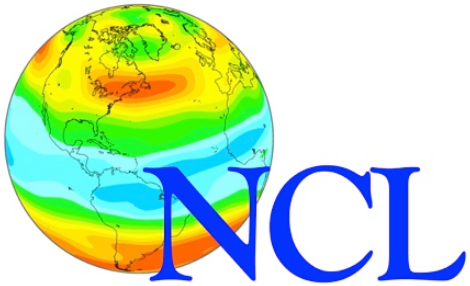
Visualizing WRF-ARW and MPAS Data Using NCL



JOINT WRF AND MPAS USERS' WORKSHOP

Mary Haley • CISL / TDD / VAST

June 15, 2018

The National Center for Atmospheric Research is sponsored by the National Science Foundation

Goals for this 90-minute tutorial

- Help you understand WRF-ARW and MPAS data
- Explain basic NCL scripts for plotting WRF-ARW and MPAS data
- Demonstrate ways to customize and create nice graphics
- Provide tips along the way

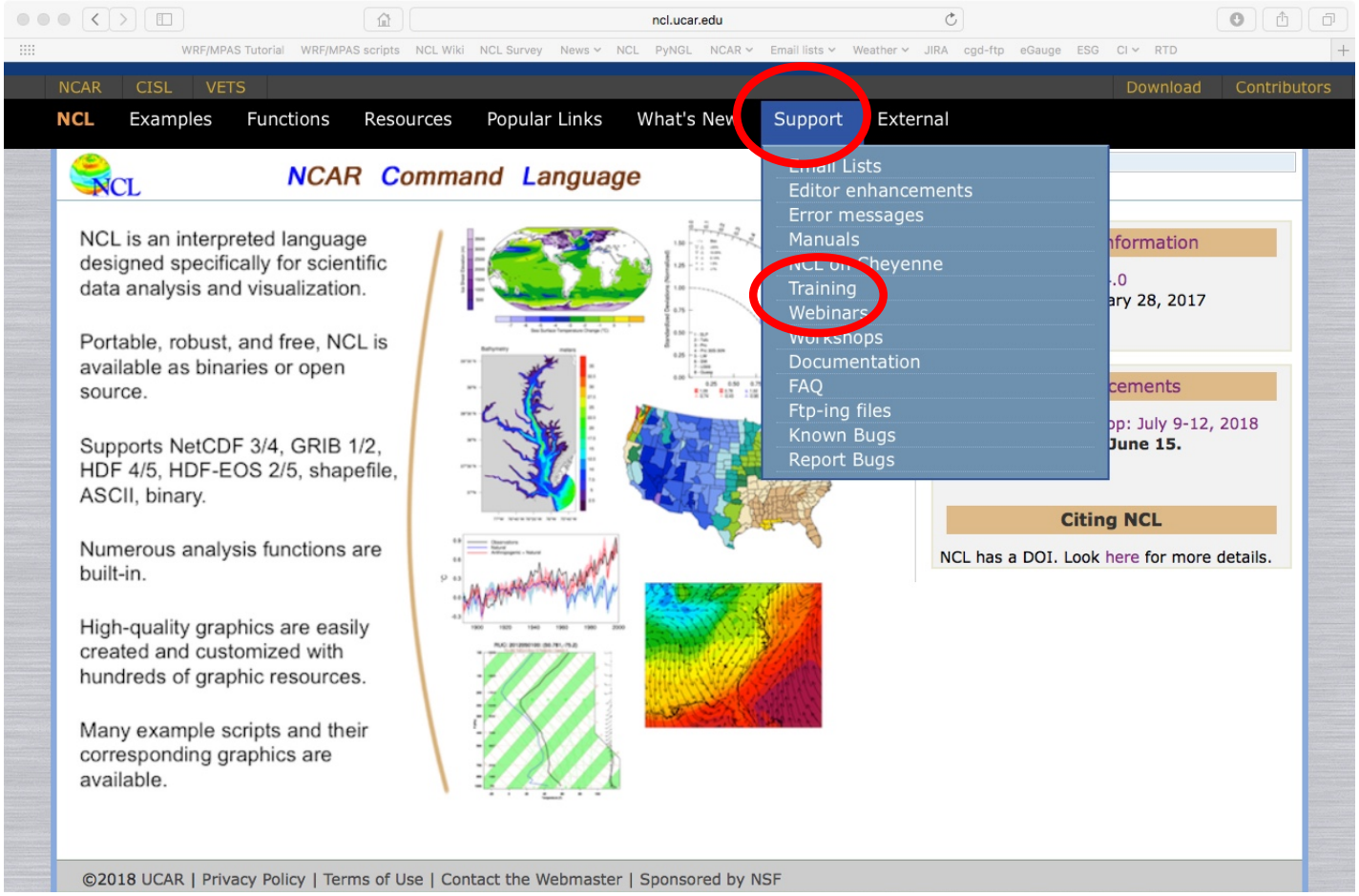
Bookmark this page

http://www.ncl.ucar.edu/Training/Tutorials/WRF_MPAS_Users_Workshop/

or go to:

http://www.ncl.ucar.edu

Support -> Training -> Tutorials



Outline

- Looking at your data
- Plotting WRF-ARW data
- Contouring MPAS data
- Tips for debugging, customizing graphics

Look at your data!

- Use "ncl_filedump" (UNIX command line)
- Use "ncdump –h" (UNIX command line)
- Write an NCL script

ncl_filedump / ncdump -h

`ncl_filedump` comes with NCL
`ncdump` is part of the NetCDF package, has to be installed separately

From a UNIX terminal window, type:

```
ncl_filedump wrfout_d01_2000-01-24_12:00:00.nc
```

or

```
ncdump -h wrfout_d01_2000-01-24_12:00:00
```

```
filename:       wrfout_d01_2000-01-24_12:00:00
path:   wrfout_d01_2000-01-24_12:00:00
   file global attributes:
      TITLE :  OUTPUT FROM WRF V3.1 MODEL
      START_DATE : 2000-01-24_12:00:00
      SIMULATION_START_DATE : 2000-01-24_12:00:00
      WEST-EAST_GRID_DIMENSION : 74
. . .
   dimensions:
      Time = 9  // unlimited
      DateStrLen = 19
      west_east = 73
      south_north = 60
      bottom_top = 27
      bottom_top_stag = 28
      soil_layers_stag = 4
      west_east_stag = 74
      force_layers = 8
      south_north_stag = 61
   variables:
. . .
      float U ( Time, bottom_top, south_north, west_east_stag )
         FieldType :    104
         MemoryOrder :  XYZ
         description :  x-wind component
         units :        m s-1
         stagger :      X
         coordinates :  XLONG U XLAT U
```

ncl_filedump output of WRF output file

```
filename:       init
file global attributes:
     model_name : mpas
     core_name : init_atmosphere
     source : MPAS
     Conventions : MPAS
  dimensions:
     nVertLevels = 55
     nCells = 2621442
     Time = 1  // unlimited
     StrLen = 64
     nEdges = 7864320
     nVertices = 5242880
     maxEdges = 10
     . . .
  variables:
     float qv ( Time, nCells, nVertLevels )
     float qc ( Time, nCells, nVertLevels )
     character xtime ( Time, StrLen )
     float latCell ( nCells )
     float lonCell ( nCells )
     float latVertex ( nVertices )
     float lonVertex ( nVertices )
```

ncl_filedump output of MPAS init.nc file

Querying a file using an NCL script

query_file_dims.ncl

```
;--- Open WRF output file
dir      = "wrfout_sample/"
filename = "wrfout_d01_2000-01-24_12:00:00"
f = addfile(dir + filename,"r")

dnames = getvardims(f)           ; Get all dimension names
dsizes = getfiledimsizes(f)      ; Get corresponding dimension sizes
print(dnames + " : " + dsizes)   ; Print the information
```

To run the script, type the following from a UNIX terminal window:

```
ncl query_file_dims.ncl
```

Output

```
 Copyright (C) 1995-2017 - All Rights Reserved
 University Corporation for Atmospheric Research
 NCAR Command Language Version 6.4.0
 The use of this software is governed by a License Agreement.
 See http://www.ncl.ucar.edu/ for more details.

(0) Time : 9
(1) DateStrLen : 19
(2) west_east : 73
(3) south_north : 60
(4) bottom_top : 27
(5) bottom_top_stag : 28
(6) soil_layers_stag : 4
(7) west_east_stag : 74
(8) force_layers : 8
(9) south_north_stag : 61
```

Tip : "-Q" and "-n" options

Use "-n" option to remove the Copyright header and the "(1)" text:

```
ncl -Q -n query_file_dims.ncl
```

Output:

```
Time : 9
DateStrLen : 19
west_east : 73
south_north : 60
bottom_top : 27
bottom_top_stag : 28
soil_layers_stag : 4
west_east_stag : 74
force_layers : 8
south_north_stag : 61
```

Tip: Use command line arguments

query_file_dims_cla.ncl

```
f = addfile(fname,"r")

dnames = getvardims(f)           ; Get all dimension names
dsizes = getfiledimsizes(f)      ; Get corresponding dimension sizes
print(dnames + " : " + dsizes)   ; Print the information
```

```
ncl 'fname="wrfout_d01_2000-01-24_12:00:00"' query_file_dims_cla.ncl
```

More scripting examples can be found at:

http://www.ncl.ucar.edu/Applications/system.shtml

Querying a file's variables using an NCL script

query_file_vars.ncl

```
;--- Open WRF output file
dir      = "wrfout_sample/"
filename = "wrfout_d01_2000-01-24_12:00:00"
f = addfile(dir + filename,"r")

vnames = getfilevarnames(f)     ; Get all variable names on the file
print(str_join(vnames,","))     ; Print on one line with commas
```

Times,LU_INDEX,ZNU,ZNW,ZS,DZS,U,Z_FORCE,U_G,U_G_TEND,V,V_G,V_G_TEND,W,W_SUBS,W_SUBS_TEND,PH,PHB,T,TH_UPSTREAM_X,TH_UPSTREAM_X_TEND,TH_UPSTREAM_Y,TH_UPSTREAM_Y_TEND,QV_UPSTREAM_X,QV_UPSTREAM_X_TEND,QV_UPSTREAM_Y,QV_UPSTREAM_Y_TEND,U_UPSTREAM_X,U_UPSTREAM_X_TEND,U_UPSTREAM_Y,U_UPSTREAM_Y_TEND,V_UPSTREAM_X,V_UPSTREAM_X_TEND,V_UPSTREAM_Y,V_UPSTREAM_Y_TEND,MU,MUB,NEST_POS,P,PB,SR,POTEVP,SNOPCX,SOILTB,FNM,FNP,RDNW,RDN,DNW,DN,CFN,CFN1,Q2,T2,TH2,PSFC,U10,V10,RDX,RDY,RESM,ZETATOP,CF1,CF2,CF3,ITIMESTEP,XTIME,QVAPOR,QCLOUD,QRAIN,LANDMASK,TSLB,SMOIS,SH2O,SEAICE,XICEM,SFROFF,UDROFF,IVGTYP,ISLTYP,VEGFRA,GRDFLX,ACGRDFLX,SNOW,SNOWH,RHOSN,CANWAT,SST,SSTSK,LAI,Z0,VAR,CON,OA1,OA2,OA3,OA4,OL1,OL2,OL3,OL4,MAPFAC_M,MAPFAC_U,MAPFAC_V,MAPFAC_MX,MAPFAC_MY,MAPFAC_UX,MAPFAC_UY,MAPFAC_VX,MF_VX_INV,MAPFAC_VY,F,E,SINALPHA,COSALPHA,HGT,HGT_SHAD,TSK,P_TOP,T00,P00,TLP,TISO,MAX_MSTFX,MAX_MSTFY,RAINC,RAINNC,I_RAINC,I_RAINNC,SNOWNC,GRAUPELNC,EDT_OUT,SWDOWN,GLW,OLR,XLAT,XLONG,XLAT_U,XLONG_U,XLAT_V,XLONG_V,ALBEDO,ALBBCK,EMISS,NOAHRES,TMN,XLAND,UST,PBLH,HFX,QFX,LH,ACHFX,ACLHF,SNOWC,SAVE_TOPO_FROM_REAL

Querying a file's variables using an NCL script

query_file_vars_more_info.ncl

```
;--- Open WRF output file
dir      = "wrfout_sample/"
filename = "wrfout_d01_2000-01-24_12:00:00"
f = addfile(dir + filename,"r")

vnames = getfilevarnames(f)     ; Get all variable names on the file
nvars  = dimsizes(vnames)       ; Get number of variables

;---Loop across each variable and print some information about it
do nv=0,nvars-1
  vtype   = getfilevartypes(f,vnames(nv))
  dnames := getfilevardims(f,vnames(nv))
  vsizes := getfilevardimsizes(f,vnames(nv))
  vatts  := getfilevaratts(f,vnames(nv))

  print("---Variable name '" + vnames(nv) + "'")
  print("  Type             : " + vtype)
  print("  Dimension names  : " + str_join(dnames," x ") + \
        " (" + str_join(vsizes," x ") + ")")
  print("  Attributes       : " + str_join(vatts,","))
end do
```

Output

```
---Variable name 'Times'
  Type            : character
  Dimension names : Time x DateStrLen (9 x 19)
  Attributes      : missing
---Variable name 'LU_INDEX'
  Type            : float
  Dimension names : Time x south_north x west_east (9 x 60 x 73)
  Attributes      : FieldType,MemoryOrder,description,units,stagger,coordinates
---Variable name 'ZNU'
  Type            : float
  Dimension names : Time x bottom_top (9 x 27)
  Attributes      : FieldType,MemoryOrder,description,units,stagger
---Variable name 'ZNW'
  Type            : float
  Dimension names : Time x bottom_top_stag (9 x 28)
  Attributes      : FieldType,MemoryOrder,description,units,stagger
---Variable name 'ZS'
  Type            : float
  Dimension names : Time x soil_layers_stag (9 x 4)
  Attributes      : FieldType,MemoryOrder,description,units,stagger
---Variable name 'DZS'
  Type            : float
  Dimension names : Time x soil_layers_stag (9 x 4)
  Attributes      : FieldType,MemoryOrder,description,units,stagger
---Variable name 'U'
  Type            : float
  Dimension names : Time x bottom_top x south_north x west_east_stag (9 x 27 x 60 x 7
  Attributes      : FieldType,MemoryOrder,description,units,stagger,coordinates
```

Querying a variable

query_var.ncl

```
;---Open MPAS file and look at a variable
mpas_file = "../Data/mpas_sample/x1.40962.static.nc"
f         = addfile(mpas_file,"r")
ter       = f->ter
latc      = f->latCell


printVarSummary(ter)
printMinMax(ter,0)
print("dimensions of ter = " + dimsizes(ter))

printVarSummary(latc)
printMinMax(latc,0)
```

Output

```
Variable: ter
Type: double
Total Size: 327696 bytes
            40962 values
Number of Dimensions: 1
Dimensions and sizes: [nCells | 40962]
Coordinates:
Number Of Attributes: 0
(0) min=-47.96369025565024   max=5444.551303475936
(0) dimensions of ter = 40962


Variable: latc
Type: double
Total Size: 327696 bytes
            40962 values
Number of Dimensions: 1
Dimensions and sizes: [nCells | 40962]
Coordinates:
Number Of Attributes: 0
(0) min=-1.570796326794897   max=1.570796326794897
```

