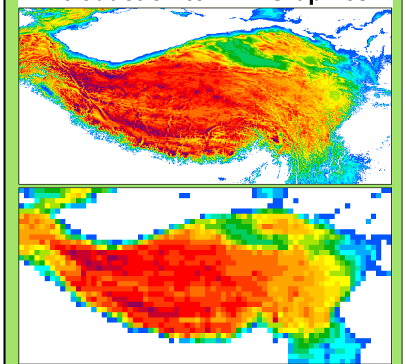
Introduction to NCL Graphics



Mary Haley

Sponsored by the National Science Foundation

Topics

- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the 5 steps
- XY plot demo
- Contour/map demo
- Tips & common mistakes
- Other special topics
- Python

Quick notes

- Questions welcome…but raise hand, get my attention, and speak slowly and clearly! If that doesn't work, gesture wildly and I'll come over.
- Link to post-workshop survey will be emailed.
- Slides may have been added or changed.
- ***I may skip over some slides. I left them in so you could read them on your own.***

Important note

Some of you are running NCL version **6.1.0-beta**.

This is a new version that represents a **MAJOR** overhaul of our graphics.

We changed some defaults in our graphical output, which means you may see differences if you are using NCL **V6.0.0** or earlier.

I will try to mention these differences throughout the lecture.

Differences b/w V6.0.0 and V6.1.0

V6.0.0	V6.1.0
Default font is times-roman	Default font is helvetica
Default color table has 32 colors	Default color table has 256 colors
Default function code is ":"	Default function code is "~"
gsnSpreadColors is False	gsnSpreadColors is True
lbLabelAutoStride is False	lbLabelAutoStride is True
Limited to 256 colors	Colors are "infinite"
Named colors have to first be added to colormap	Named colors do not have to be added to colormap
You can't have partially transparent colors	You can use transparent colors

Goals for this lecture

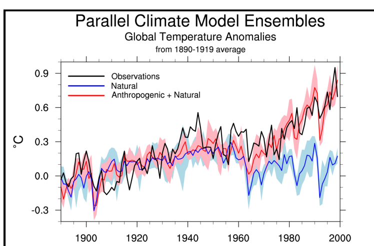
- Familiarize you with the structure of an NCL graphics script
- Get you started with understanding plot "resources"
- Show you the most common things users do with NCL graphics
- Show you debugging tips and common user mistakes
- Tips on creating images for Web and PowerPoint
- How to customize your NCL environment
- Provide you with useful documentation links

Topics

- o Quick notes & goals for this lecture
- o Tour of NCL visualizations
- o NCL Graphics – the 5 steps
- o XY plot demo
- o Contour/map demo
- o Tips
- o Other special topics
- o Python

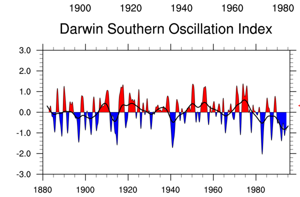
Quick tour of graphics interfaces

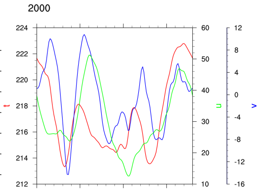
- Over 40 plotting interfaces
- Some highly specialized (bar charts, skew-T, wind roses, histograms, taylor diagrams)
- Special scripts using CERFACS examples:
  http://www.ncl.ucar.edu/Training/Workshops/CERFACS/
- Hundreds of examples:
  http://www.ncl.ucar.edu/Applications/
- Graphical functions documentation:
  http://www.ncl.ucar.edu/Document/Graphics/Interfaces/



XY Plots

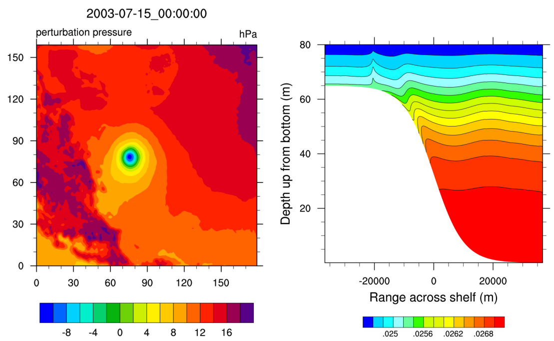
- Can fill between curves
- Can have different labels on axes
- Can have more than one axis





