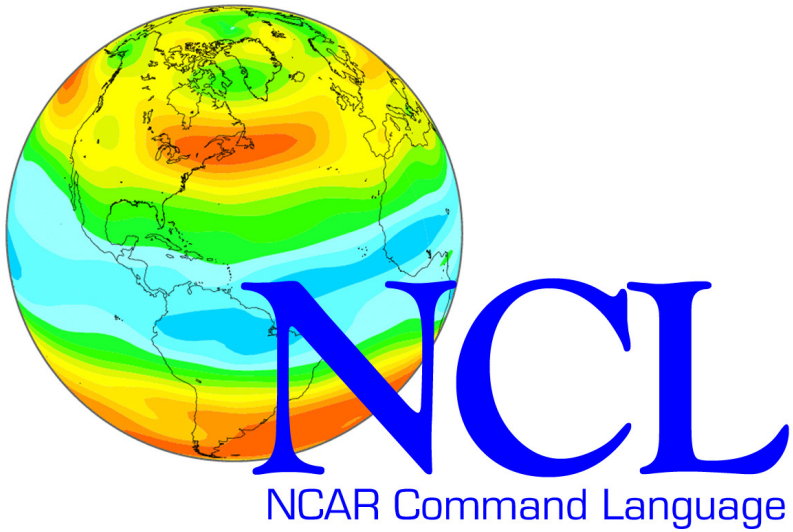
Introduction to NCL Graphics



Part 2 in a series

September 26, 2014

Mary Haley

Sponsored by the National Science Foundation

CISL

NESL

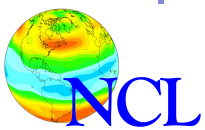
NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Notes

- Second in a series of lectures on NCL Graphics
- Still don't know yet how many in the series ☺
- Lectures based on slides used in our NCL Workshops
- Geared towards new users of NCL
- Assumptions:
  - You are familiar with basic NCL language features
  - You are familiar with basic structure of an NCL graphics script (covered in first lecture)
  - You know about the NCL/NetCDF data model (attributes, coordinate arrays, dimension names, missing values)

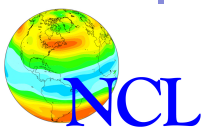
Goals for this series of lectures

- Get you comfortable with creating plots in NCL
- Show you the most common things people do with NCL graphics
- Give you tips for editing, debugging, creating nice graphics
- Answer any questions you may have about NCL graphics



NCL Graphics topics for this lecture

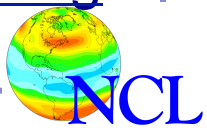
- Line-by-line example of creating a contour plot script
- Interactive demo



Example *contour1a.ncl*

- gsn_csm_contour
- Simple contour plot
- Data retrieved from netCDF file
- Data has _FillValue attribute, which is automatically "seen" by gsn_csm script
- No plot options (resources) set

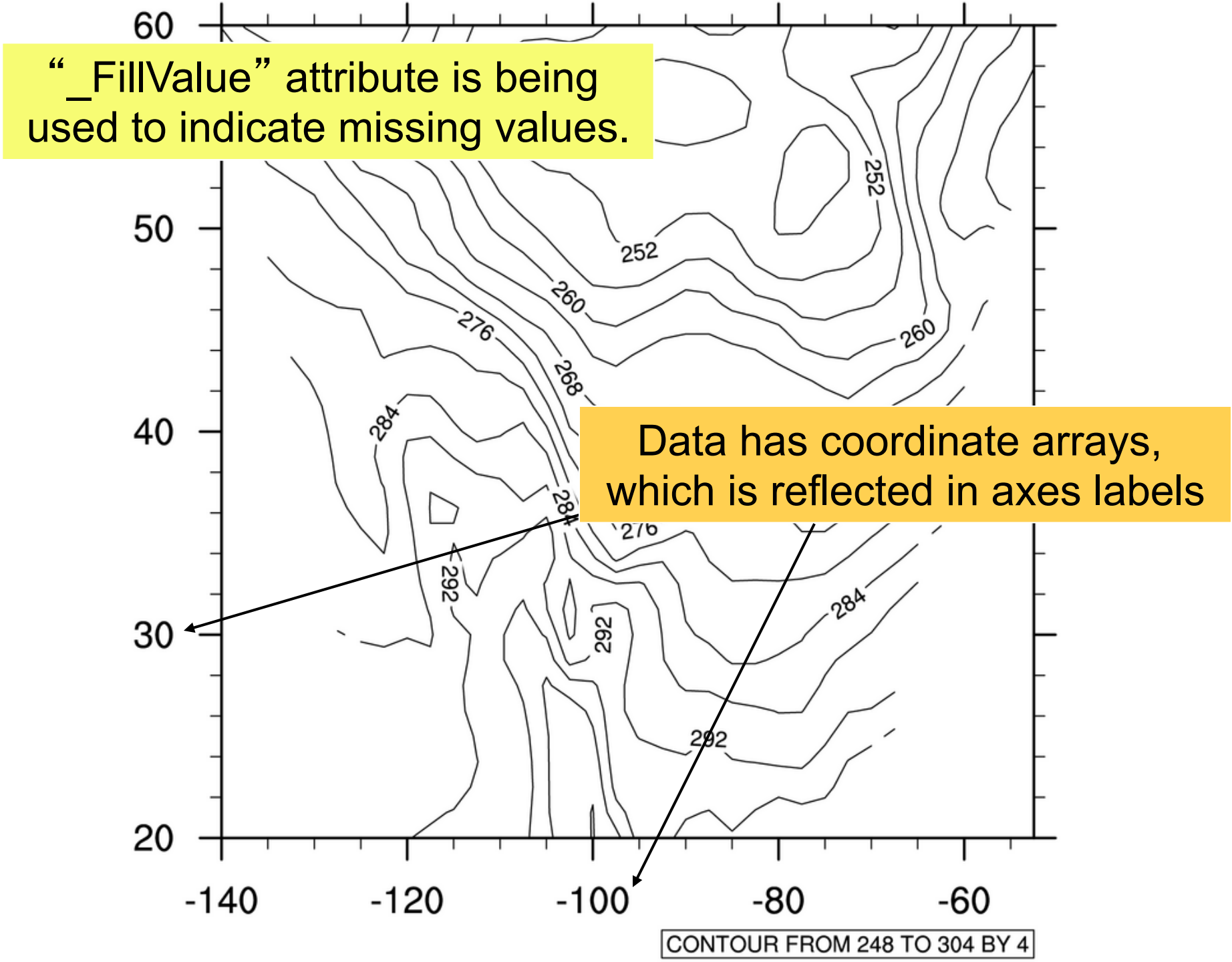
http://www.ncl.ucar.edu/Training/Workshops/Scripts/#Contouring



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

begin
  tf = addfile("Tstorm.cdf","r")
  T  = tf->t(0,:,:)       ; Read first time step
                          ; of temperature data.
  printVarSummary(T)      ; Very useful for debugging!

  wks = gsn_open_wks("ps","contour1a")

  res = True              ; No plot options set.
  plot = gsn_csm_contour(wks,T,res)
end
```

Output from "printVarSummary(T)"

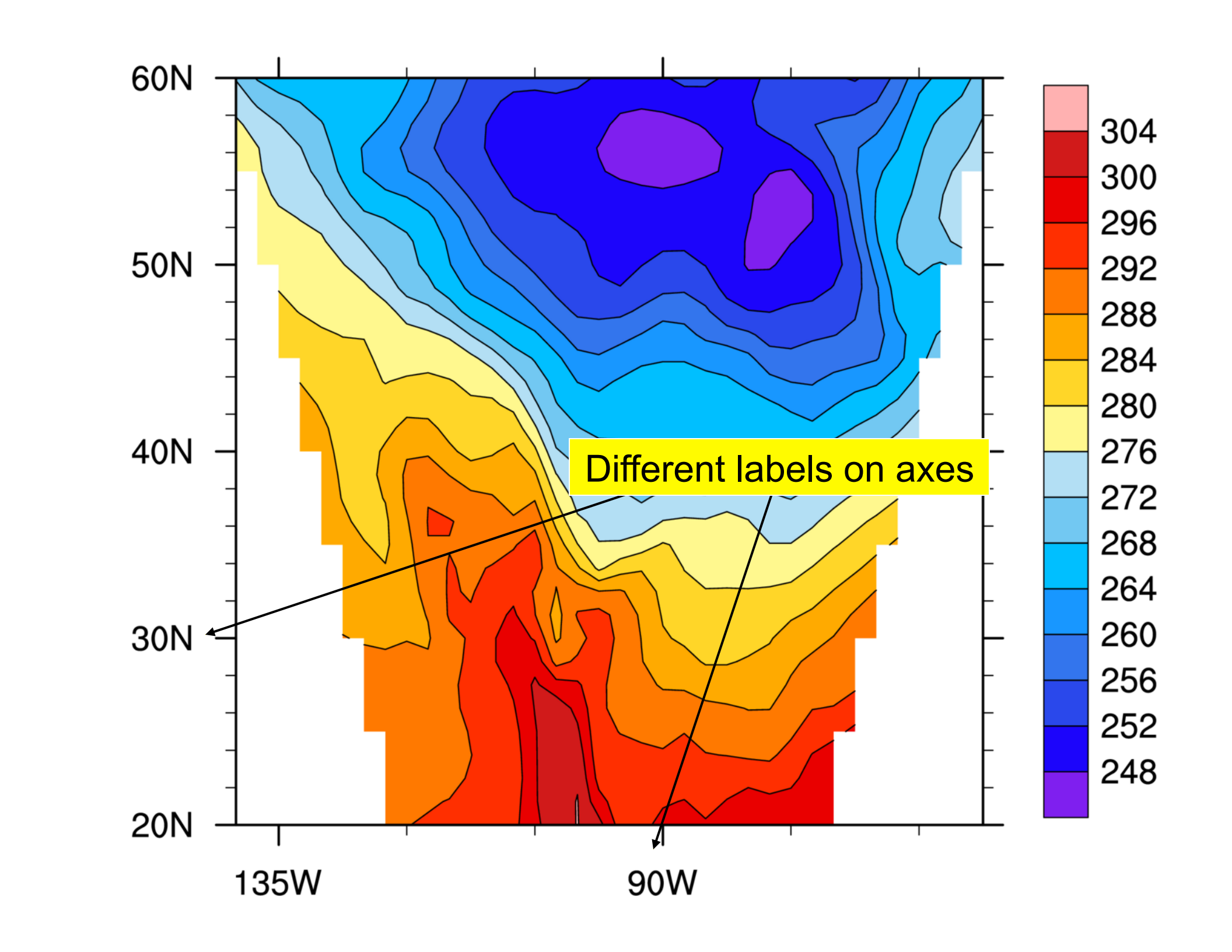
```
Variable: T
Type: float
Total Size: 4752 bytes
            1188 values
Number of Dimensions: 2
Dimensions and sizes:   [lat | 33] x [lon | 36]
Coordinates:
            lat: [20..60]
            lon: [-140..-52.5]
Number Of Attributes: 2
  timestep :    0
  _FillValue :  -9999
```


"_FillValue" attribute is being used to indicate missing values.
Data has coordinate arrays, which is reflected in axes labels
60
50
40
30
20
-140
-120
-100
-80
-60
252
260
276
268
284
292
CONTOUR FROM 248 TO 304 BY 4

Example *contour1b.ncl*

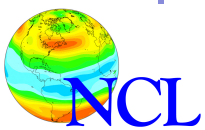
- "units" attribute added to both lat and lon coordinate arrays
- Contour fill turned on
- Resources introduced:
  - cnFillOn - turn on contour fill
  - lbOrientation - change labelbar orientation

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

begin
  tf = addfile("Tstorm.cdf","r")
  T  = tf->t(0,:,:)                ; Get first time step
  T&lon@units = “degrees_east”  ; Add some units
  T&lat@units = “degrees_north”

  wks = gsn_open_wks("ps","contour1b")

  res                  = True
  res@cnFillOn         = True        ; Turn on contour fill
  res@lbOrientation    = “Vertical”  ; Move labelbar

  plot = gsn_csm_contour(wks,T,res)
end
```