Partial list of query functions

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

- getfilevarnames
- getfilevardims
- getfilevardimsizes
- getfilevartypes
- getfilevaratts
- printVarSummary
- printMinMax
- dimsizes
- print

Outline

- Looking at your data
- Plotting WRF-ARW data
- Contouring MPAS data
- Tips for debugging, customizing graphics

Main WRF-NCL function: **wrf_user_getvar**

wrf_user_getvar - Get fields from input file and/or calculate diagnostics

```
a = addfile("wrfout_d01_2005-08-28_00:00:00","r")

cttmp = wrf_user_getvar(a,"ctt",0) ;  0 → first time step

 slp  = wrf_user_getvar(a,"slp",1) ;  1 → second time step

  tc = wrf_user_getvar(a,"tc",-1) ; -1 -> all time steps

 hgt = wrf_user_getvar(a,"ter",0) ; terrain, 1st time step
```

Main WRF-NCL function: **wrf_user_getvar**

avo	absolute vorticity [10-5 s-1]
eth	Equivalent Potential Temperature [K]
cape_2d	Returns 2D fields mcape/mcin/lcl/lfc
cape_3d	Returns 3D fields cape and cin
ctt	Cloud Top Temperature [degC]
dbz	Reflectivity [dBZ]
mdbz	Maximum reflectivity [dBZ]
geopt	Full model geopotential [m2 s-2]
helicity	Storm Relative Helicity [m-2/s-2]
omg	Omega [C]
p	Full model pressure [Pa]
pressure	Full model pressure [hPa]
pvo	potential vorticity [PVU]
pw	Precipitable Water
rh2	2m Relative Humidity [%]
rh	Relative Humidity [%]
slp	Sea level pressure [hPa]

ter	Model terrain height [m]
td2	2m dew point temperature [C]
td	Dew point temperature [C]
tc	Temperature [C]
theta	Potential temperature [K]
tk	Temperature [K]
tv	Virtual temperature [K]
twb	Wet bulb temperature [K]
updraft_helicity	Updraft helicity [m-2/s-2]
ua	U component of wind on mass points
va	V component of wind on mass points
wa	W component of wind on mass points
uvmet10	10m U and V components of wind rotated to earth coordinates
uvmet	U and V components of wind rotated to earth coordinates
z / height	Full model height [m]

Visualizing WRF with gsn_csm_xxx scripts

To plot data in NATIVE WRF map projection defined on file:

1. Call "`wrf_map_resources`" to set up map resources
2. Set `tfDoNDCOverlay` resource to True
3. Set `gsnAddCyclic` resource to False
4. Call one of the gsn_csm_xxx_map functions:
   - `gsn_csm_contour_map`
   - `gsn_csm_vector_map`
   - `gsn_csm_streamline_map`

gsn_csm script – native projection

```
a   = addfile("wrfout_d01_2000-01-24_12:00:00","r")
hgt = wrf_user_getvar(a,"HGT",0)
```

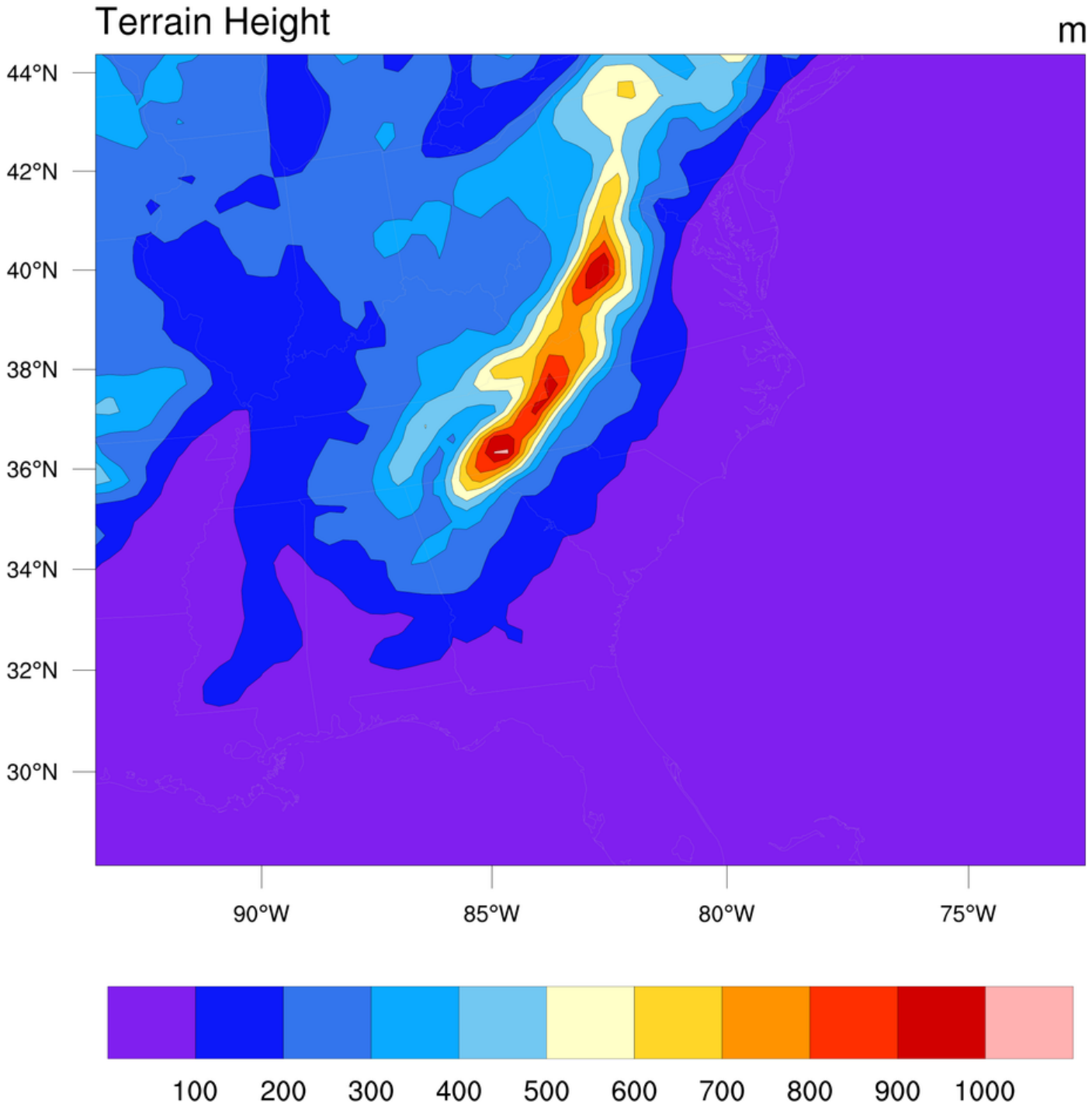
```
wks = gsn_open_wks("x11","wrf_demo")
```

```
;---Required to properly set up WRF map projection
 res = True
 res = wrf_map_resources(a,res)
 res@tfDoNDCOverlay = True
 res@gsnAddCyclic   = False
```

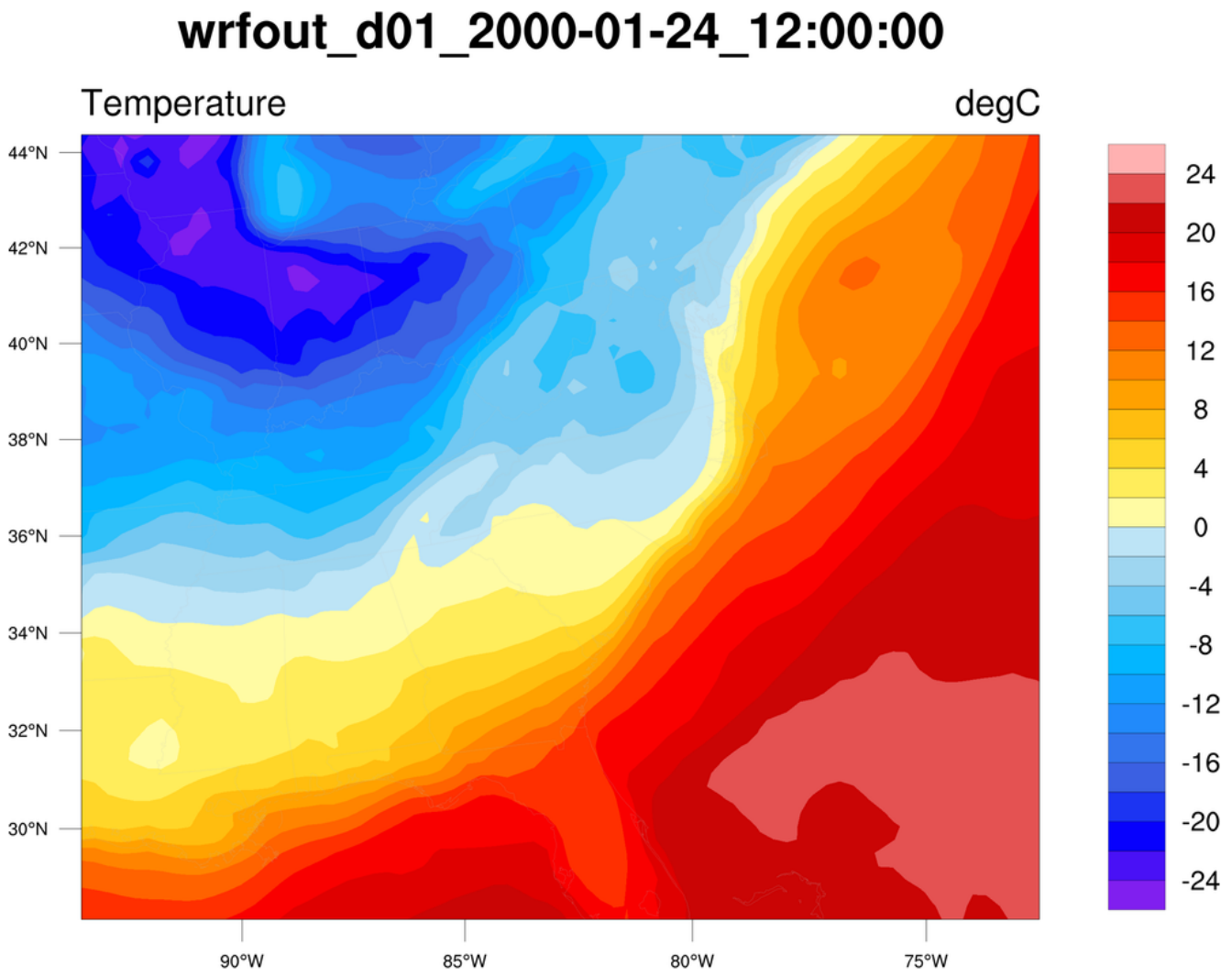
```
 res@cnFillOn       = True     ; Turn on color fill
 res@cnLinesOn      = False    ; Turn off contour lines
 res@tiMainString   = filename ; Set a main title
```

```
plot = gsn_csm_contour_map(wks,hgt,res)
```

wrfout_d01_2000-01-24_12:00:00



Demo #1 – plotting native WRF data



Visualizing WRF with gsn_csm_xxx scripts

To plot data in NON-NATIVE map projection:

1. Set special "lat2d" / "lon2d" attributes
2. Set options for the map projection you want
3. Set `gsnAddCyclic` resource to False
4. Call one of the gsn_csm_xxx_map functions:
   - `gsn_csm_contour_map`
   - `gsn_csm_vector_map`
   - `gsn_csm_streamline_map`

gsn_csm script – non-native projection

```
filename = "wrfout_d01_2000-01-24_12:00:00"
a    = addfile(filename,"r")
hgt = wrf_user_getvar(a,"hgt",0)
```

```
;---Required for using different map projection
hgt@lat2d = wrf_user_getvar(a,"XLAT",0)
hgt@lon2d = wrf_user_getvar(a,"XLONG",0)
```

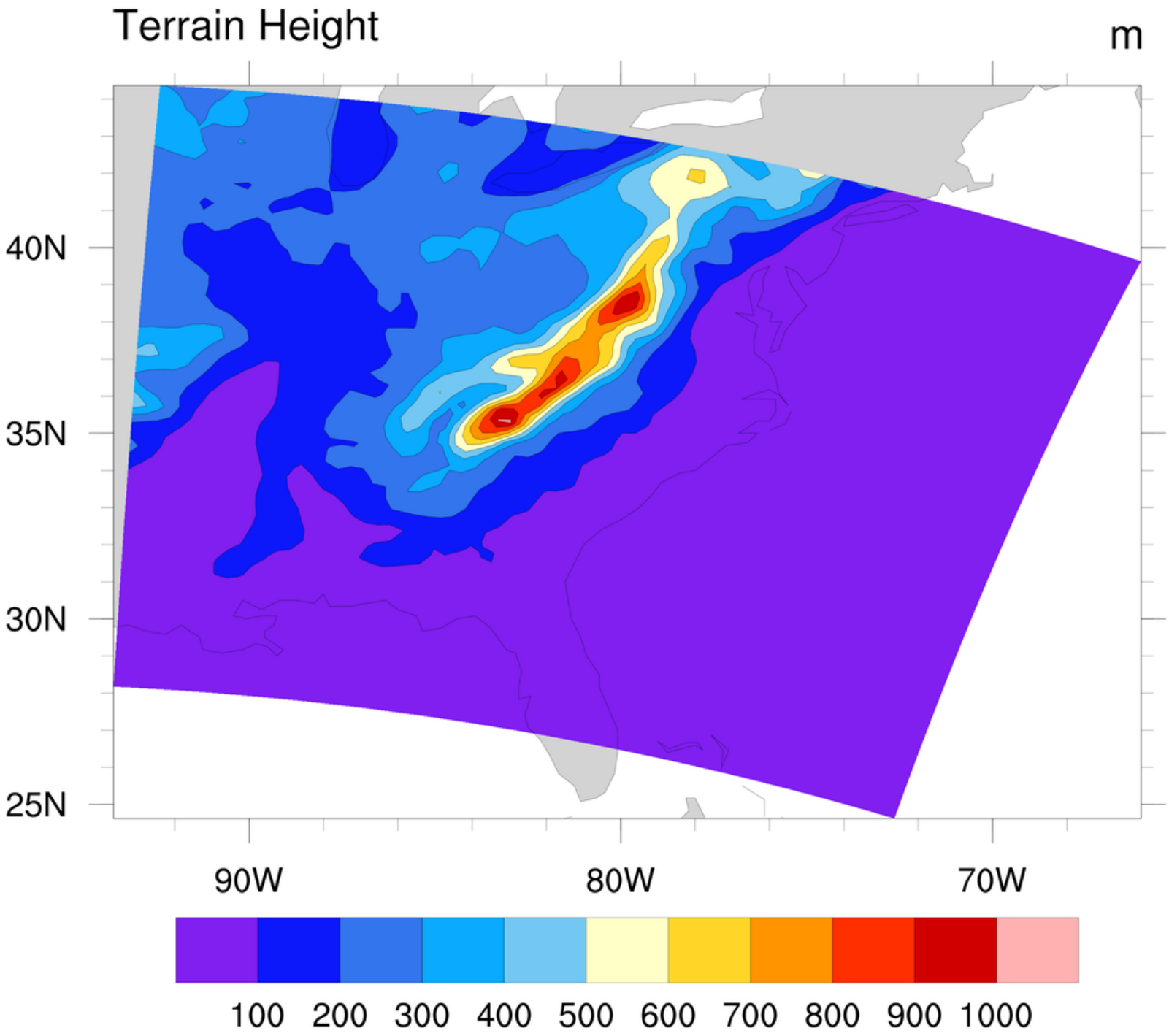
```
wks = gsn_open_wks("x11","wrf_demo")
```

```
res@mpMinLatF       = min(hgt@lat2d)  ; Select area of
res@mpMaxLatF       = max(hgt@lat2d)  ; map to view.
res@mpMinLonF       = min(hgt@lon2d)
res@mpMaxLonF       = max(hgt@lon2d)
res@gsnAddCyclic    = False
```

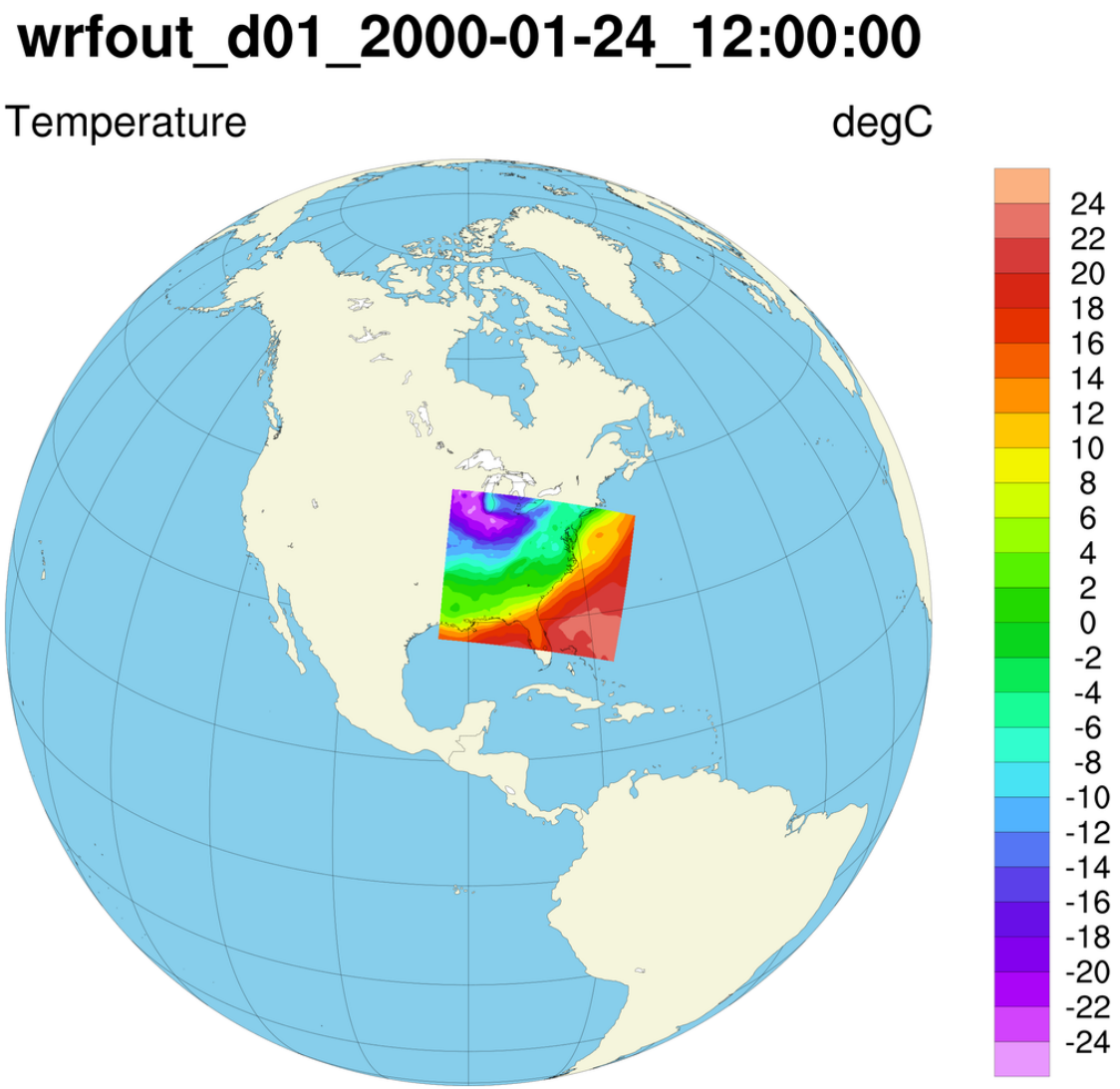
```
res@cnFillOn        = True      ; Turn on color fill
res@cnLinesOn       = False     ; Turn off contour lines
res@tiMainString    = filename ; set a main title
```

```
plot = gsn_csm_contour_map(wks,hgt,res)
```

wrfout_d01_2000-01-24_12:00:00



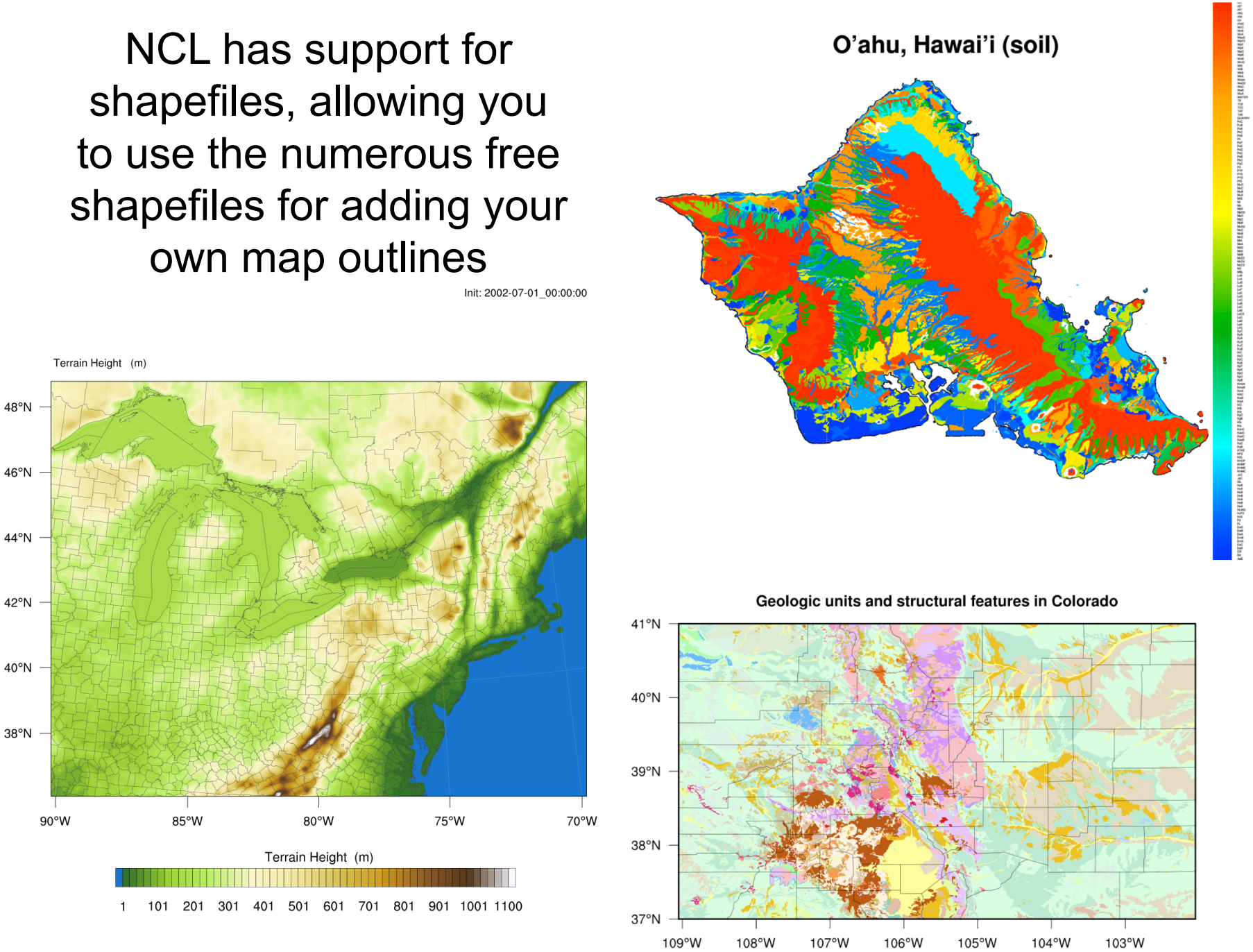
Demo #2 – Plotting non-native WRF



Shapefiles

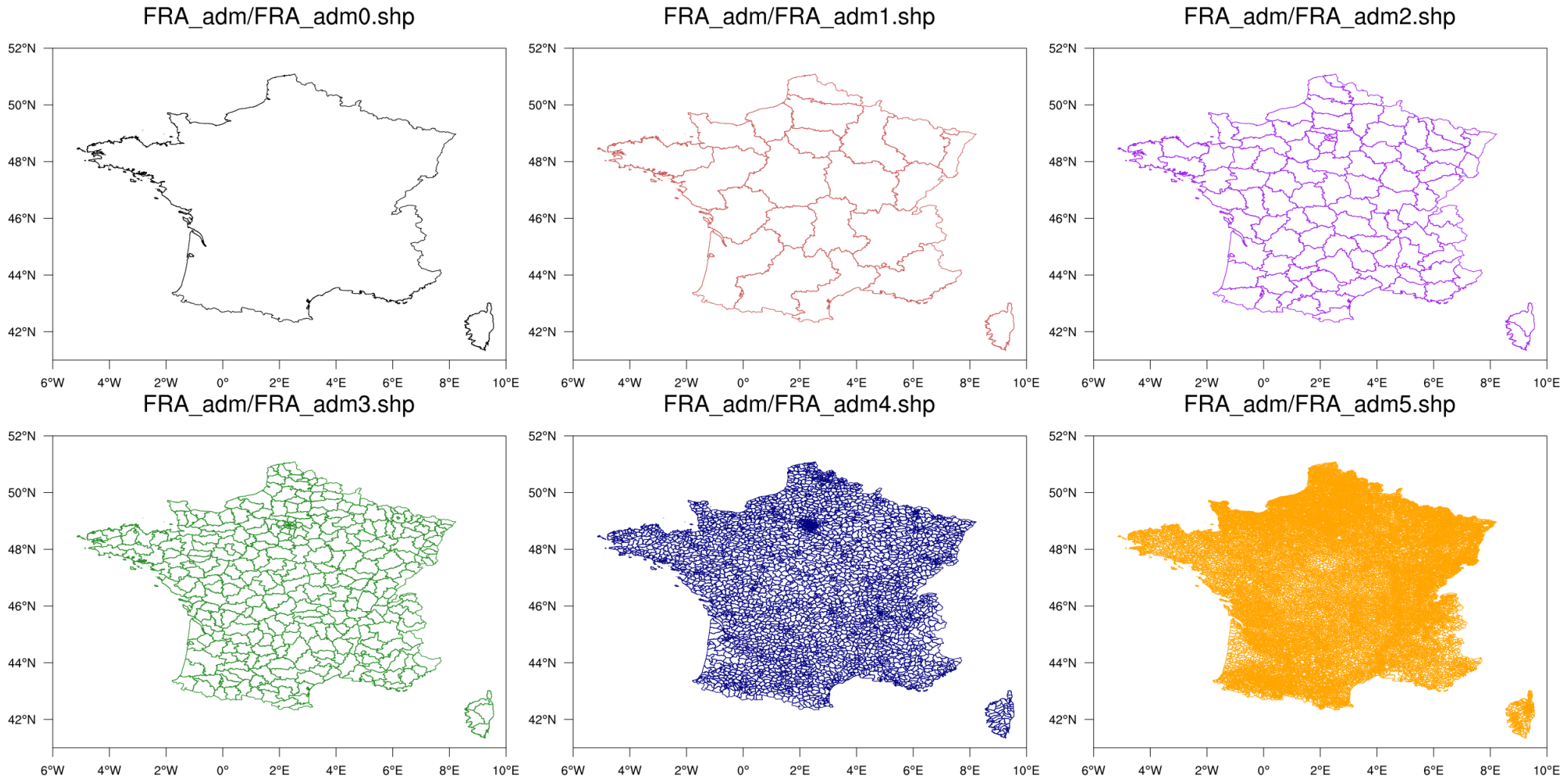
- Files that store geographic information
  - Country outlines, roads, lakes
  - Population data, election data
  - Lat/lon locations of sites of interest
- Use to add nicer map outlines
- Use to mask data
- Can find lots of free shapefiles on the web

NCL has support for shapefiles, allowing you to use the numerous free shapefiles for adding your own map outlines



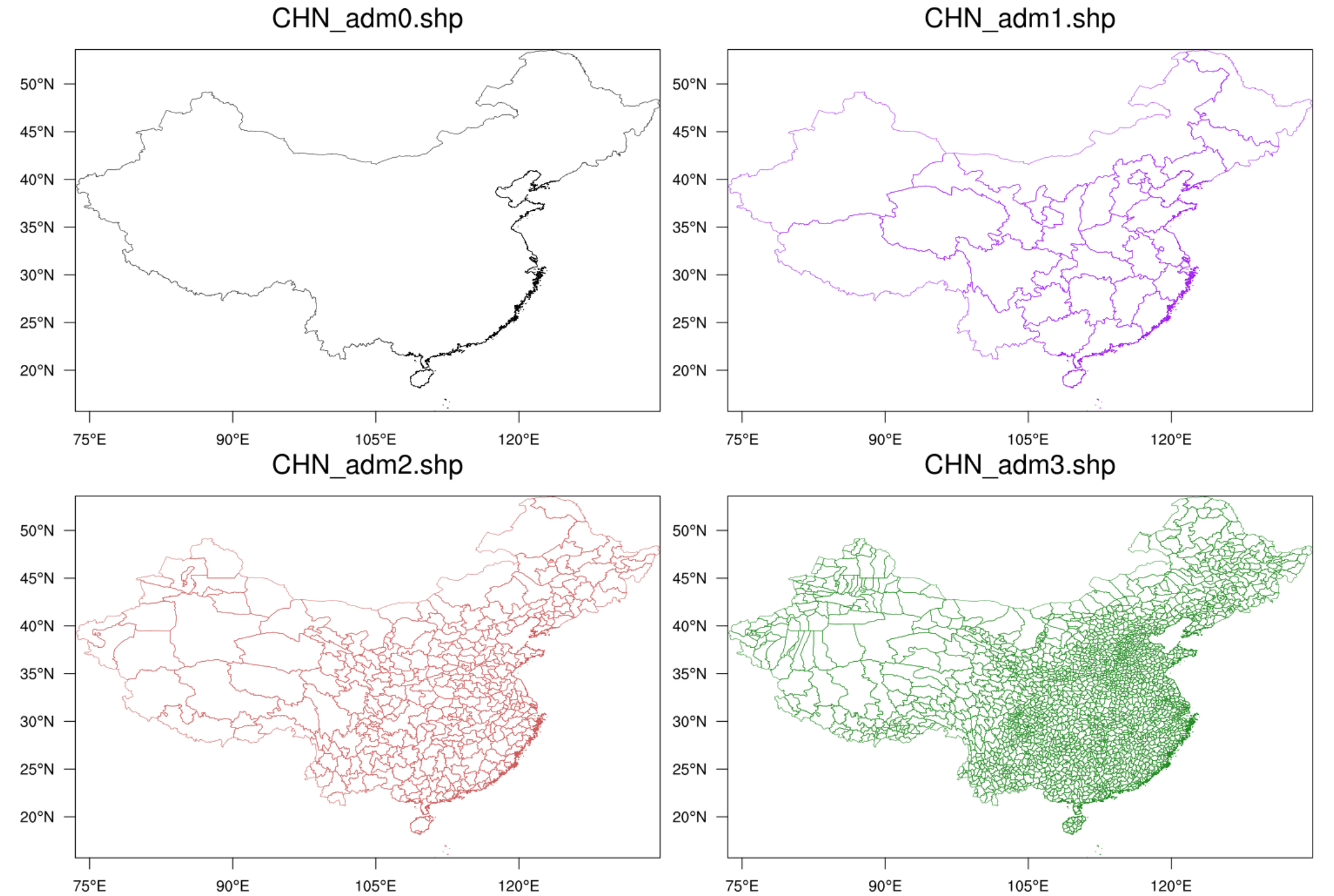
OUTPUT FROM WRF V3.3 MODEL
WE = 225 ; SN = 175 ; Levels = 28 ; Dis = 8km ; Phys Opt = 6 ; PBL Opt = 1 ; Cu Opt = 1

Shapefiles give you detailed geographical outlines



Global Administrative Areas database (**http://www.gadm.org**) offers consistent administrative boundaries at many levels. The level 0 database (nations) is good to use for global or mesoscale results, level 1 is the first level of sub-national administration (typically states/provinces and territories) while level 2 offers the second level of administration and is potentially useful for high-resolution plots.