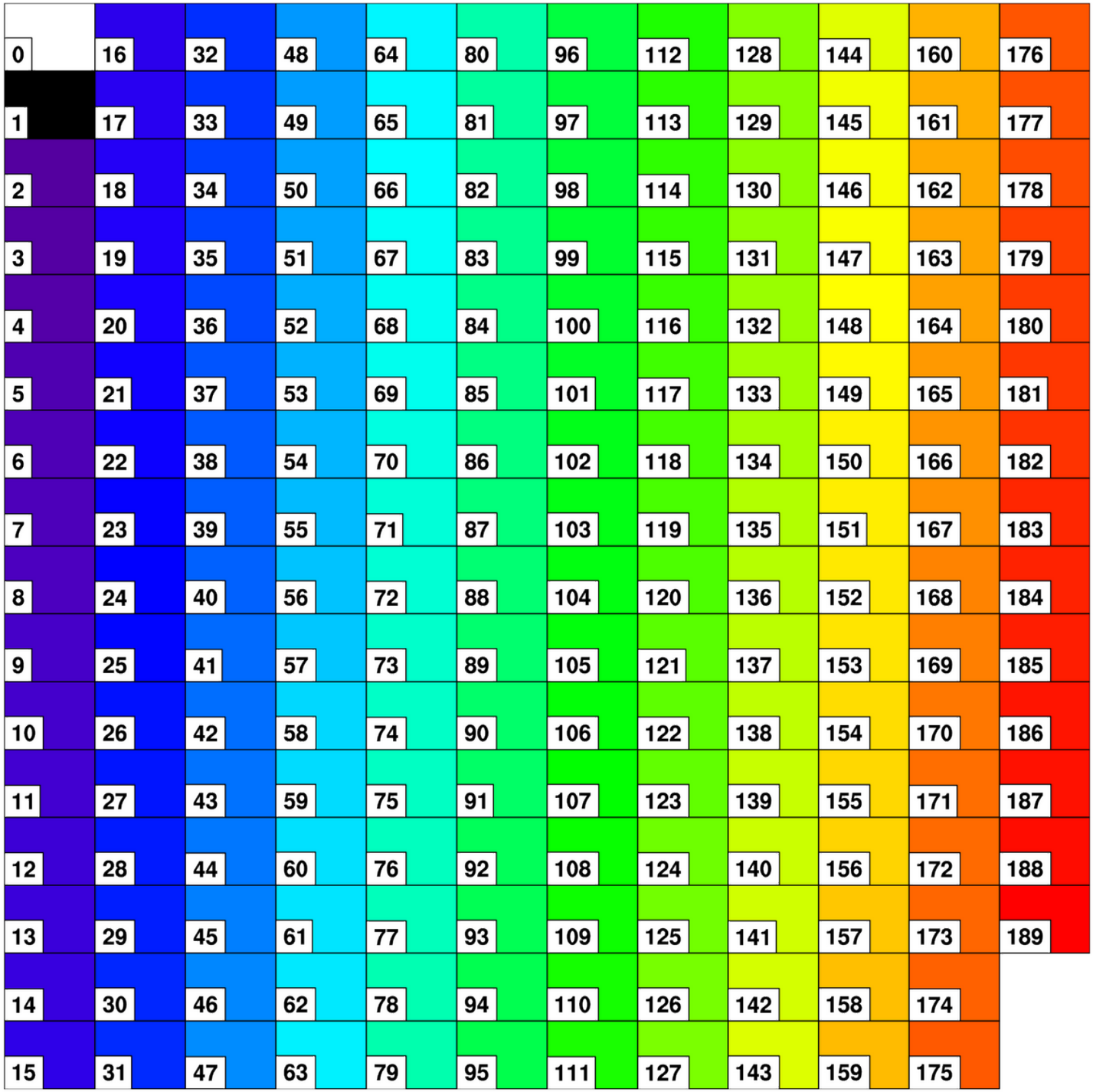

60N
50N
40N
30N
20N
135W
90W
304
300
296
292
288
284
280
276
272
268
264
260
256
252
248
Different labels on axes

Example *contour1d.ncl*

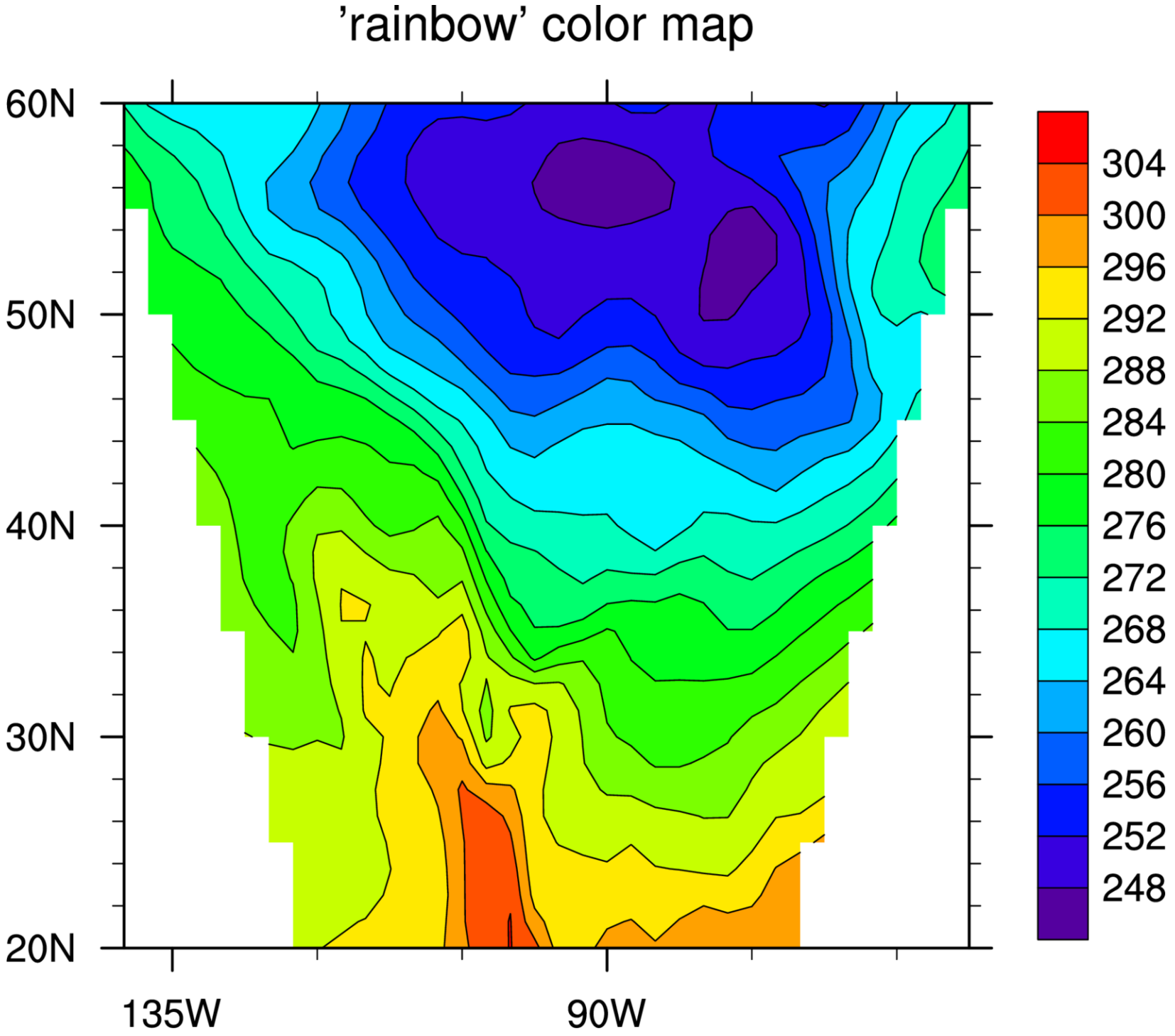
- Color map changed
- Title added
- Resources introduced:
  - cnFillPalette – change contour color map