China shapefiles from gadm.org/country

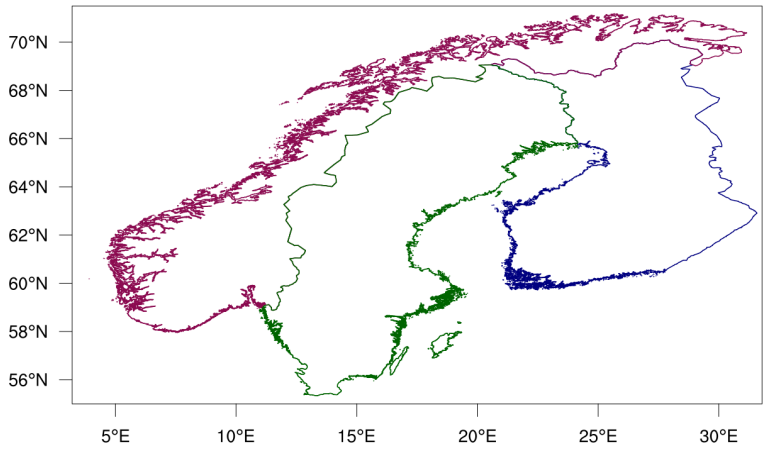


Boundaries from gadm.org shapefiles

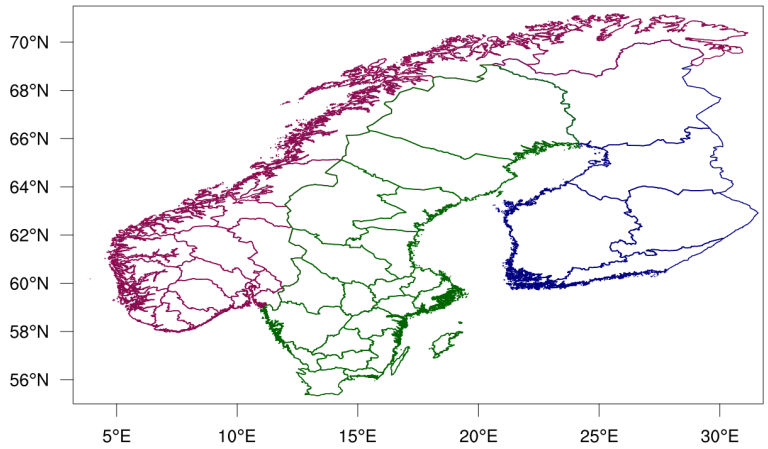
Norway, Sweden, and Finland shapefile bounaries from gadm.org

Finland has 2 more “adm” levels

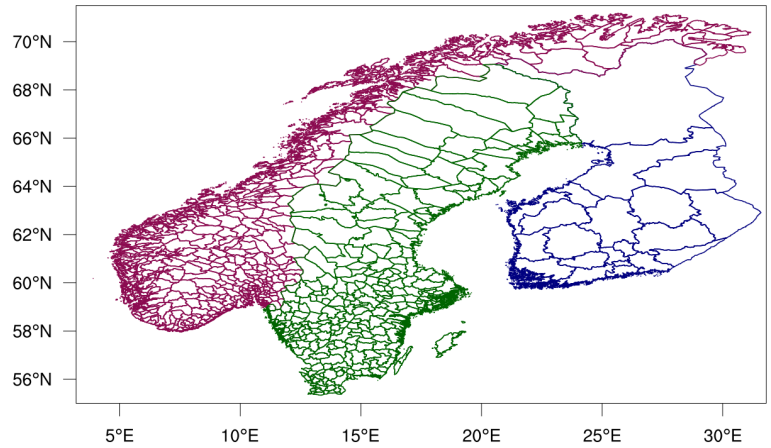
ADM0 shapefile outlines



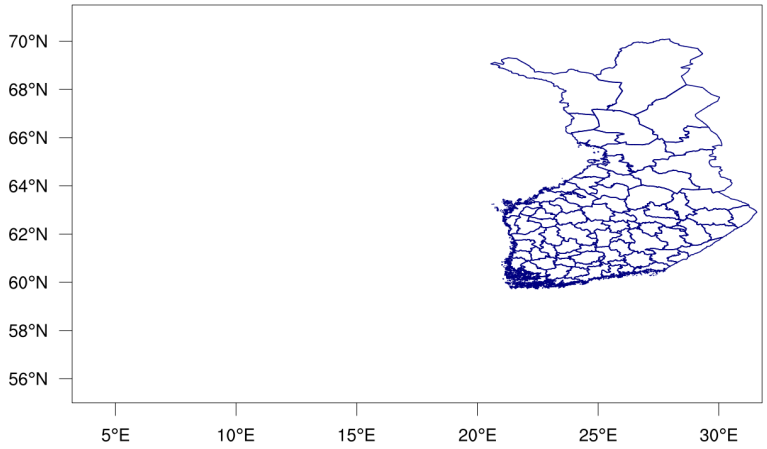
ADM1 shapefile outlines



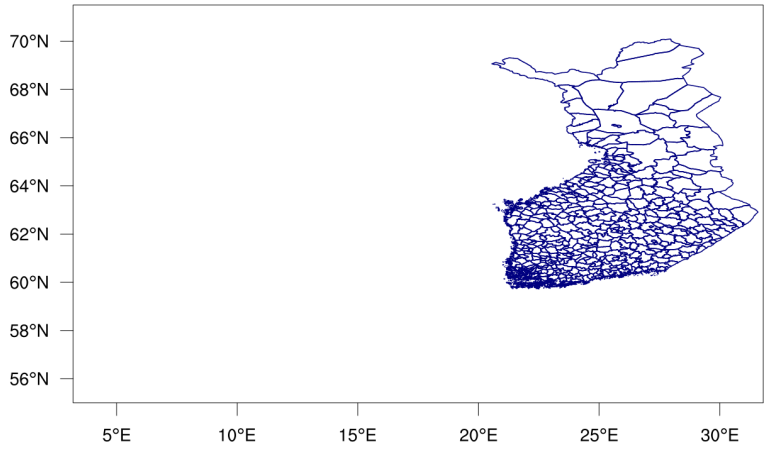
ADM2 shapefile outlines



ADM3 shapefile outlines

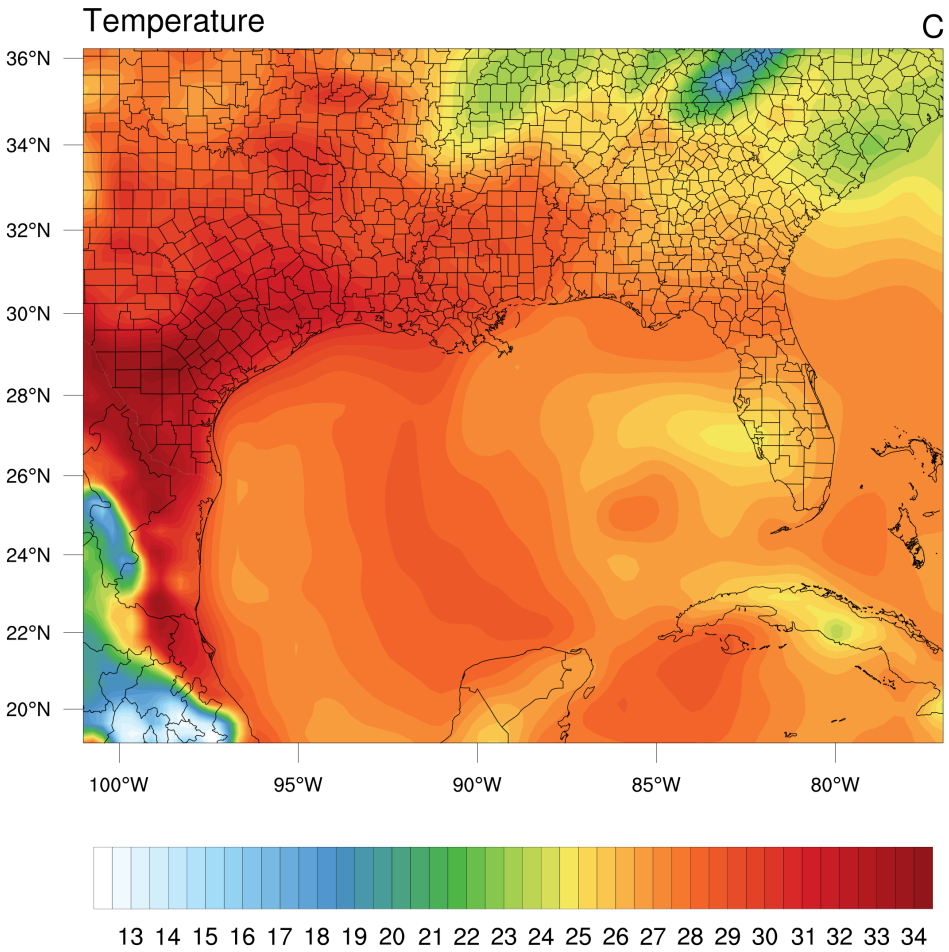


ADM4 shapefile outlines

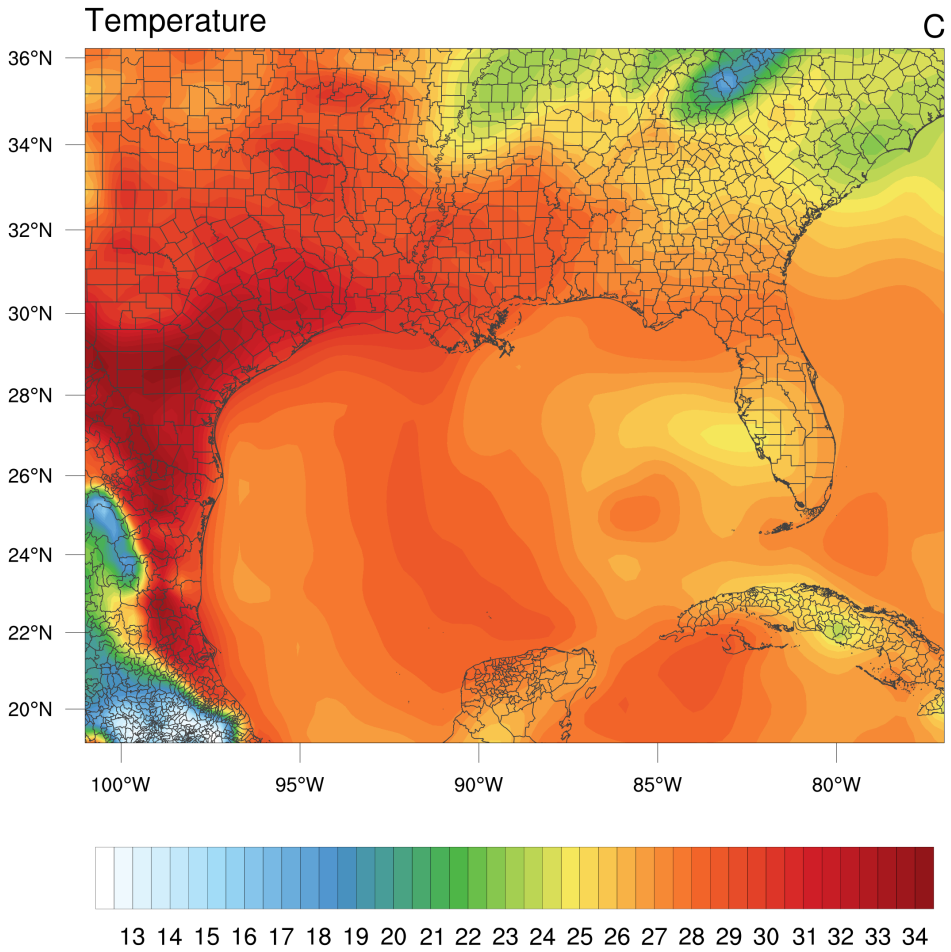


USA, Mexico, Cuba shapefile outlines added

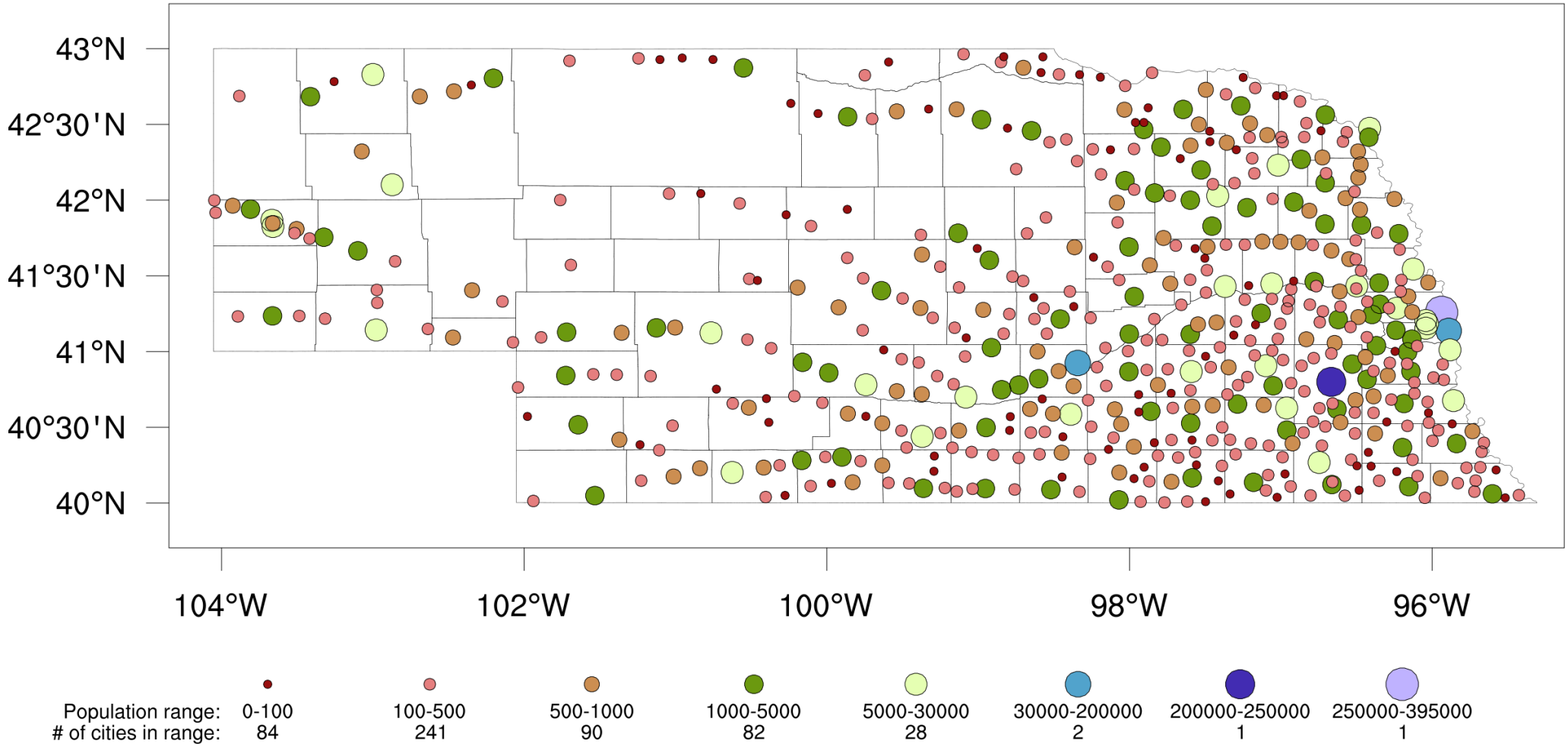
NCL map outlines



Map outlines via shapefiles

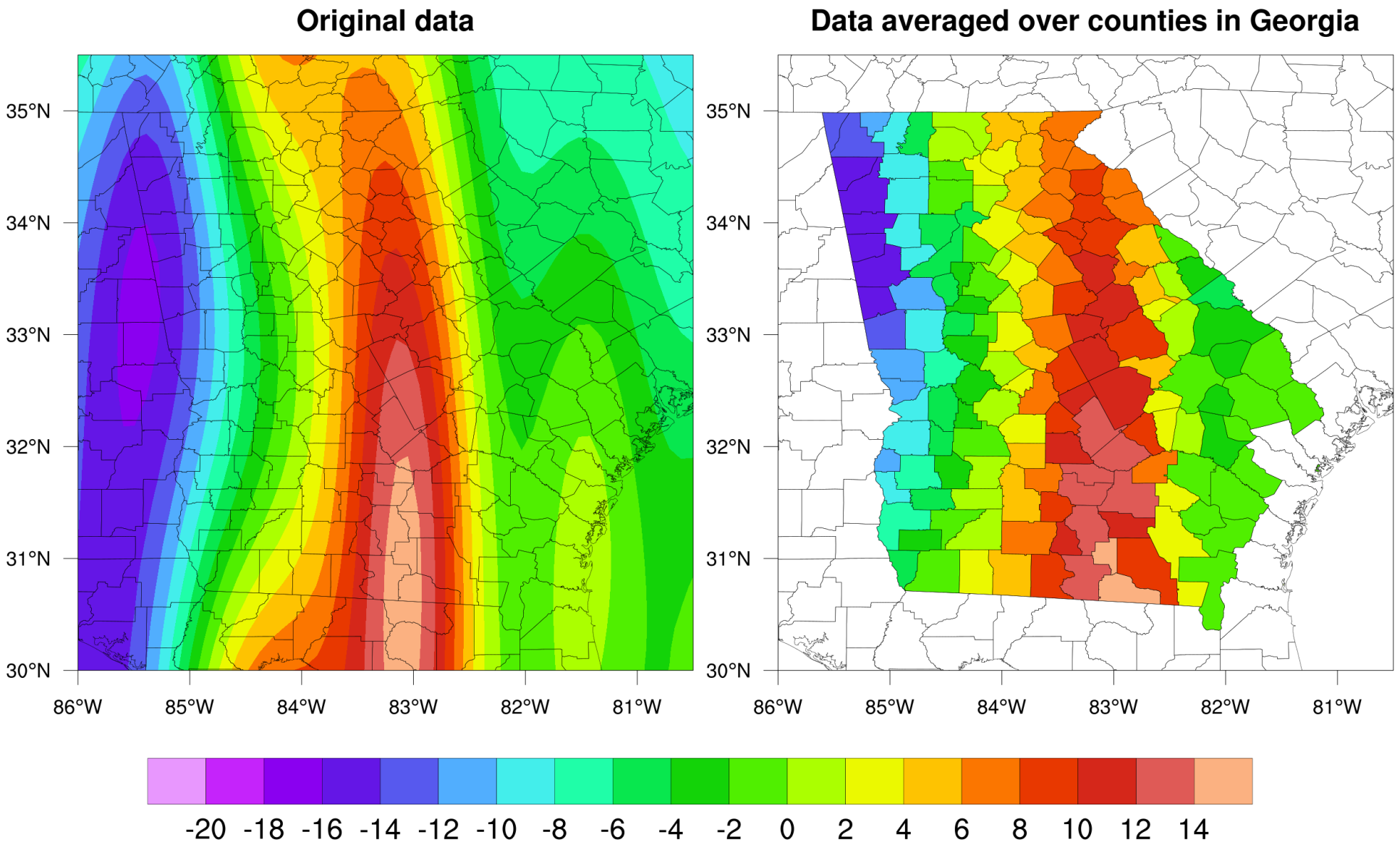


Population of Nebraska cities



Population data downloaded from Nebraska Department of Natural Resources
http://dnr.nebraska.gov/boundaries-plss

Use to mask and/or average data over a particular area



http://www.ncl.ucar.edu/Applications/shapefiles.shtml#ex13

Demo #3 - Shapefiles

How to download from gadm.org

Where to find more examples:

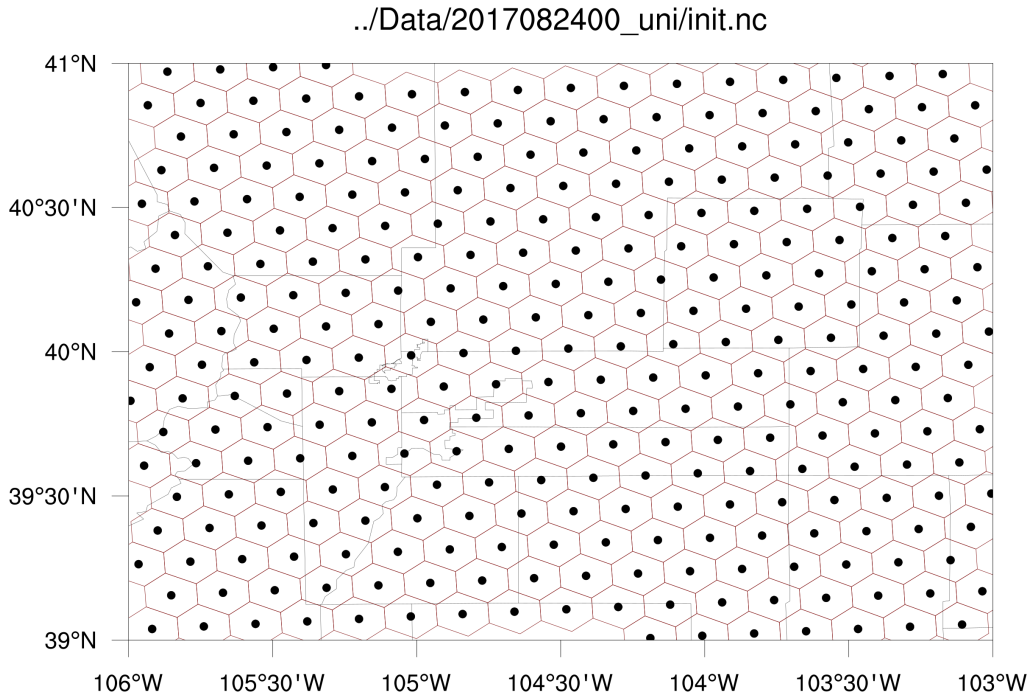
http://www.ncl.ucar.edu/Applications/shapefiles.shtml

Outline

- Looking at your data
- Plotting WRF-ARW data
- Contouring MPAS data
- Tips for debugging, customizing graphics

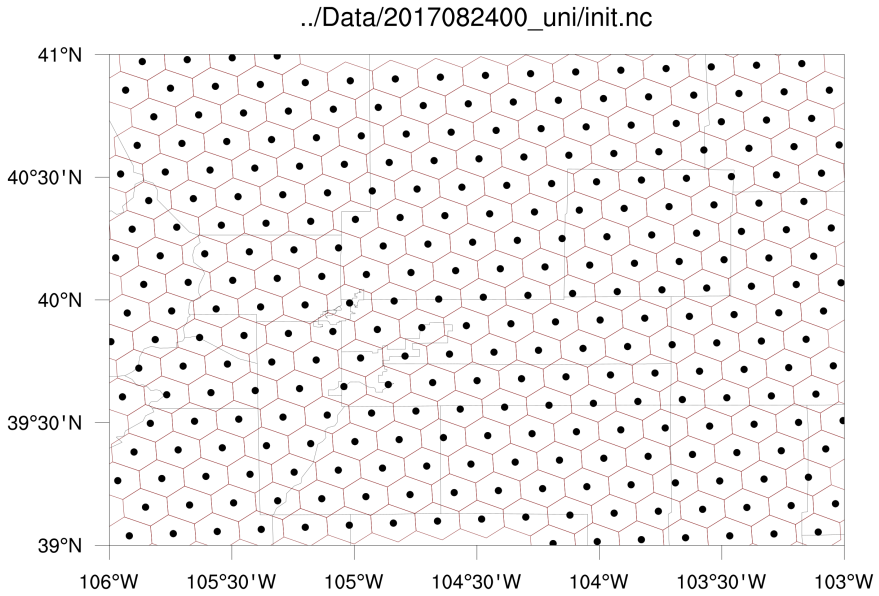
Plotting MPAS data – raster contours

- Use raster fill! Default "area fill" is memory and time intensive.
- The latitude / longitude arrays for each cell center are called latCell / lonCell. The vertexes are latVertex / lonVertex
- Need to convert these lat/lon arrays from radians to degrees.
- Need to set sfXArray / sfYArray to lonCell / latCell

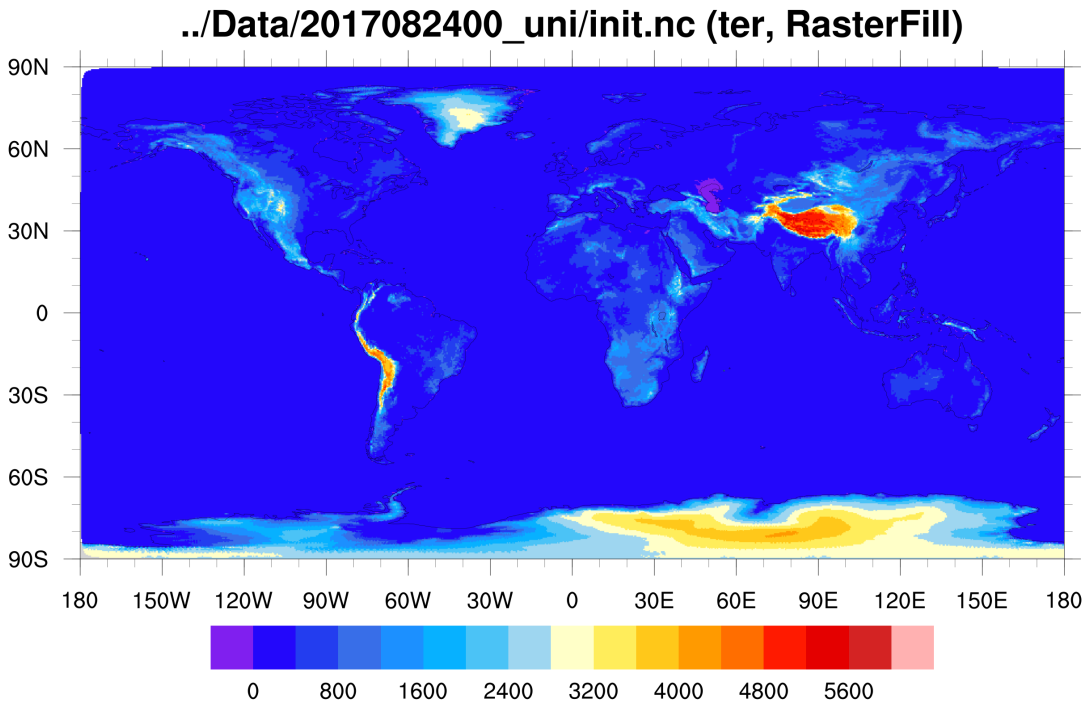


Plotting MPAS data – cell fill contours

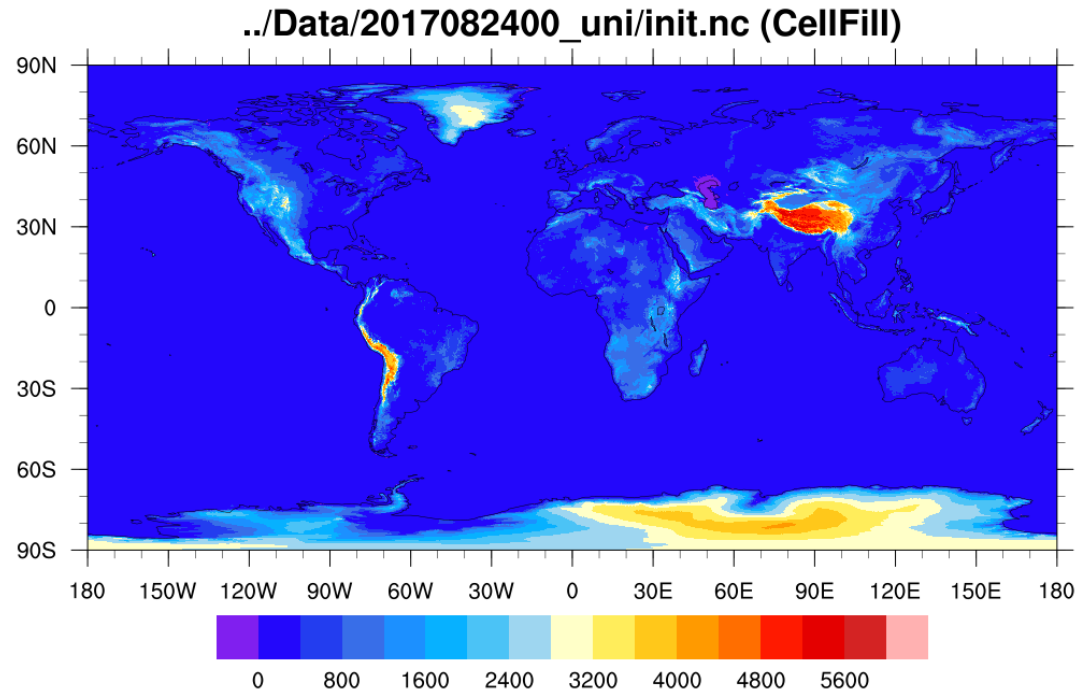
- Can also use "CellFill" for nicer looking plots
- Nicer looking because you are providing the polygon border for each cell point
- Slower because you must construct the polygon borders for each cell, which is time-consuming
- …Need to set sfXArray / sfYArray AND sfXCellBounds / sfYCellBounds