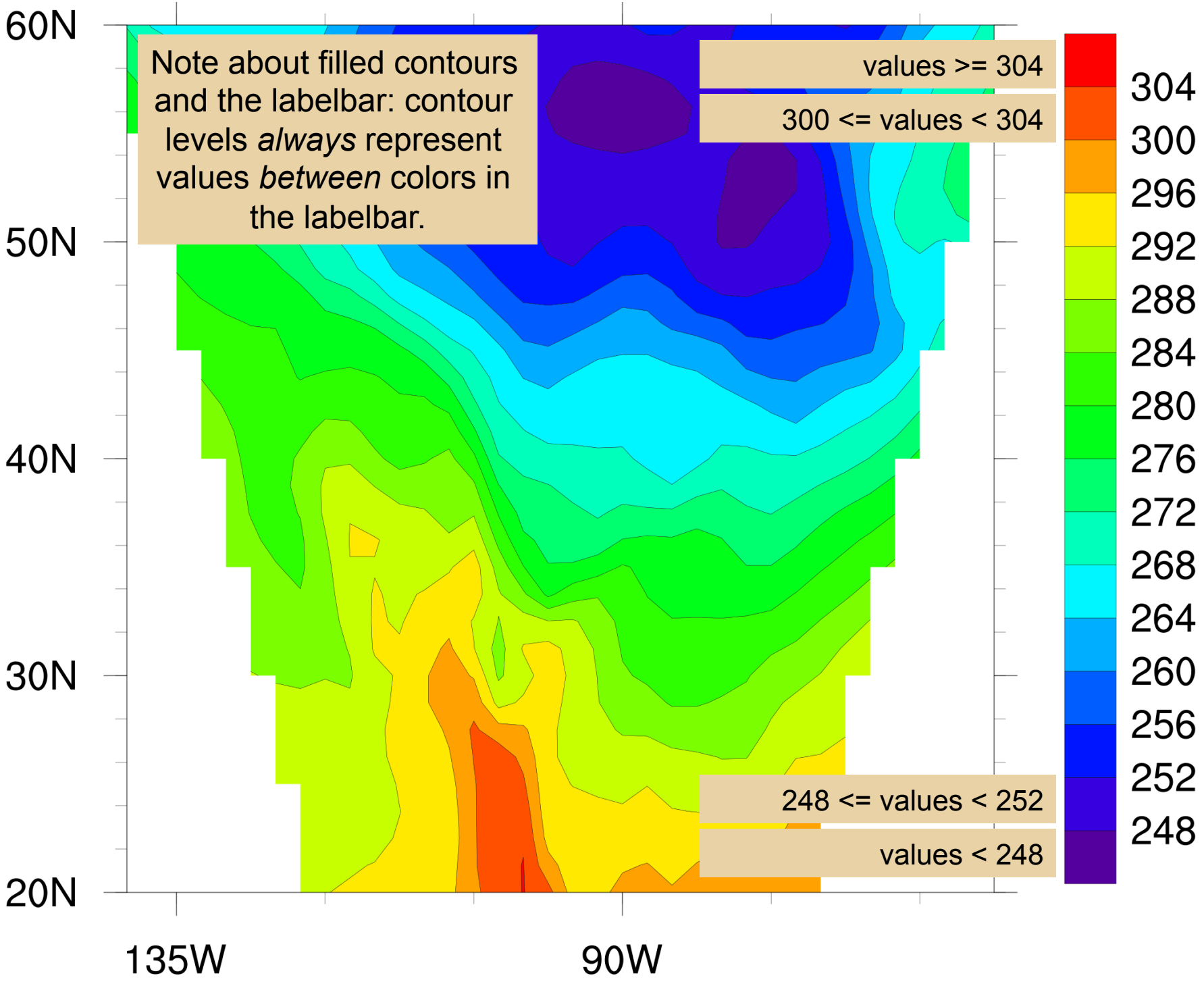


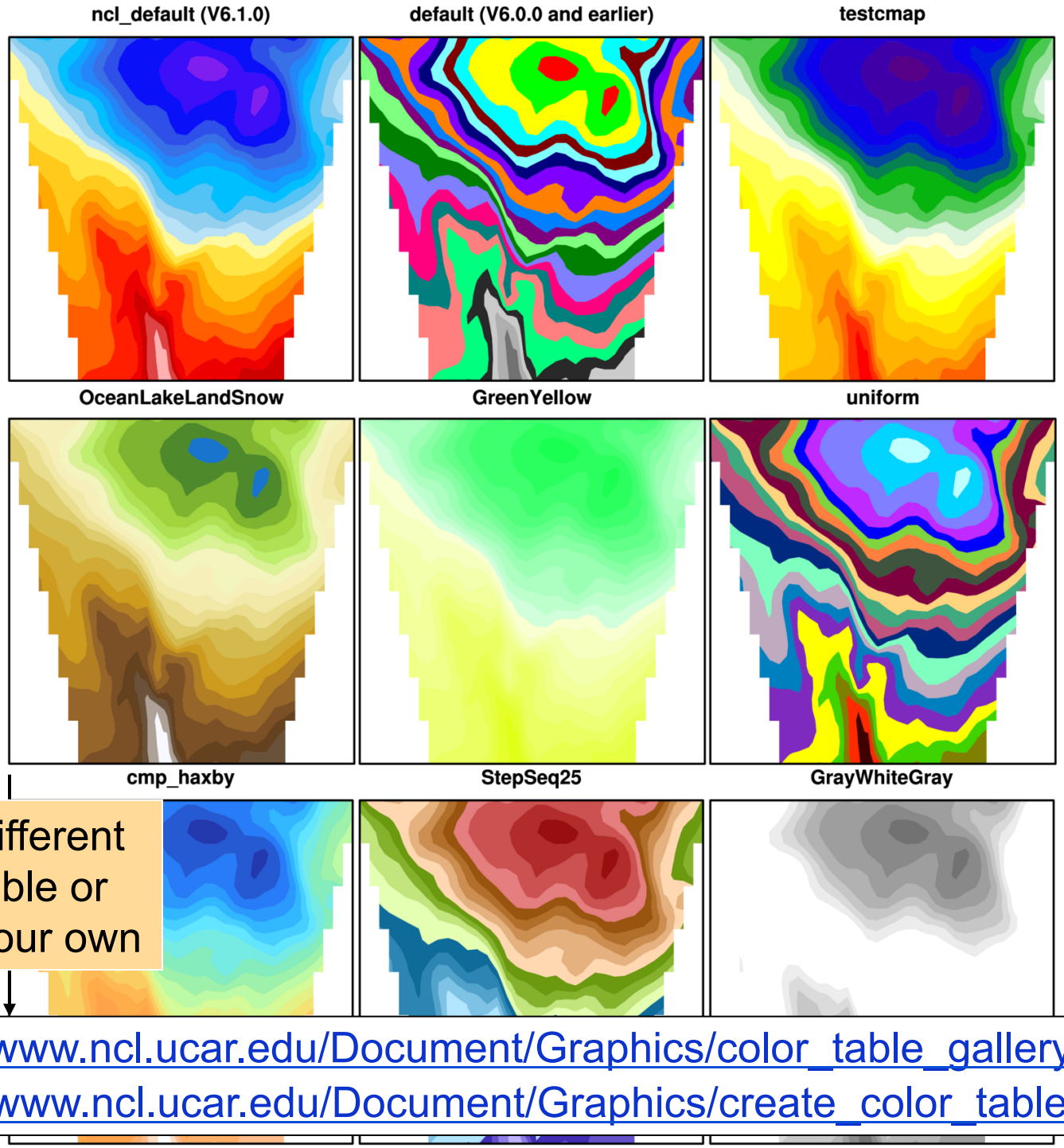
“rainbow” color map



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

begin
  tf = addfile("Tstorm.cdf","r")
  T  = tf->t(0,:,:)
  T&lon@units = “degrees_east”  ; Add some units
  T&lat@units = “degrees_north”

  wks = gsn_open_wks("ps","contour1d")

 ;;;gsn_define_colormap(wks,"rainbow")    ; OLD WAY

  res                   = True
  res@cnFillOn          = True          ; Turn on contour fill
  res@lbOrientation     = “Vertical”    ; Move labelbar
  res@cnFillPalette     = “rainbow”

  plot = gsn_csm_contour(wks,T,res)
end
```


'rainbow' color map
60N
50N
40N
30N
20N
135W
90W
304
300
296
292
288
284
280
276
272
268
264
260
256
252
248


Note about filled contours and the labelbar: contour levels *always* represent values *between* colors in the labelbar.
values >= 304
300 <= values < 304
248 <= values < 252
values < 248
60N
50N
40N
30N
20N
135W
90W
304
300
296
292
288
284
280
276
272
268
264
260
256
252
248

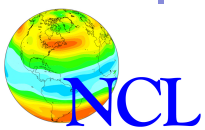


Pick a different color table or create your own

http://www.ncl.ucar.edu/Document/Graphics/color_table_gallery.shtml
http://www.ncl.ucar.edu/Document/Graphics/create_color_table.shtml

NCL Graphics topics for this lecture

- Line-by-line example of creating a contour plot script
- Interactive demo



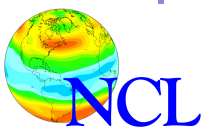
Interactive demo

- Contours over a map
- The contour2x*.ncl slides following this one will be skipped and mimicked in the demo.
- If you want, download "contour_map_template.ncl" and "uv300.nc" from:

  http://www.ncl.ucar.edu/Training/Webinars/NCL_Graphics/

Example *contour2a.ncl*

- gsn_csm_contour_map
- New data file used
- Contour plot overlaid on a cylindrical equidistant map
- Data must have lat/lon coordinate arrays, and "units" must be "degrees_east", "degrees_north"
- No resources introduced (yet)



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

begin
  f = addfile("uv300.nc","r")
  U = f->U(0,:,:)   ; U will have metadata attached.

  printVarSummary(U)      ; U has coord arrays and attrs
  printVarSummary(U&lon)  ; U&lon has "units" attribute

  wks = gsn_open_wks("ps","contour2a")   ; "contour2a.ps"

  res = True                    ; No plot options set.
  plot = gsn_csm_contour_map(wks,U,res)
end
```

Output from "printVarSummary(U)"

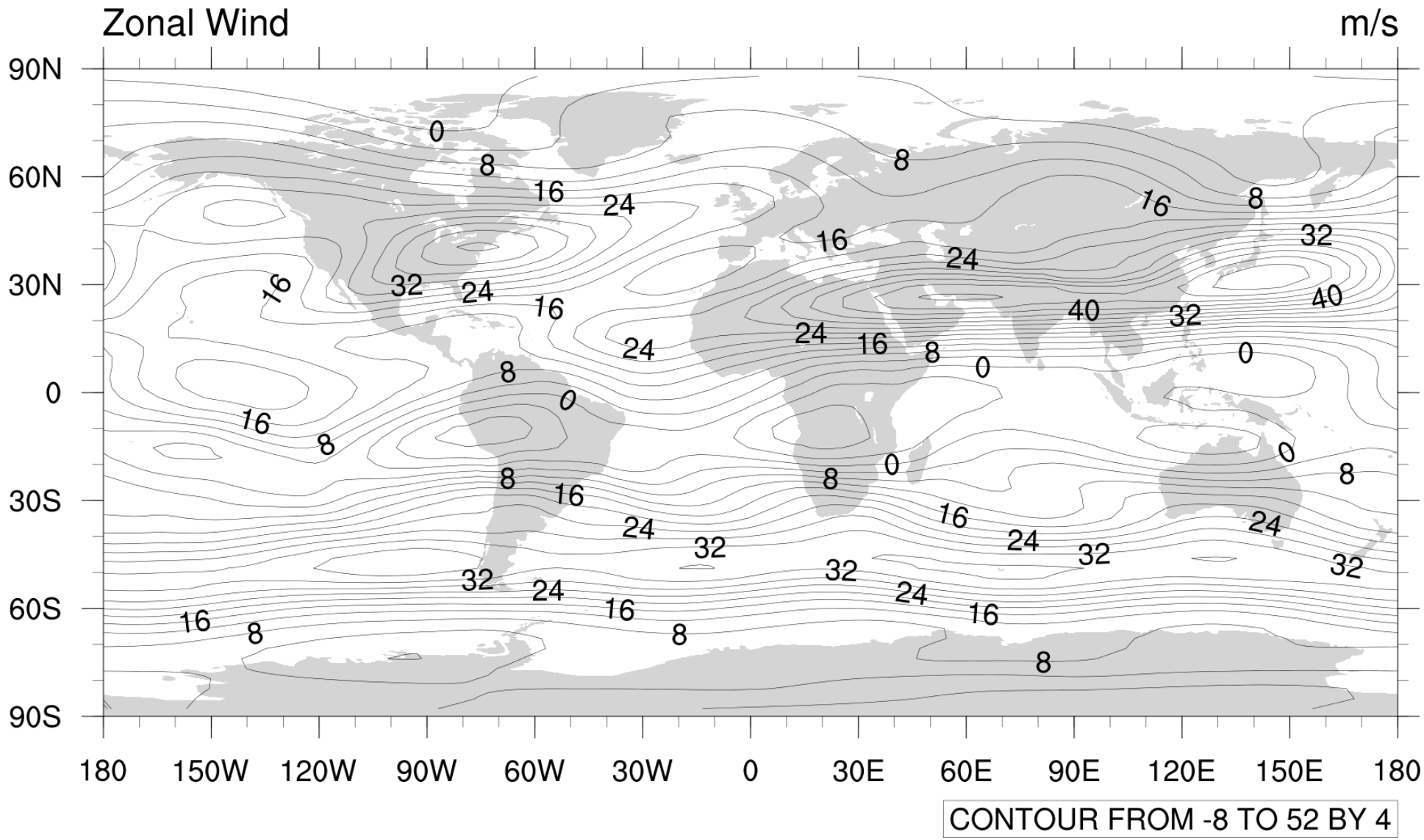
```
Variable: U
Type: float
Total Size: 32768 bytes
            8192 values
Number of Dimensions: 2
Dimensions and sizes:   [lat | 64] x [lon | 128]
Coordinates:
            lat: [-87.8638..87.8638]
            lon: [-180..177.1875]
Number Of Attributes: 5
  time :    1
  _FillValue :    -999
  long_name :     Zonal Wind
  short_name :    U
  units :   m/s
```

Longitude data is not cyclic. gsn_csm_xxxx scripts will try to add cyclic point by default.