Comparing RasterFill and CellFill



7.9 seconds

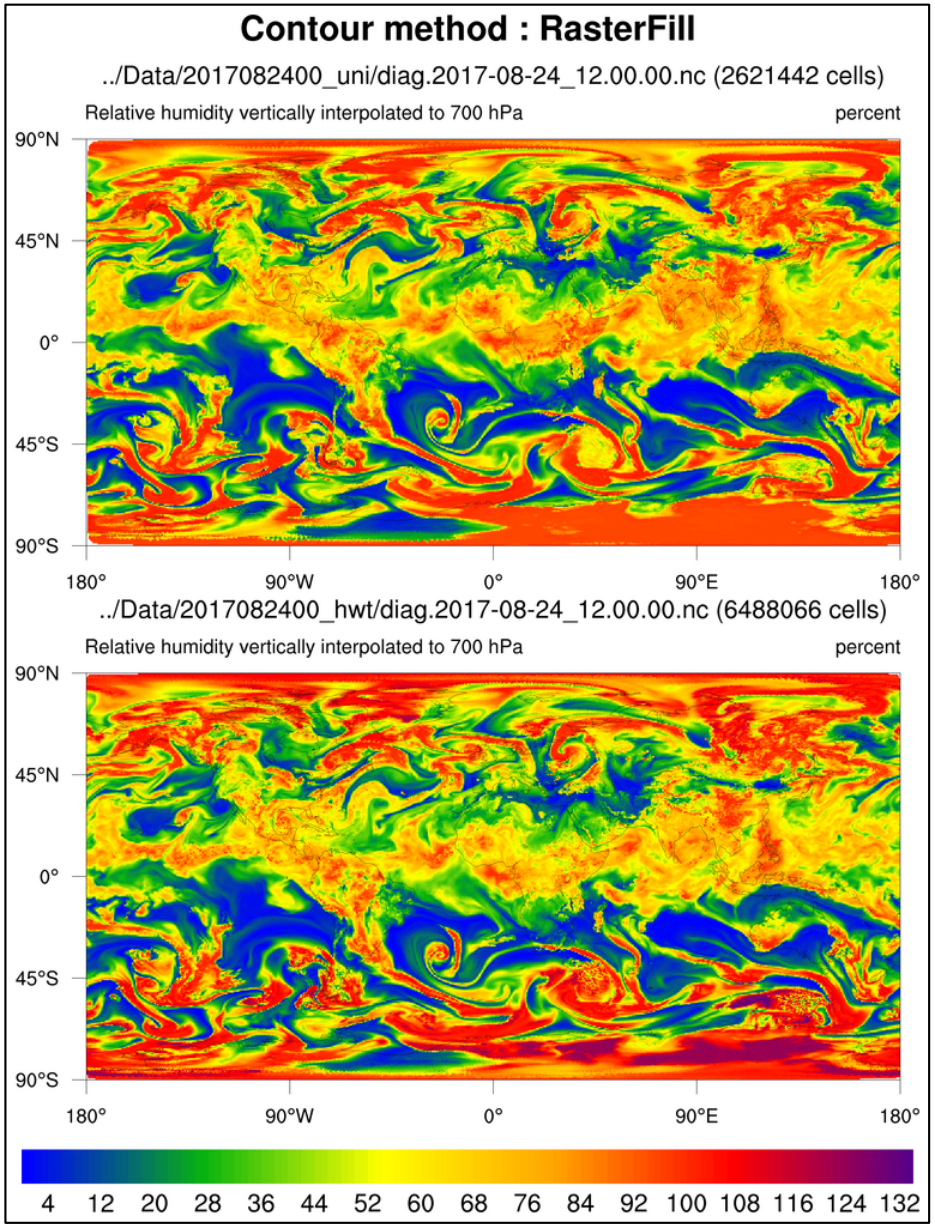


63 seconds

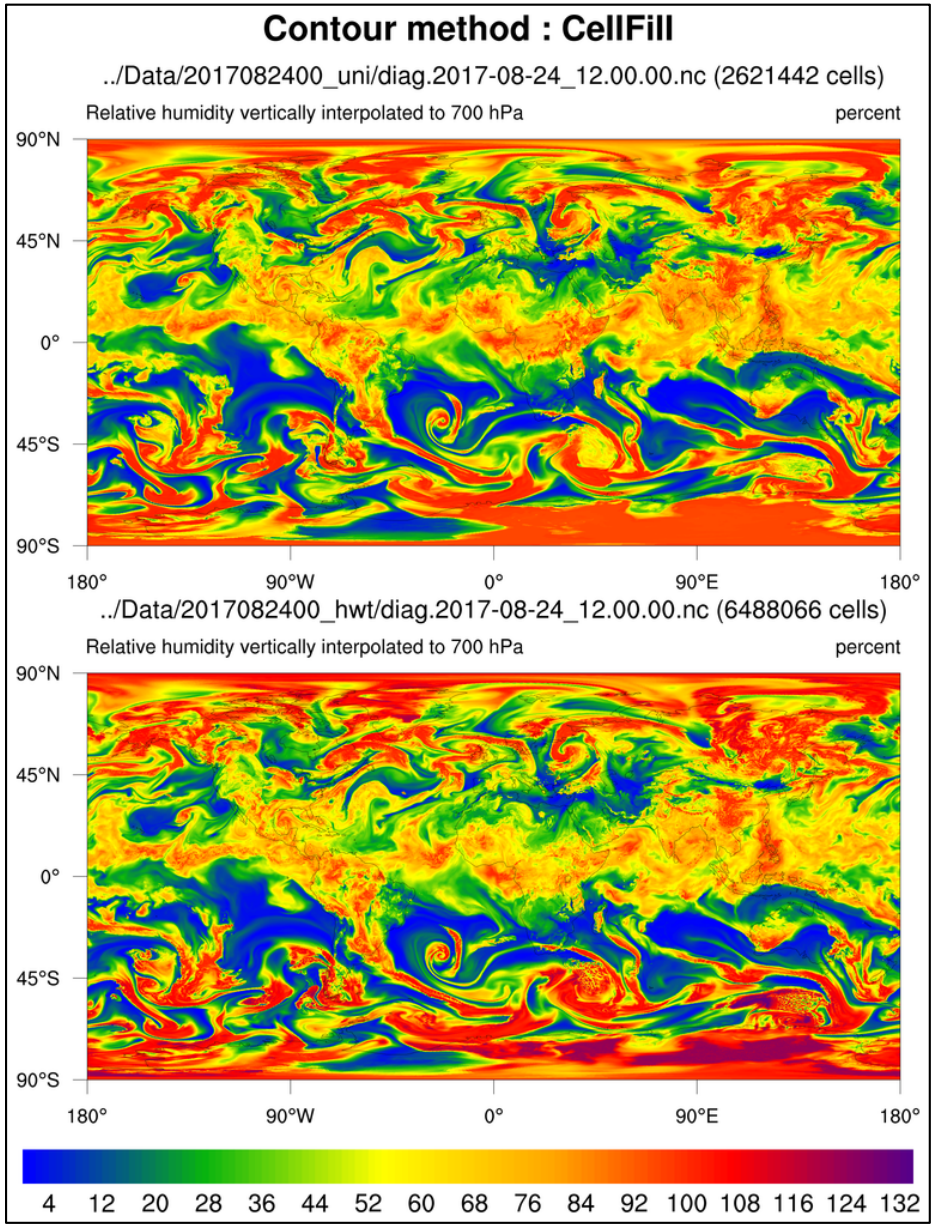
(~30 seconds of this is for constructing the cell edges)

Comparing global raster and cell filled plots

Constructing the cell boundaries is slow!