Output from "printVarSummary(U&lon)"

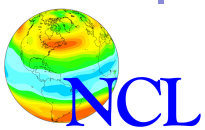
```
Variable: lon (coordinate)
Type: float
Total Size: 512 bytes
            128 values
Number of Dimensions: 1
Dimensions and sizes:    [lon | 128]
Coordinates:
Number Of Attributes: 3
  units : degrees_east
  long_name :  longitude
  short_name : lon
  units :        degrees_east
```

Similarly, U&lat has "degrees_north" attribute

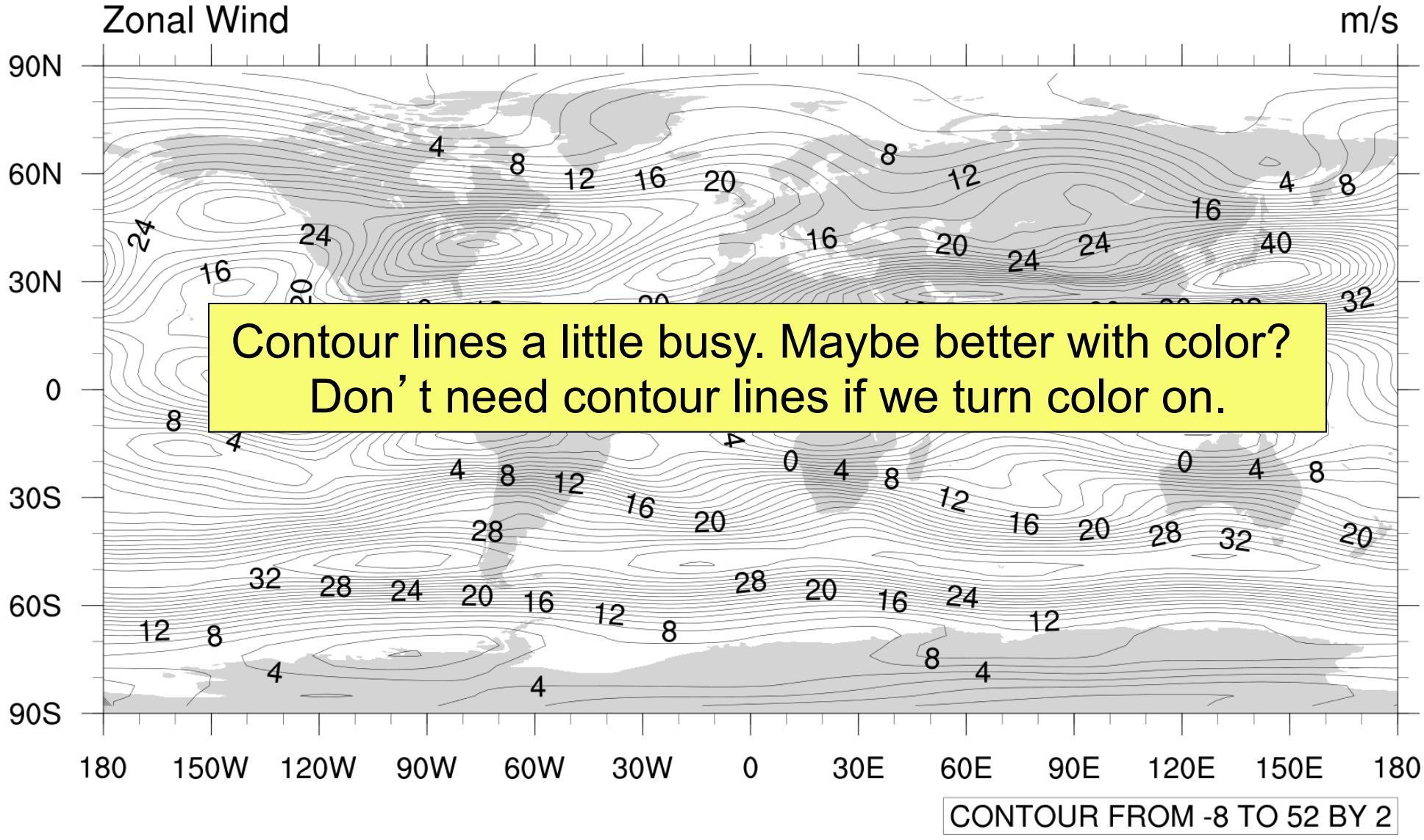

Zonal Wind
m/s
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180
CONTOUR FROM -8 TO 52 BY 4

Example *contour2b.ncl*

- Contour levels set manually
- Resources introduced:
  - cnLevelSelectionMode - mode for setting contour levels
  - If "cnLevelSelectionMode" is "ManualLevels", then use these 3 resources: cnMinLevelValF, cnMaxLevelValF, cnLevelSpacingF
- Can set "cnLevelSpacingF" by itself.



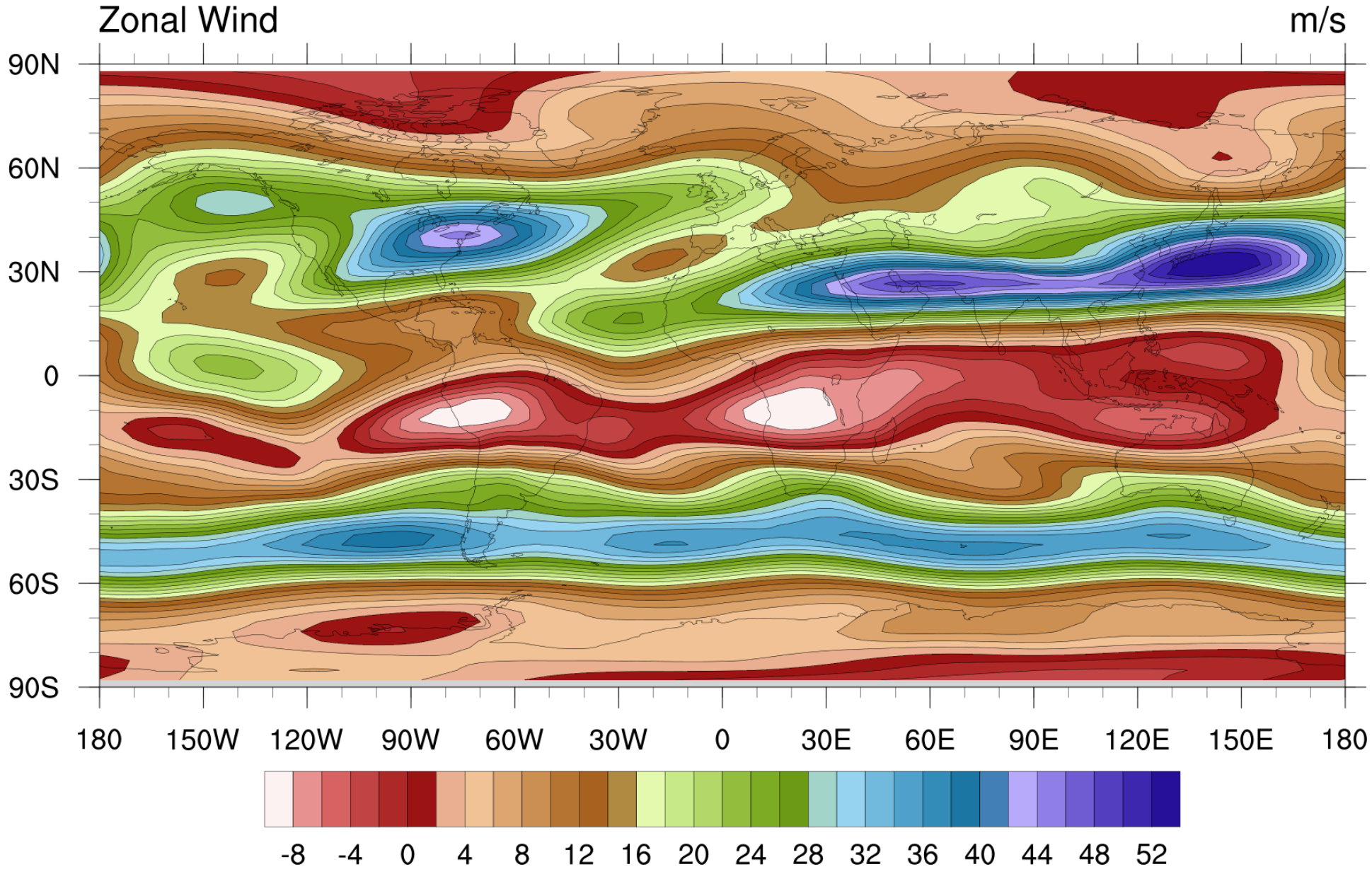
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl”

begin
  f = addfile(”uv300.nc","r")
  T = f->U(0,:,:)

  wks = gsn_open_wks("ps","contour2b")

  res                          = True
  res@cnLevelSelectionMode = "ManualLevels"
  res@cnMinLevelValF       = -8   ; Min contour
  res@cnMaxLevelValF       = 52   ; Max contour
  res@cnLevelSpacingF      =  2   ; Spacing

 ; res@cnLevelSelectionMode = “ExplicitLevels”
 ; res@cnLevels    = (/-8,0,10,15,20,30,45,50/)

  plot = gsn_csm_contour_map(wks,U,res)
end
```


Zonal Wind
m/s
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180
Contour lines a little busy. Maybe better with color?
Don’t need contour lines if we turn color on.
CONTOUR FROM -8 TO 52 BY 2

Example *contour2c.ncl*

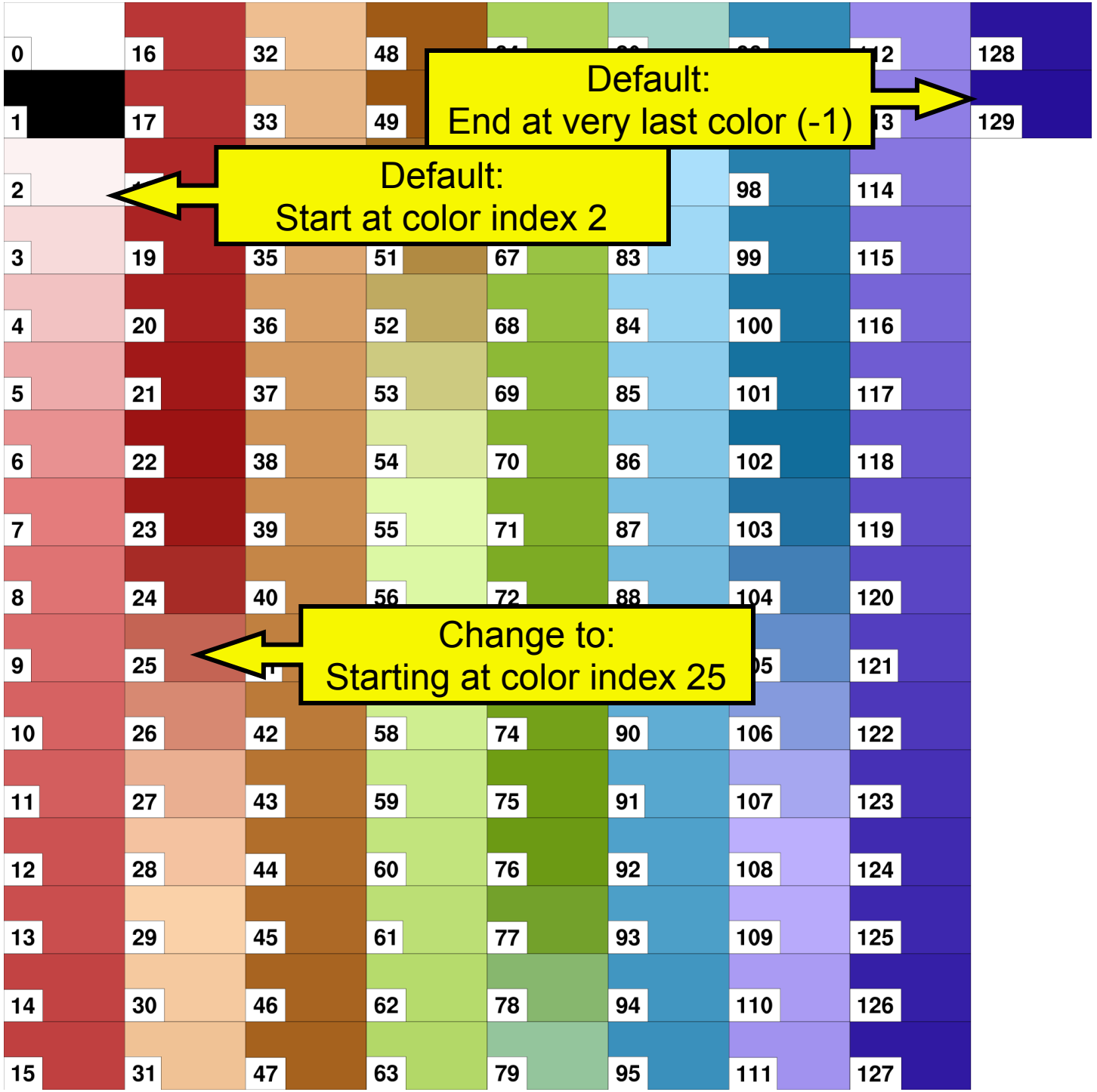
- Contour fill turned on
- Color map changed

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

begin
  f = addfile(”uv300.nc","r")
  T = f->U(0,:,:)

  wks = gsn_open_wks("ps","contour2c")

  res                           = True
  res@cnLevelSelectionMode = "ManualLevels"
  res@cnMinLevelValF       = -8   ; Min contour
  res@cnMaxLevelValF       = 52   ; Max contour
  res@cnLevelSpacingF      =  2   ; Spacing

  res@cnFillOn             = True
  res@cnFillPalette        = “MPL_StepSeq”

  plot = gsn_csm_contour_map(wks,U,res)
end
```

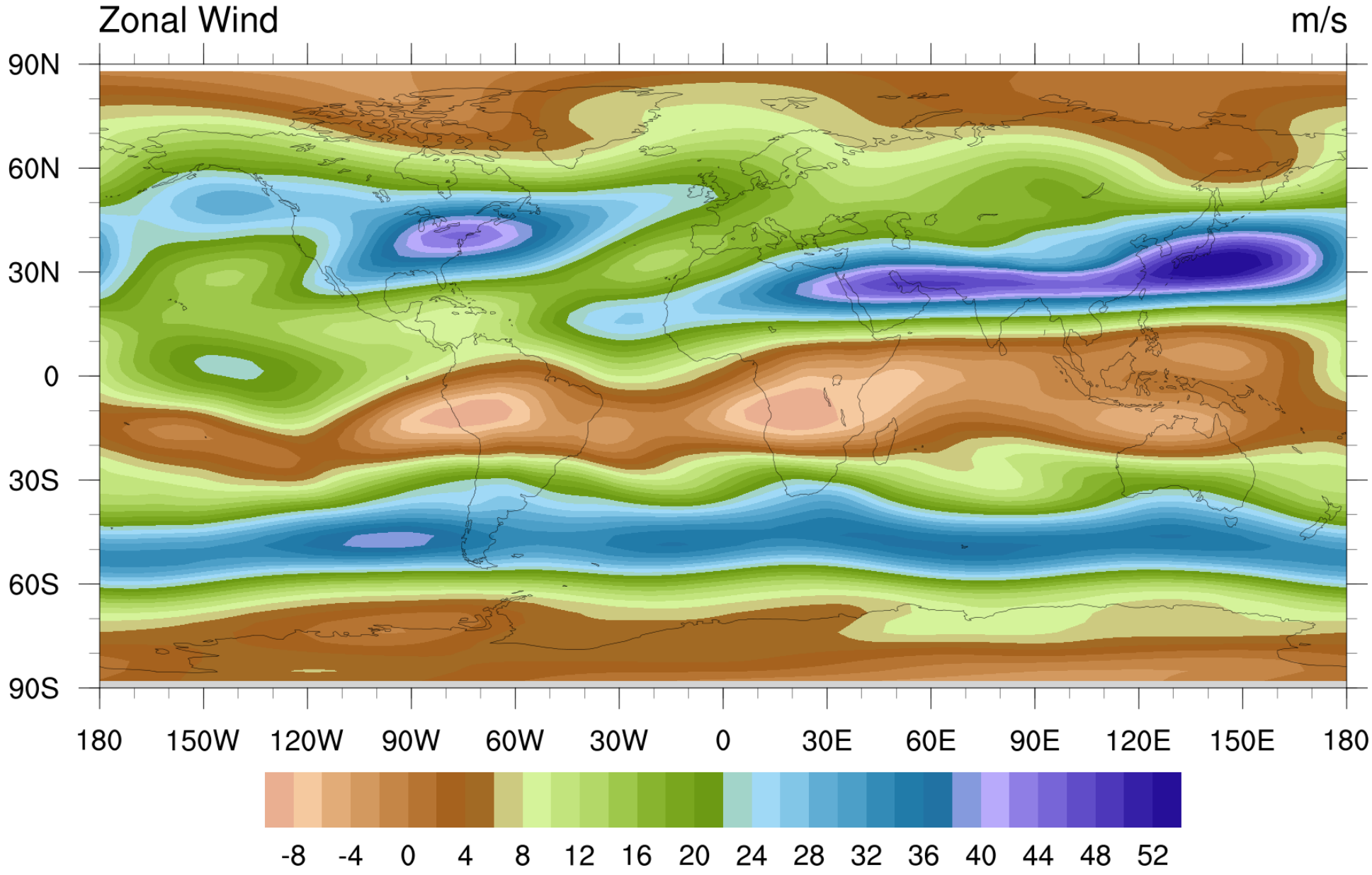

Zonal Wind
m/s
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180
-8
-4
0
4
8
12
16
20
24
28
32
36
40
44
48
52

Example *contour2d.ncl*

- Contour and labelbar box lines turned off
- Only part of color map spanned
- Resources introduced:
  - cnLinesOn - turns contour lines on/off
  - lbBoxLinesOn - turns labelbar box lines on/off


Default:
End at very last color (-1)
Default:
Start at color index 2
Change to:
Starting at color index 25

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

begin
  f = addfile(”uv300.nc","r")
  T = f->U(0,:,:)

  wks = gsn_open_wks("ps","contour2d")

  res                          = True
  res@cnLevelSelectionMode = "ManualLevels"
  res@cnMinLevelValF       = -8   ; Min contour
  res@cnMaxLevelValF       = 52   ; Max contour
  res@cnLevelSpacingF      =  2   ; Spacing

 cmap = read_colormap_file(“MPL_StepSeq”)  ; 130 x 3 RGB array
 res@cnFillPalette          = cmap(25:,:)   ; Start at index 25
 res@cnLinesOn              = False         ; contour lines off
 res@lbBoxLinesOn           = False         ; labelbar box
                                            ; lines off