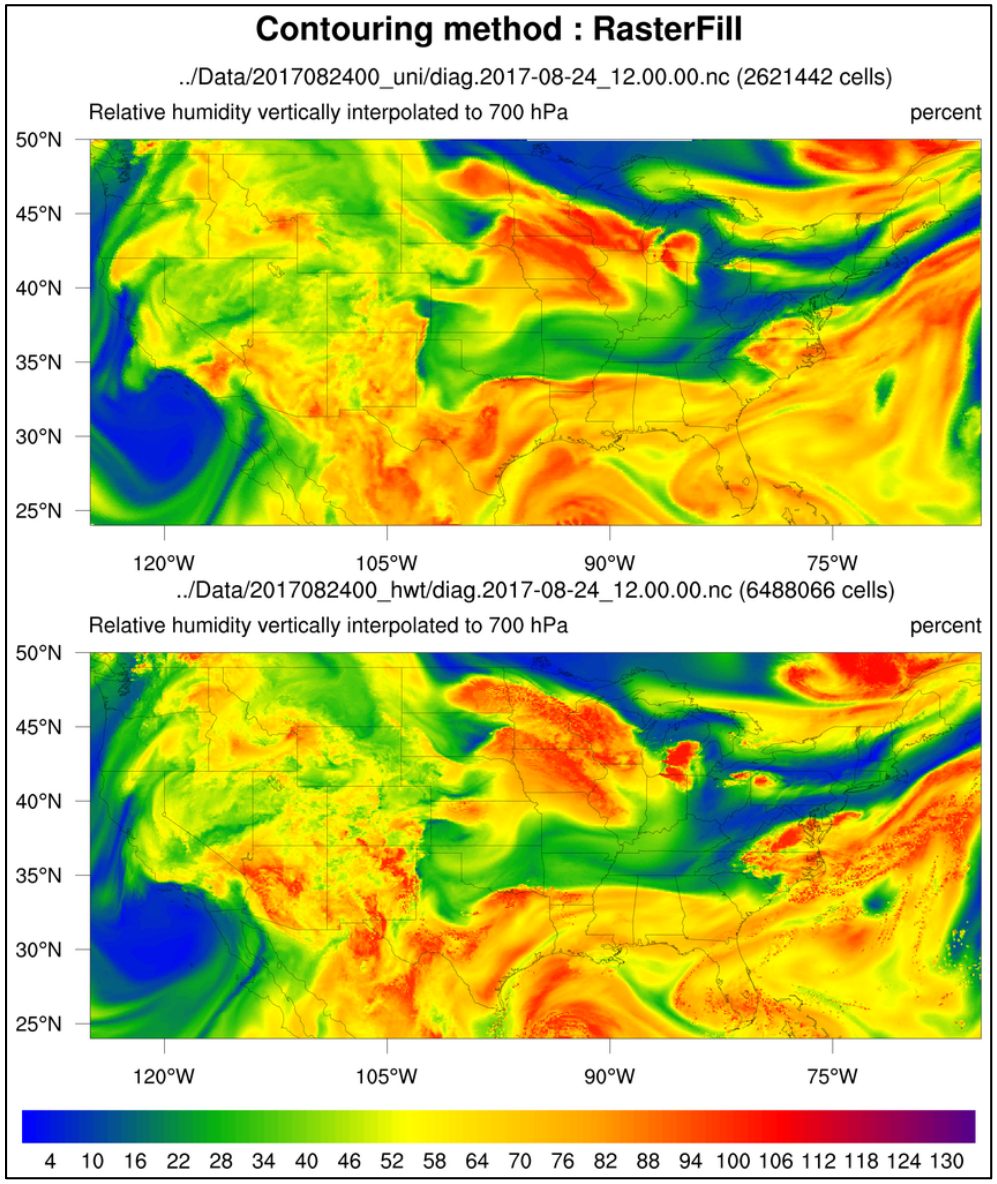


27 seconds

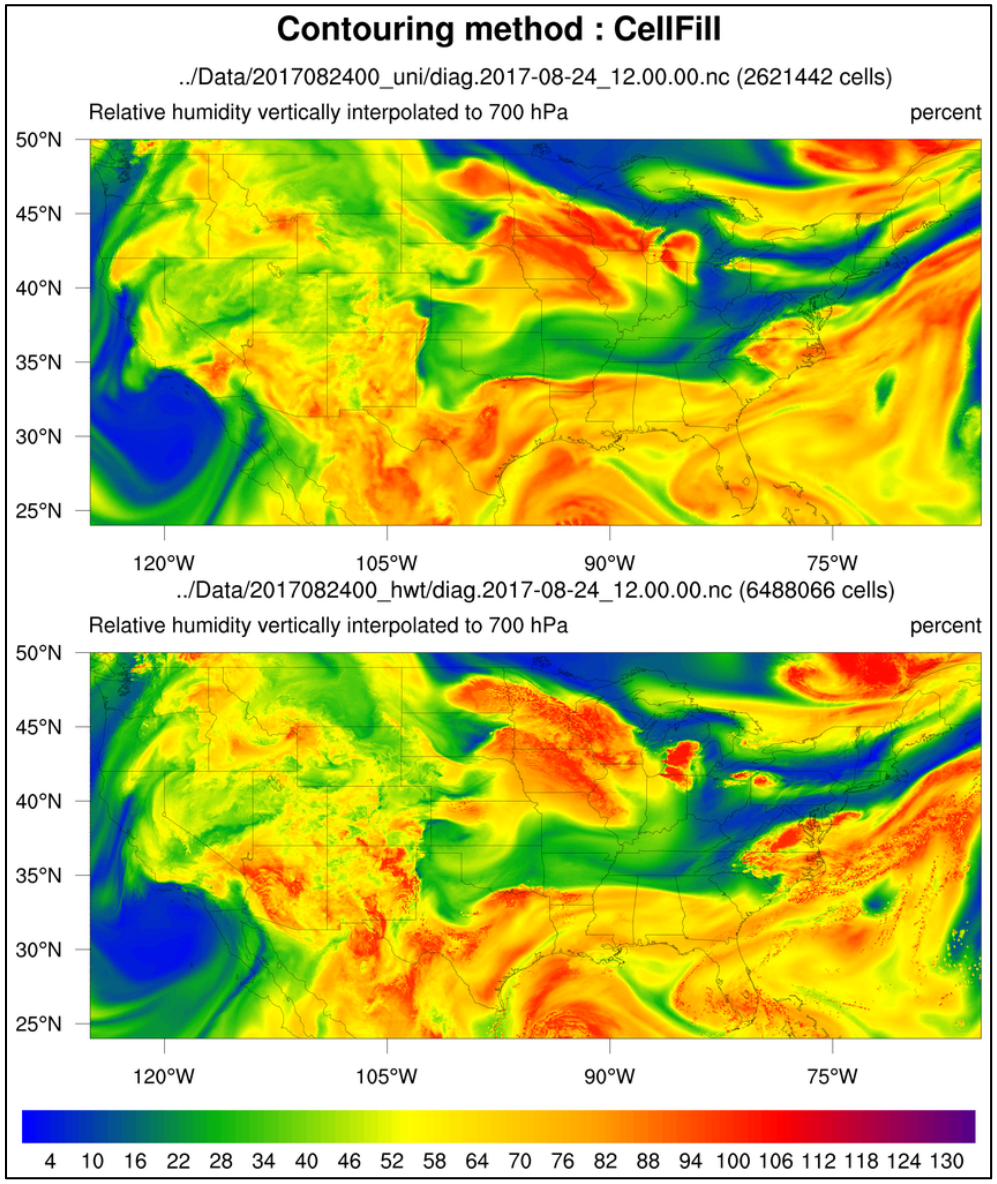


215 seconds

Comparing zoomed in raster and cell filled plots



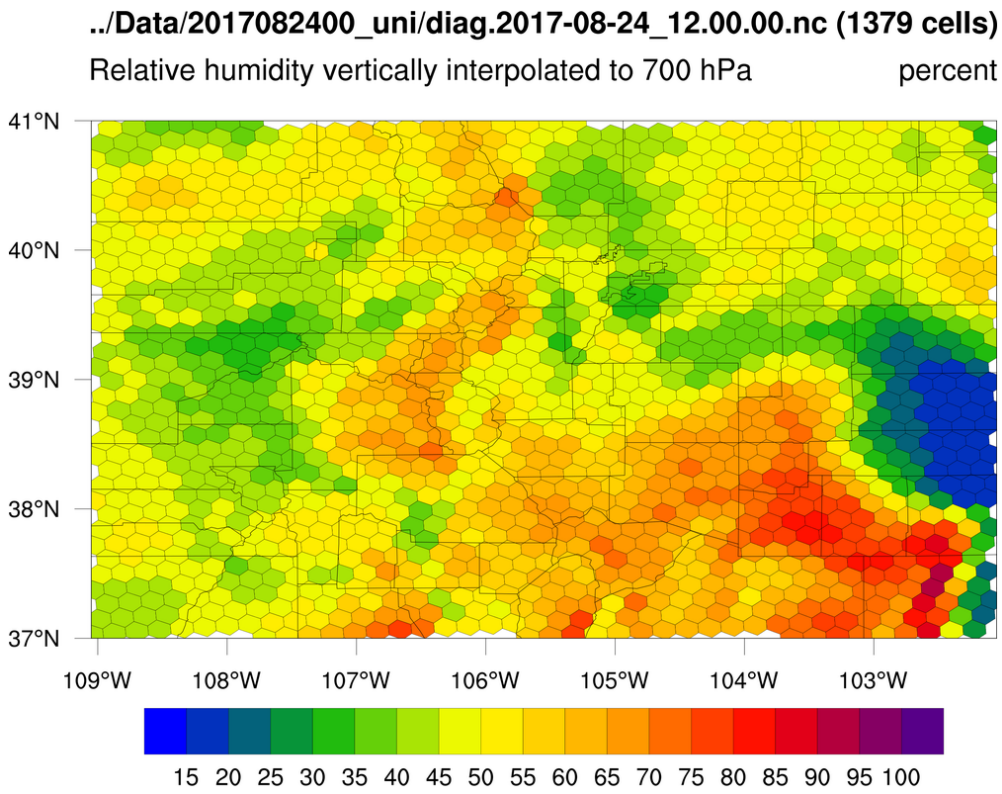
8.1 seconds



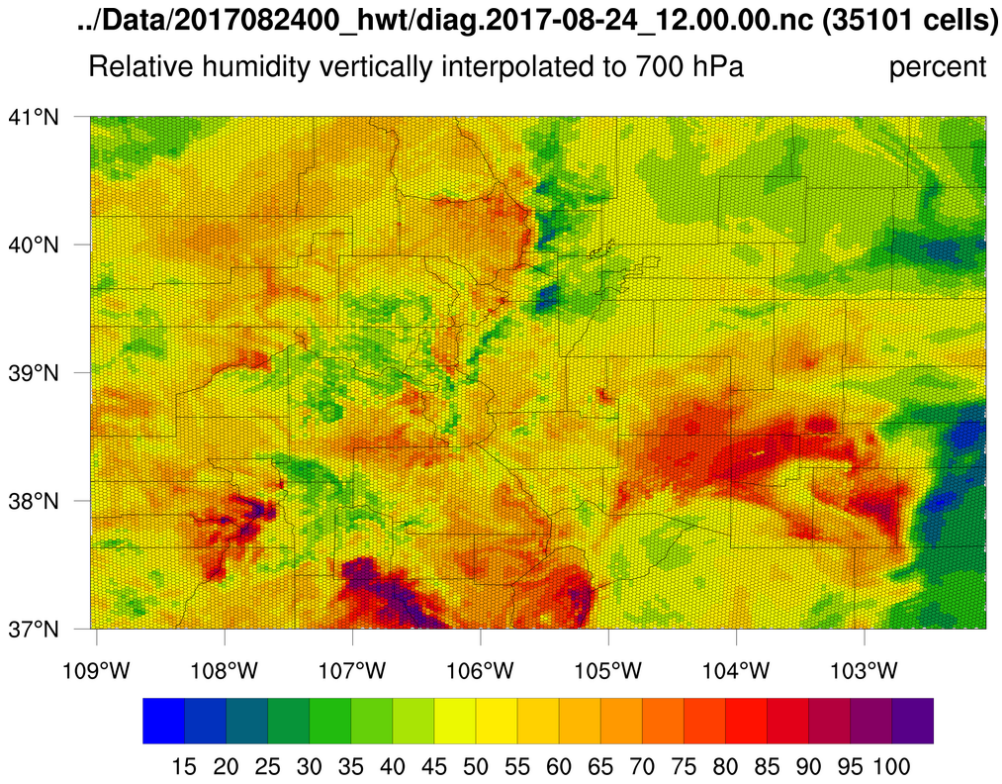
142 seconds

Demo #4 – Contouring MPAS

Using RasterFill and CellFill

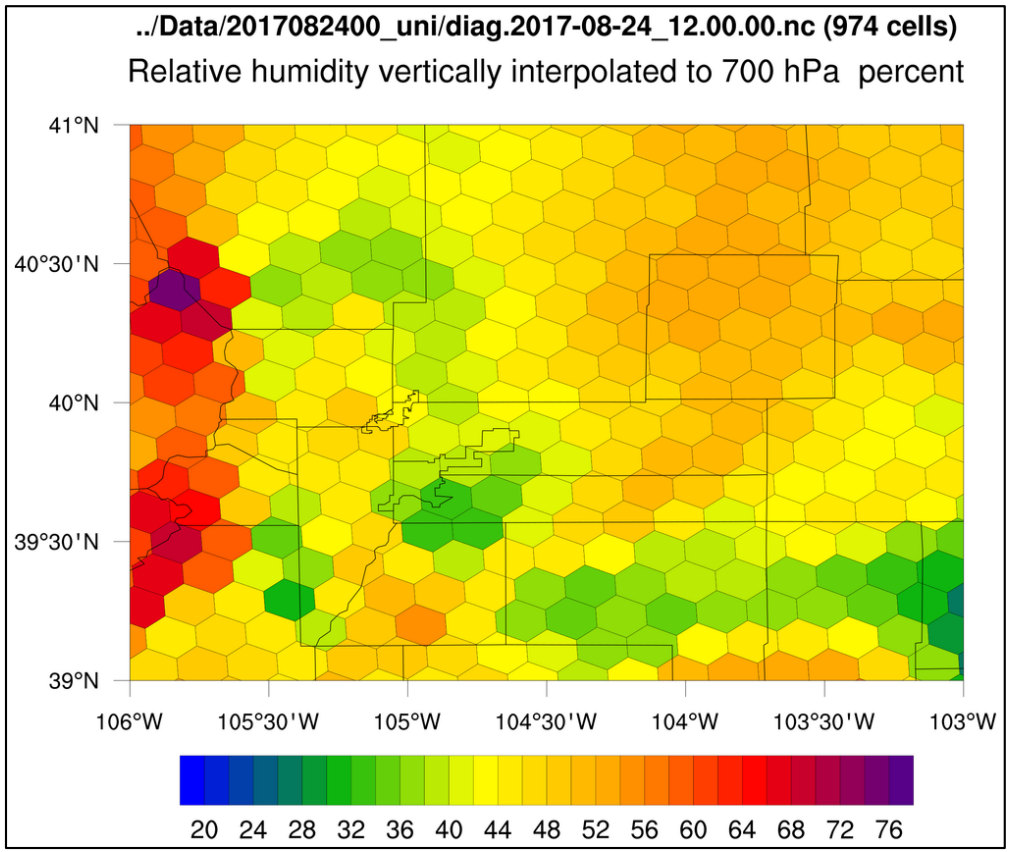


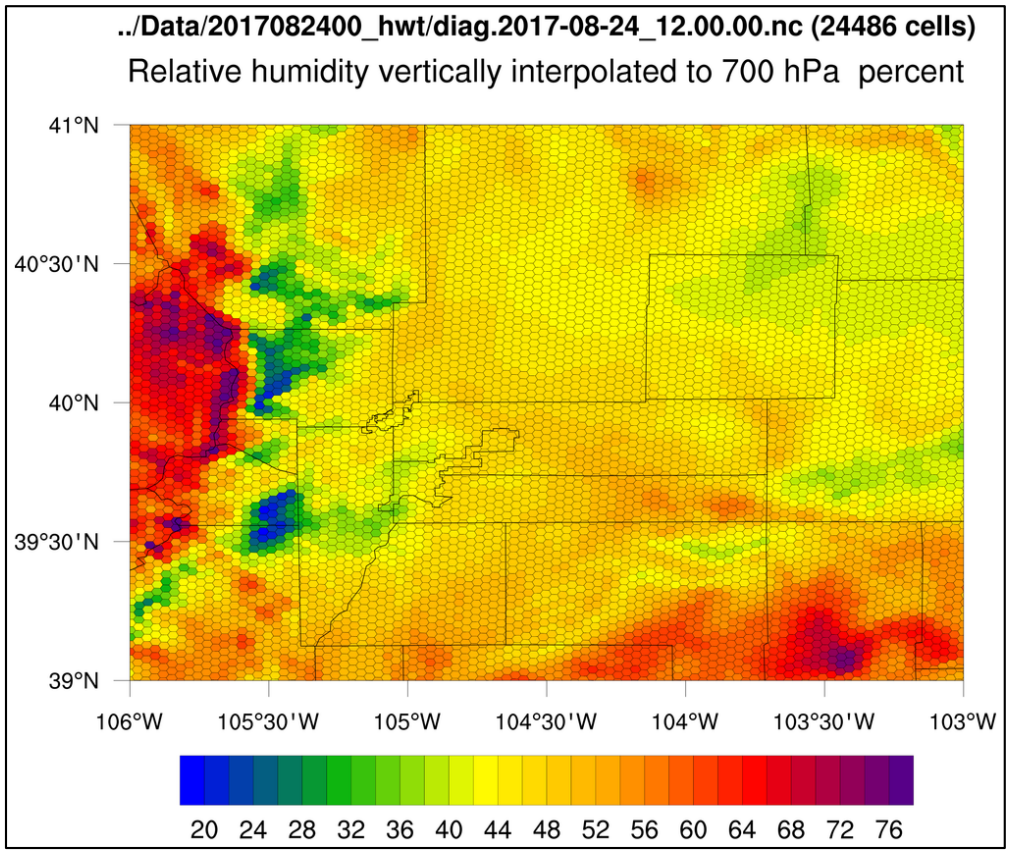
CellFill allows us to outline the cell edges



Zoom in even further

(Colorado counties)





The (somewhat) good news about CellFill

Once you calculate "verticesOnCell", you can write to a NetCDF file and then use them in subsequent scripts.

Outline

- Looking at your data
- Plotting WRF-ARW data
- Contouring MPAS data
- Tips for debugging, customizing graphics

Debugging tips

- Start with an existing script, if possible
- Use editor enhancements for coloring of syntax, comments, resources, functions, etc
- Use indentation (even though not needed)
- Use printVarSummary, printMinMax, print to examine variables
- Carefully read documentation for functions
- Read errors and warnings carefully ☺

Tips for debugging graphics scripts

- Make sure spelling the resource name correctly
- Use editor enhancements to help w/resources
- Use the NCL Applications Page for help!
- Use "x11" output while debugging

  (except maybe for really slow graphical scripts)
- Set gsnMaximize resource to True for largest possible plot

Is X11 window too small?