  plot = gsn_csm_contour_map(wks,U,res)
end
```


Zonal Wind
m/s
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180
-8
-4
0
4
8
12
16
20
24
28
32
36
40
44
48
52

Contouring exercises and examples

http://www.ncl.ucar.edu/Training/Workshops/Exercises/

Click on:

- Contour plot exercises
- Contours over map exercises (set 1)
- Contours over map exercises (set 2)

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

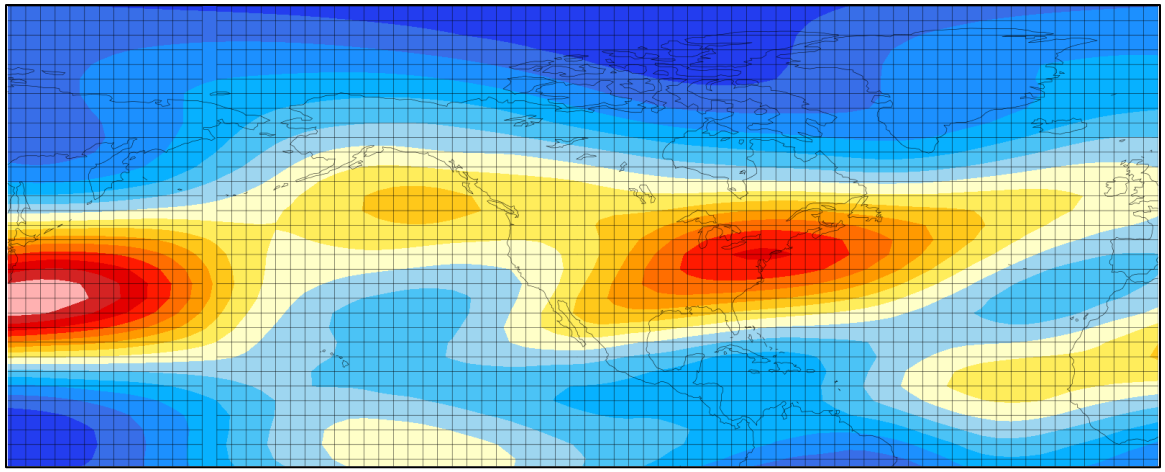
Look for "contour" categories:

- Contours: no map
- Contour effects
- Contour labels
- Labelbars

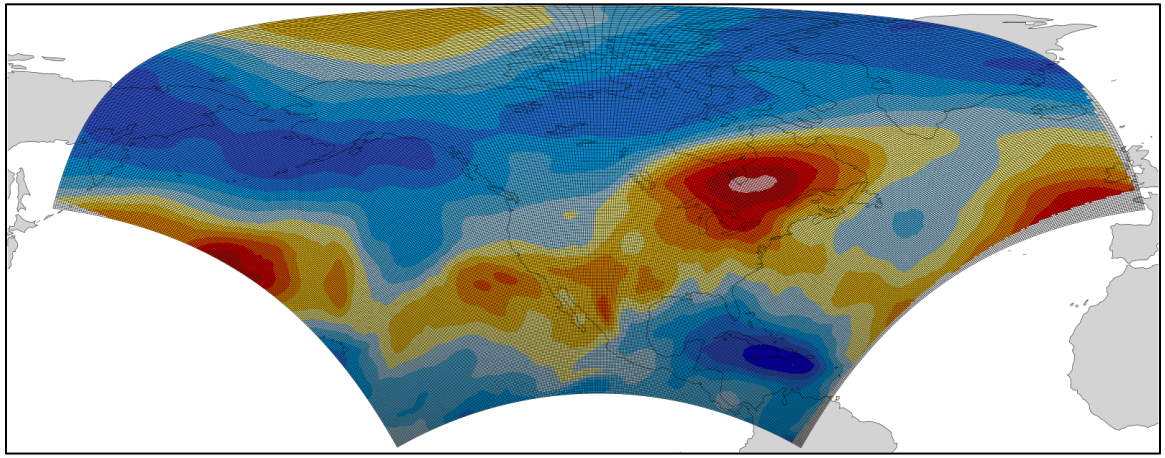
Advanced topics

- Changing the labeling style of labelbars: http://www.ncl.ucar.edu/Applications/labelbar.shtml#ex14
- Setting contour levels to get "white in the middle": http://www.ncl.ucar.edu/Applications/color.shtml#ex15
- Controlling individual contours with shading (patterns): http://www.ncl.ucar.edu/Applications/overlay.shtml#ex5
- Controlling individual contour lines with color and/or thickness: http://www.ncl.ucar.edu/Applications/conOncon.shtml#ex7

Types of lat/lon grids

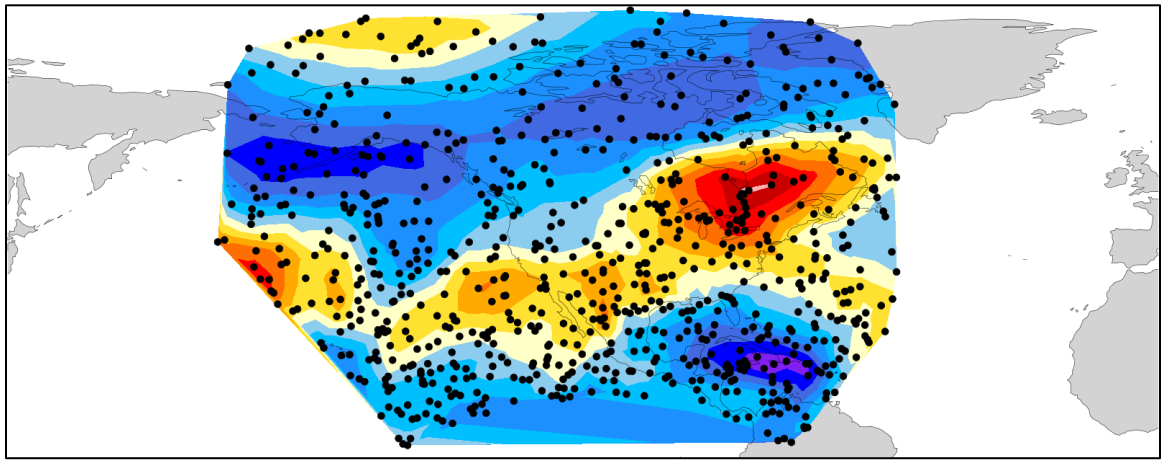


Rectilinear
(1D coordinate arrays)



Curvilinear
(2D lat/lon coordinates)

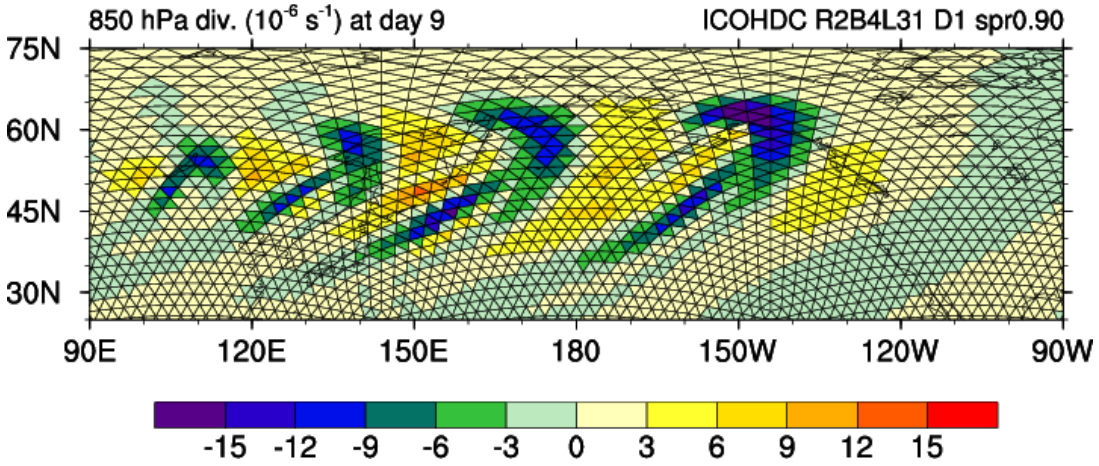
WRF-ARW, POP, NARR, are examples of curvilinear data

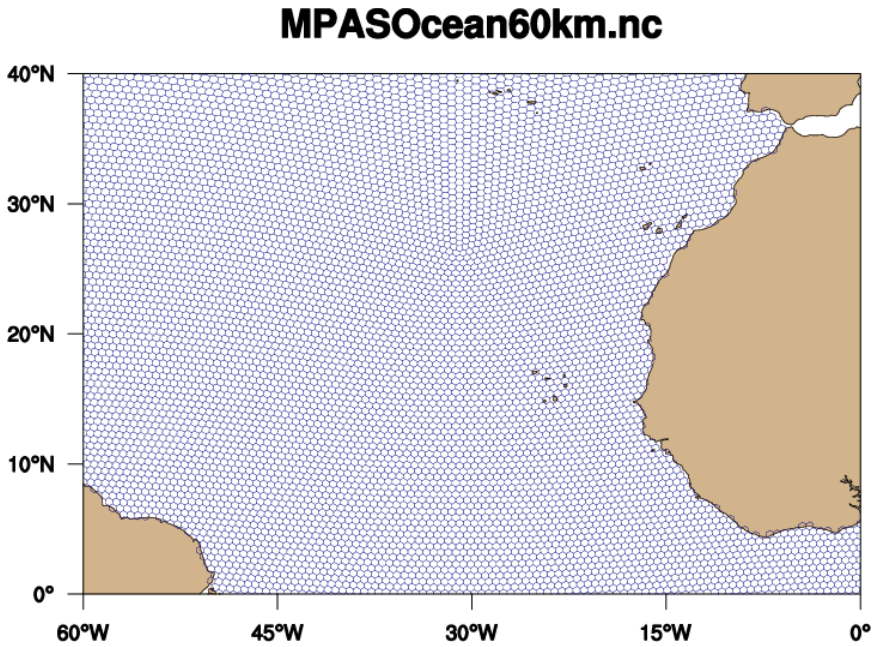


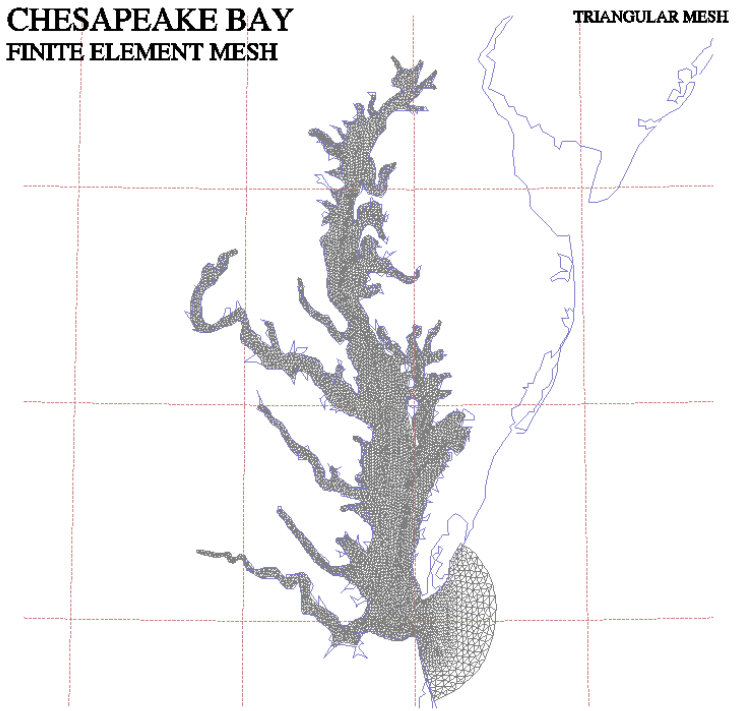
Unstructured
1D lat/lon coordinates

MPAS is an example of unstructured data

More examples of unstructured grids







Special topic on contouring data

- One-dimensional (1D) coordinate arrays (rectilinear)
  - gsn_csm_xxxx scripts automatically look for coordinate arrays
- 2D lon/lat arrays (curvilinear, WRF files, satellite data)
  - Use sfXArray, sfYArray resources
    - res@sfXArray = lon
    - res@sfYArray = lat

Use this method for WRF data

- Unstructured coordinates (1D X,Y,Z arrays)
  - Also use sfXArray, sfYArray resources
    - res@sfXArray = x
    - res@sfYArray = y

Might need:
res@gsnAddCyclic = False

Example: 2D lat/lon arrays

- Assume file is from sea ice model: "iceh_mavg.0014-02.nc"
- Has a variable "hi" w/no coordinate arrays

```
Dimensions and sizes:   [lat | 384] x [lon | 320]
Coordinates:
Number Of Attributes: 7
  time :         4804
  units :        m
  long_name :    grid box mean ice thickness
  coordinates : i j time
  _FillValue :   1e+30
  time_rep :     averaged
```

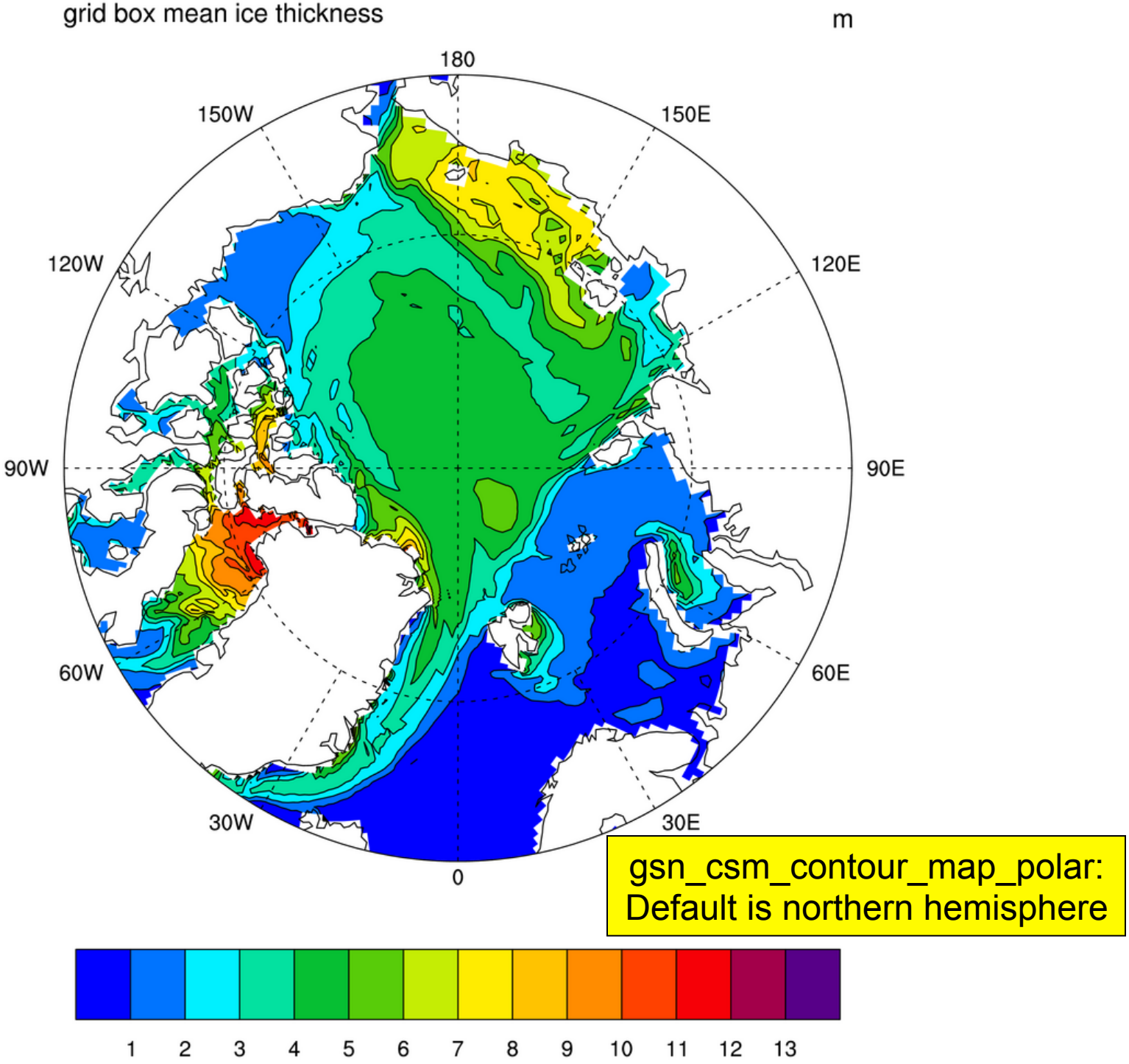
- File does have two-dimensional lat/lon variables

```
float TLON ( lat, lon )
          long_name :     grid center longitude
          units :         degrees_east
float TLAT ( lat, lon )
          long_name :     grid center latitude
          units :         degrees_north
```

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

begin
  f = addfile("iceh_mavg.0014-02.nc","r")
  hi          = f->hi(0,:,:)                ; Ice coverage
  printVarSummary(hi)                       ; Note no coord arrays

  wks = gsn_open_wks("ps","ice")                ; ice.ps
  gsn_define_colormap(wks,"BlAqGrYeOrReVi200")

  res                   = True                ; Plot mods desired
  res@sfXArray          = f->TLON  ; 2D lat/lon arrays, must
  res@sfYArray          = f->TLAT  ; be same dimensions as "hi"

  res@cnFillOn          = True                ; Turn on color fill
  res@mpMinLatF         = 65                  ; Specify min lat

  plot = gsn_csm_contour_map_polar(wks,hi,res)
end
```

Setting sfX/YArray is equivalent to something you may have seen before:
Use this method if you need to add a cyclic point! (`res@gsnAddCyclic = True`)

```
hi@lat2d = f->TLAT
hi@lon2d = f->TLON
```



gsn_csm_contour_map_polar:
Default is northern hemisphere

Example: 1D x,y,z data

- Assume file is from ARPEGE grid
- Has a variable "SUTOPRSU" with no coordinate arrays
- Variable has a time dimension and a degenerate dimension
- Separate file contains 1D lat, lon data

Example: 1D x,y,z data (cont'd)

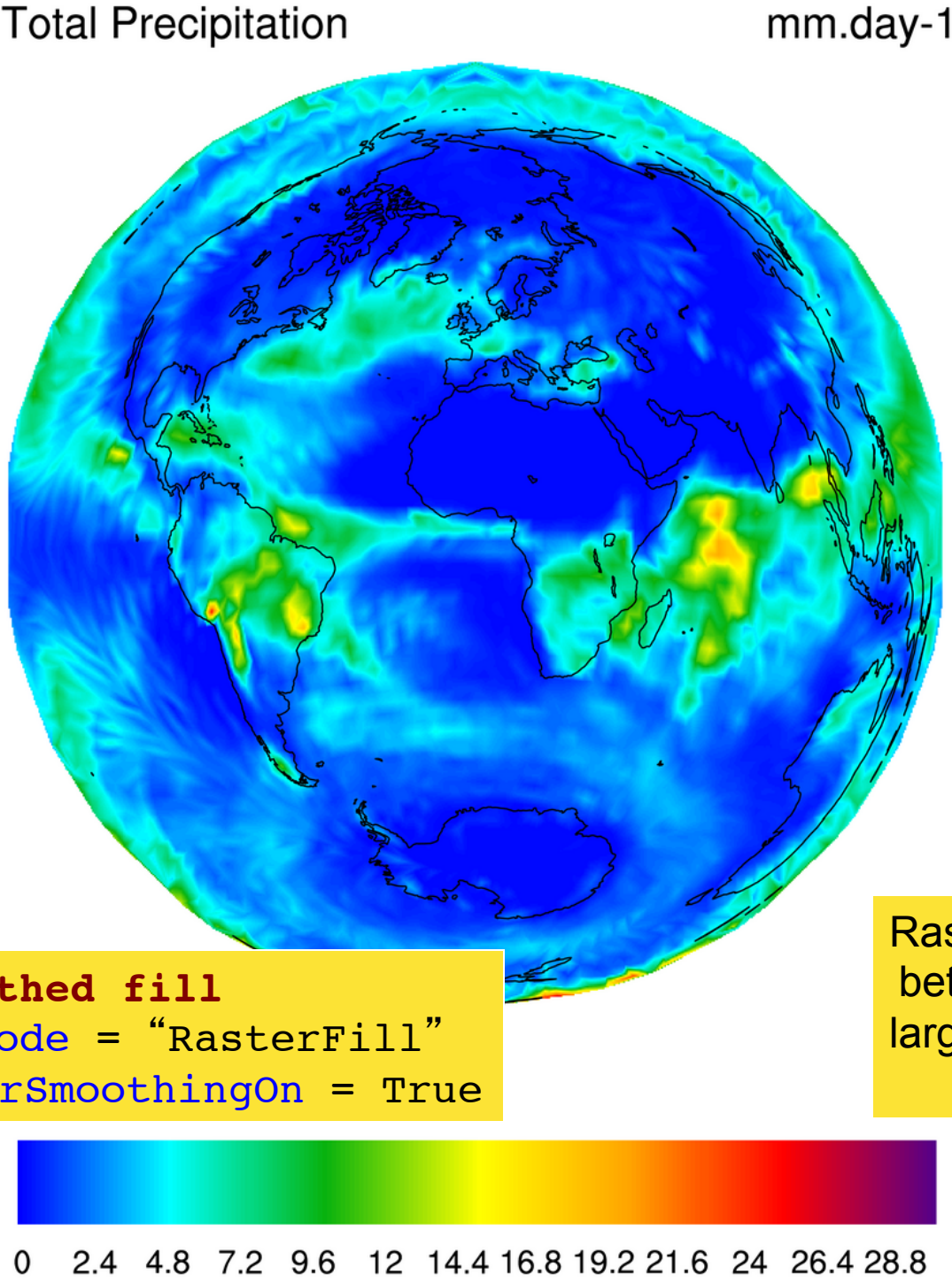
```
Variable: SUTOPRSU (file variable)
Type: float
Number of Dimensions: 3
Dimensions and sizes:   [time_counter | 240] x [jpjb |
1] x [jpib | 6232]
Coordinates:
            time_counter: [   0.. 239]
Number Of Attributes: 4
  short_name :  PRECI TOTALE
  gridtype :    BT42REDU
  long_name :   Total Precipitation
  units :       mm.day-1
```

Variable: bt42_lat (file variable)
Type: double
Number of Dimensions: 2
Dimensions and sizes: [jpjf | 1] x [jpif | 6232]

Variable: bt42_lon (file variable)
Type: double
Number of Dimensions: 2
Dimensions and sizes: [jpjf | 1] x [jpif | 6232]

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

begin
  g = addfile("arpege_grd.nc","r")
  f = addfile("IBF_1m_000101_002012_SUTOPRSU.nc","r")

  wks = gsn_open_wks("x11","arpege")
  gsn_define_colormap(wks,"BlAqGrYeOrReVi200")

  res                           = True
  res@sfXArray                  = g->bt42_lon(0,:)  ; 1D arrays
  res@sfYArray                  = g->bt42_lat(0,:)
  res@cnFillOn                  = True              ; Turn on contour fill
  res@cnFillMode                = "RasterFill"      ; Smooth raster contours
  res@cnRasterSmoothingOn       = True
  res@cnLinesOn                 = False             ; Turn off contour lines

  res@cnLevelSelectionMode  = "ExplicitLevels"    ; Set contour levels.
  res@cnLevels              =  fspan(0,30,201)