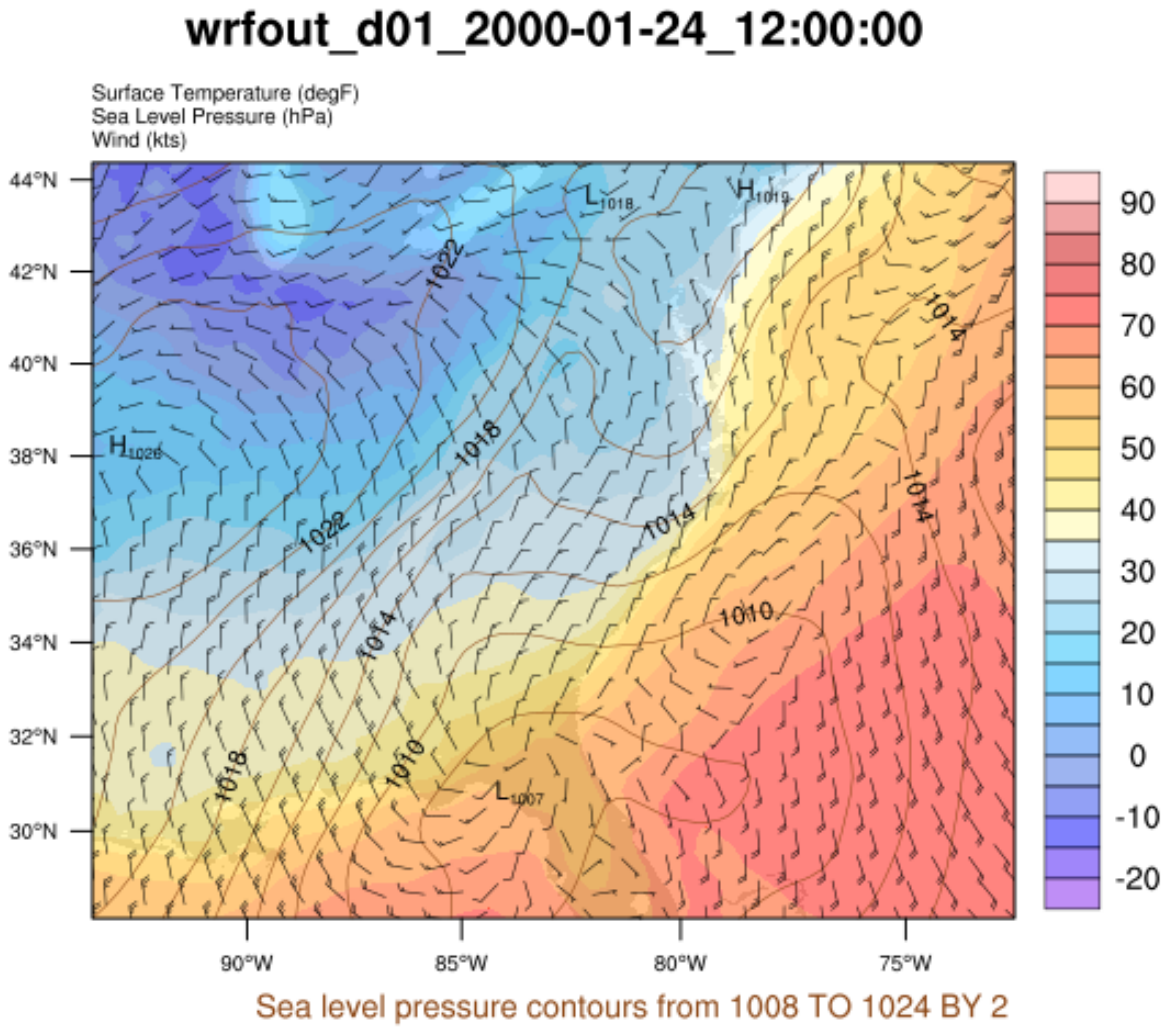
Put these two lines in your "~/.hluresfile

```
*windowWorkstationClass*wkWidth  : 1400
*windowWorkstationClass*wkHeight : 1400
```

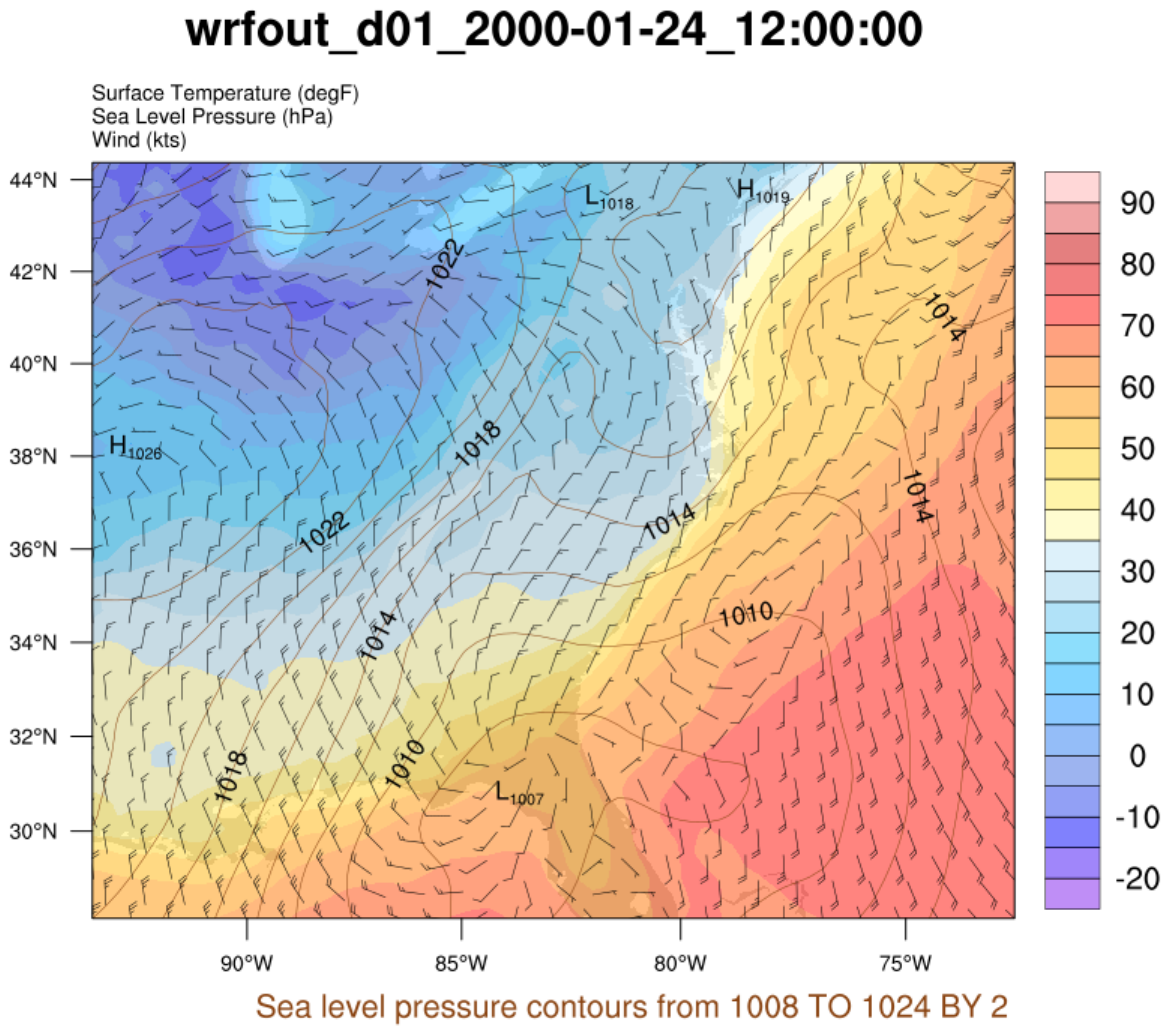
Tips for nice graphics

- Increase line thicknesses
- Use bold fonts or larger fonts
- Increase resolution of PNG images
- Trim white space using "convert"
- Use color wisely
- Use shapefile outlines for better map outlines

Sample blurry-ish image, and why so much white space?

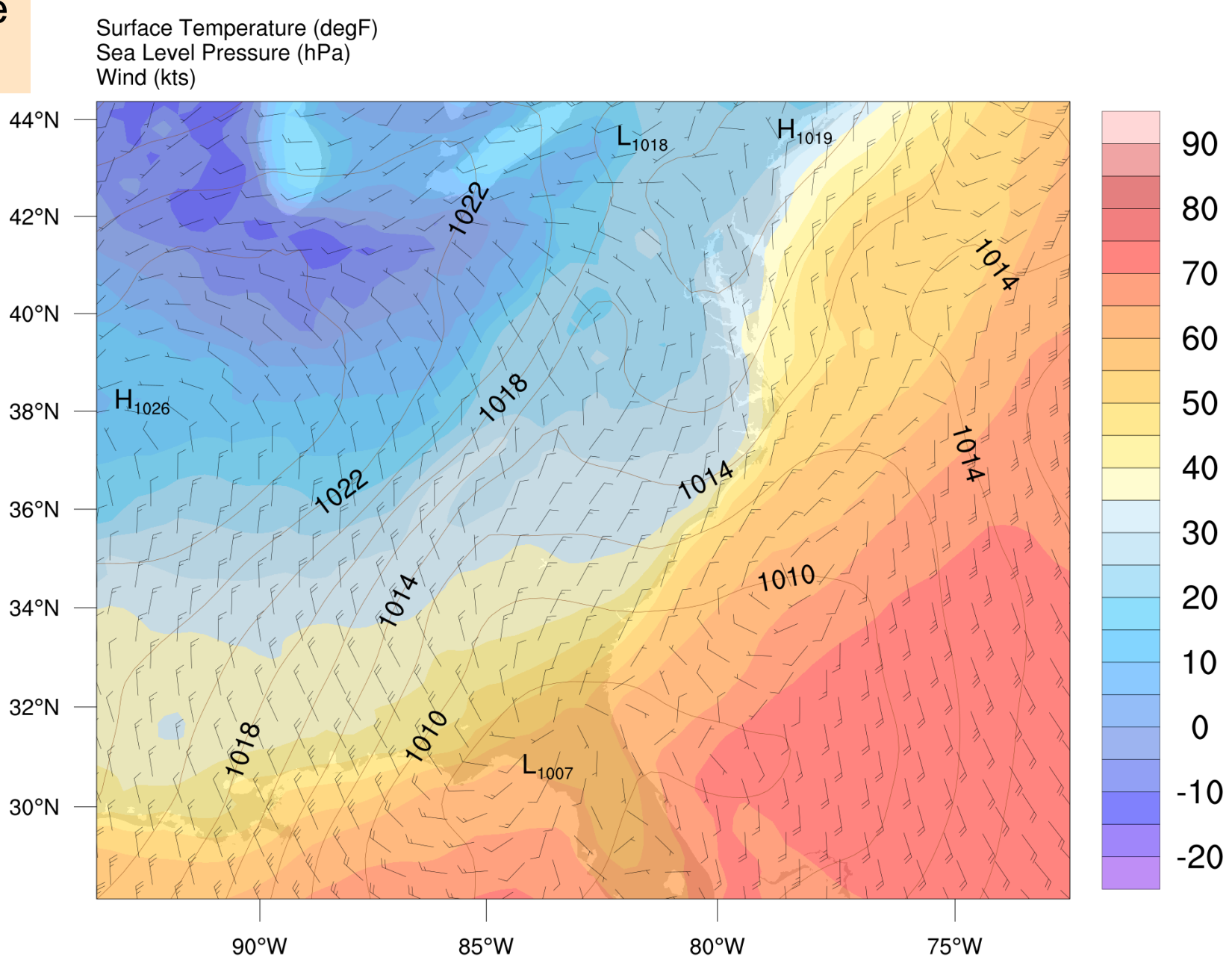


A little better (1024 x 1024), but still…with the white space!



Better!
Trimmed the image

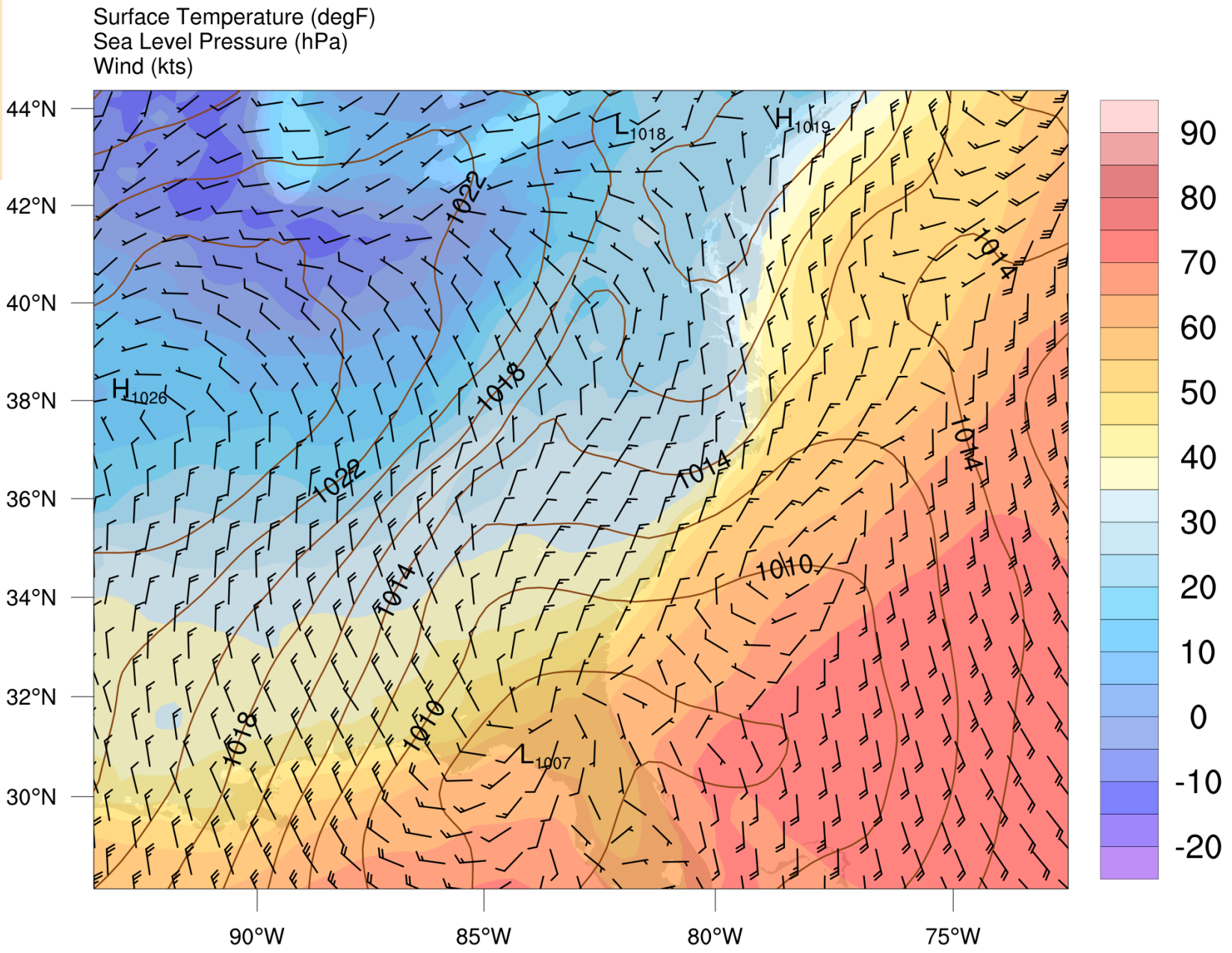
wrfout_d01_2000-01-24_12:00:00



Sea level pressure contours from 1008 TO 1024 BY 2

Much better! Increased line thickness and used bold font

wrfout_d01_2000-01-24_12:00:00



Sea level pressure contours from 1008 TO 1024 BY 2

Improving PNG images

Set the PNG size in NCL script

```
wtype           = "png"
wtype@wkWidth   = 2500
wtype@wkHeight  = 2500
wks             = gsn_open_wks(wtype,"plot")
```

Increase line thicknesses, font sizes use bold font

```
res@cnLineThicknessF = 4.0
res@vcWindBarbThicknessF = 4.0
res@cnInfoLabelFont = "Helvetica-bold"
res@tmXBLabelFontHeightF = 0.02
```

Use ImageMagick's "convert" to trim, add a small border

From UNIX command line:

```
convert -trim -border 8 -bordercolor white plot.png plot.png
```

Tips for plotting large grids / meshes

- Use "res@cnFillMode = "RasterFill" (or "CellFill")
- Use a small number of contour levels while debugging
- Zoom in on the map (smaller area is rendered, so it's faster)
- Use "ind" or "where" functions to further subset data
- Drawing to a PNG file is faster!

Tips for new and advanced users

- Read the NCL User Guide
- Visit the NCL Examples page
- Join the ncl-talk email list
- Install a UNIX editor enhancement for NCL

WRF-Python Tutorial at 10:30 AM

Bill Ladwig tutorial on WRF-Python

"Analyzing and Visualizing WRF-ARW Data

Using WRF-Python and other Python Tools"