  res@lbBoxLinesOn          = False      ; Turn off box lines

  res@mpProjection          = "LambertEqualArea"  ; Map projection

  contour = gsn_csm_contour_map(wks,f->SUTOPRSU(0,0,:),res)
end
```

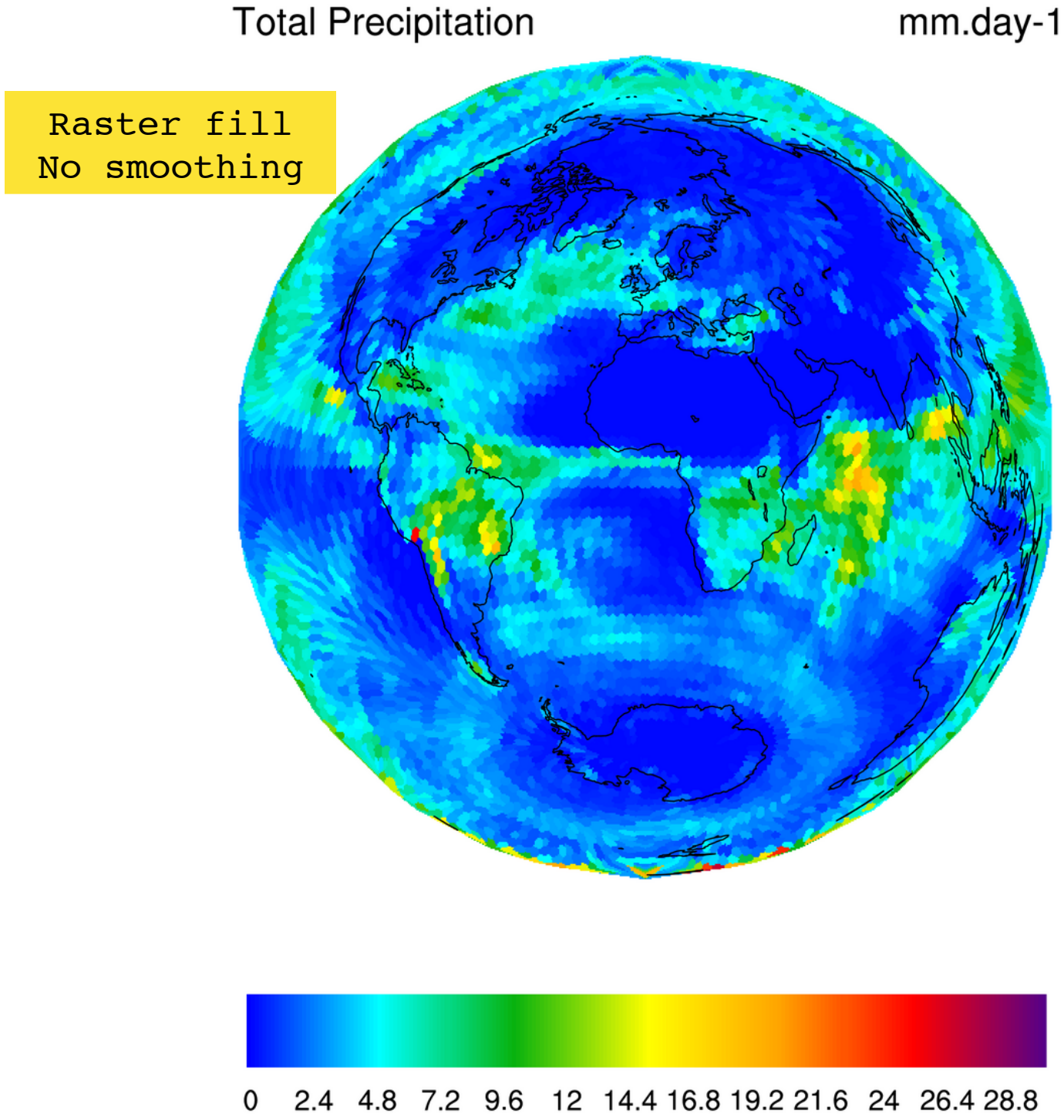
"RasterFill" mode can be *much* faster.



Raster smoothed fill

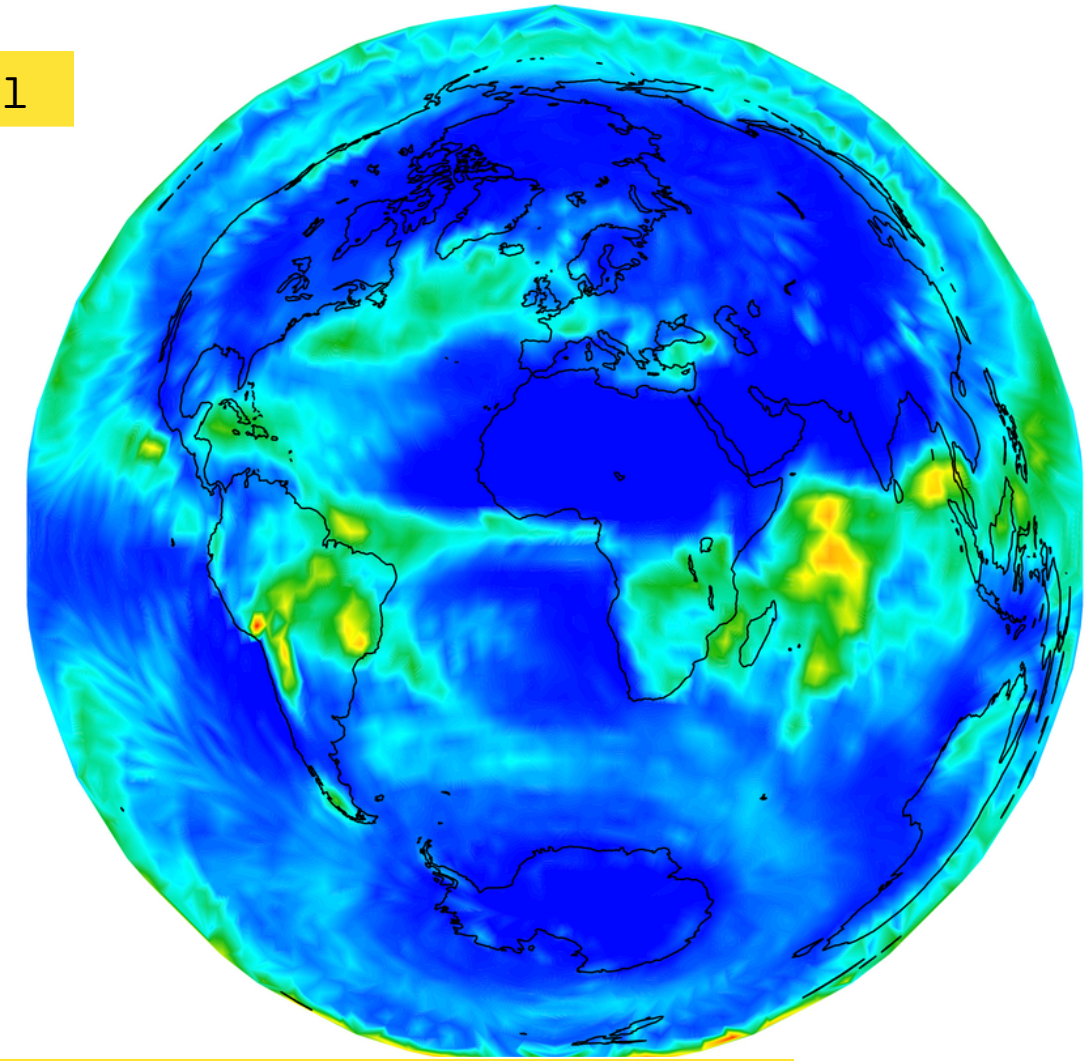
```
res@cnFillMode = "RasterFill"
res@cnRasterSmoothingOn = True
```

Raster contours look better if you have a large grid and/or lots of contours


Total Precipitation
mm.day-1
Raster fill
No smoothing
0 2.4 4.8 7.2 9.6 12 14.4 16.8 19.2 21.6 24 26.4 28.8

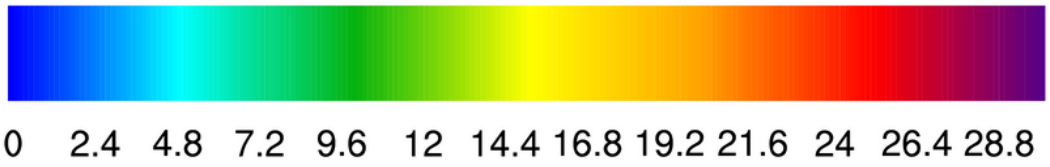
Total Precipitation mm.day-1

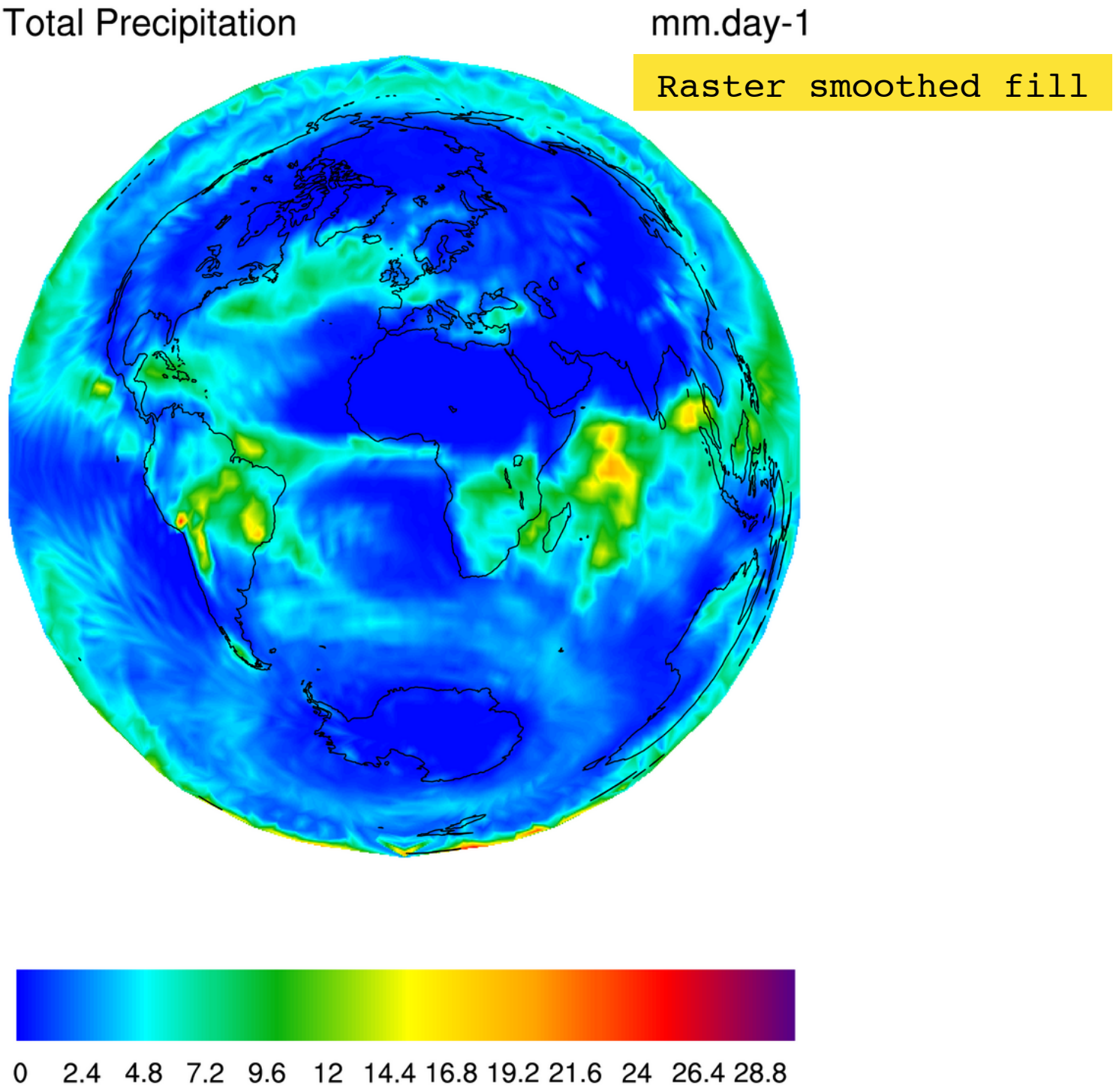
Area fill



Timings

```
Area fill:    268.458u 2.392s 4:35.23
Raster fill:    1.494u 0.152s 0:01.75
```



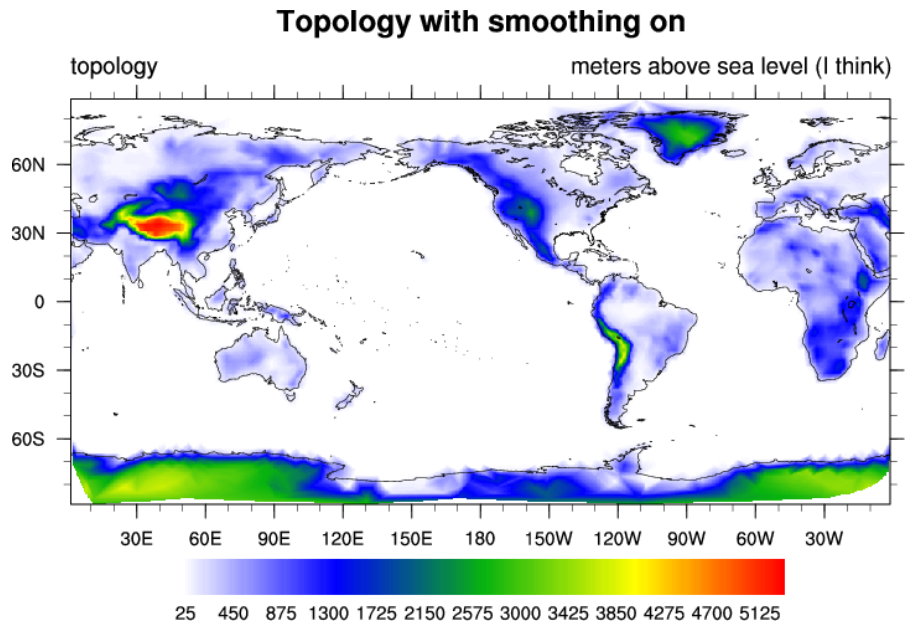

Total Precipitation
mm.day-1
Raster smoothed fill
0 2.4 4.8 7.2 9.6 12 14.4 16.8 19.2 21.6 24 26.4 28.8

Examples of contouring non-rectilinear data

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

Click on “non-uniform grids/random data

- Adaptive grids
- ORCA grids
- Triangular meshes
- etc.

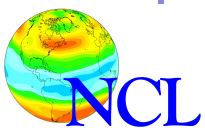


Sample “.hluresfile”

```
! Color map
*wkColorMap                    : rainbow

*Font                          : times-roman

! Function code [Default is a ~]
*TextFuncCode                  : :

! Set size of x11 window
*wkWidth                       : 1000
*wkHeight                      : 1000
```

Common mistakes or problems

- "cnLineColour" is not a resource in ContourPlot at this time"
  - Misspelling a resource, "cnLineColour"
  - Using the wrong resource with the wrong plot (i.e. using "vcRefMagnitudeF" in a contour plot).
- "The units attribute of the Y coordinate array is not set to one of the allowable units values (i.e. 'degrees_north'). Your latitude labels may not be correct."
  - Lack of (or wrong) "units" attribute attached to your data's coordinate arrays



More common mistakes or problems

- Data values in plot look off-scale
  - Maybe "_FillValue" attribute not set or not correct.
- "_NhlCreateSplineCoordApprox: Attempt to create spline approximation for Y axis failed: consider adjusting trYTensionF value"
  - Data is too irregularly spaced in the X or Y direction. May need to subset it.

Debugging tips

- Use interactive mode to try "quick" things that you don't understand in NCL.
- Otherwise, use "batch" script mode. That is, write a script and then run it:

```
ncl myscript.ncl
```

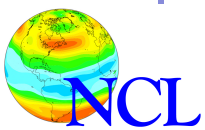
- Start with an existing script, if possible. You can use templates provided at:

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

- Start small, don't set a bunch of resources all at once
- Group resources by type

More debugging tips

- If graphics look wrong, comment out a bunch of resources and add them back slowly to see where problem is
- Use "printVarSummary" to examine variables
  - Missing coordinate arrays
  - No "_FillValue" or wrong "_FillValue"
- To further examine data, use:
  - print(min(x)) and print(max(x)) ; Minimum/maximum of data
  - printMinMax(x,0)
  - print(num(ismissing(x))) ; Count number of msg vals
- Read errors and warnings carefully
- Use an enhanced UNIX editor! ☺



Questions?