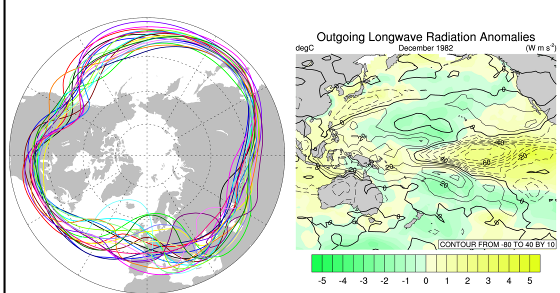
Contour plots (no maps)

We call the color bar at the bottom a “label bar”



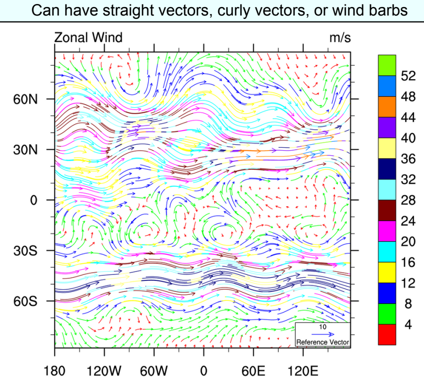
Contour over maps

- Can overlay multiple contour plots (line and filled)
- Can have dashed lines, colored lines, thick lines, etc.



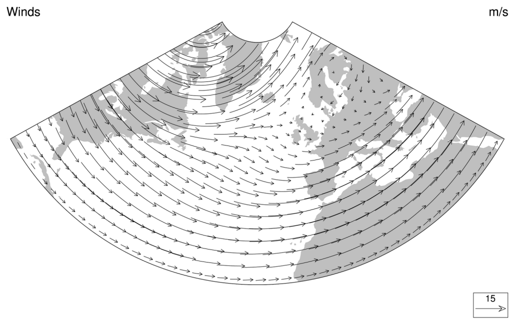
Vector plots

Can have straight vectors, curly vectors, or wind barbs



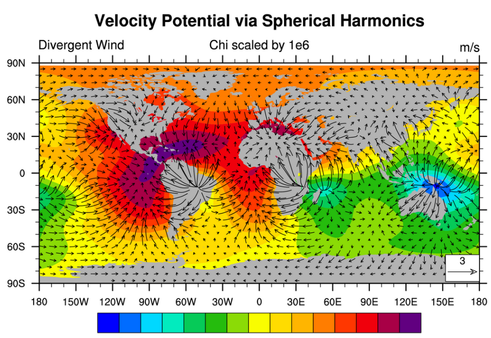
Vectors over maps

This is a "masked" lambert conformal plot



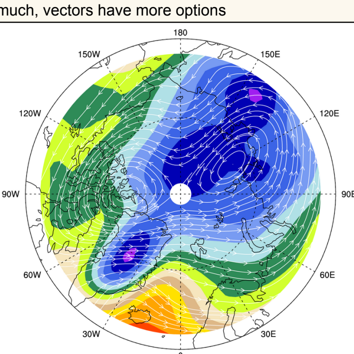
Vectors over maps

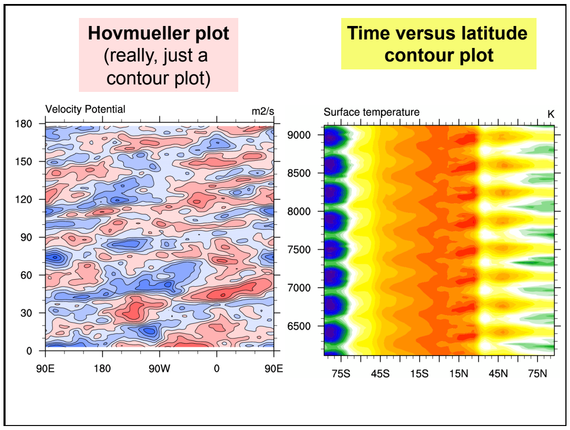
Can "overlay" vectors and contours on same plot

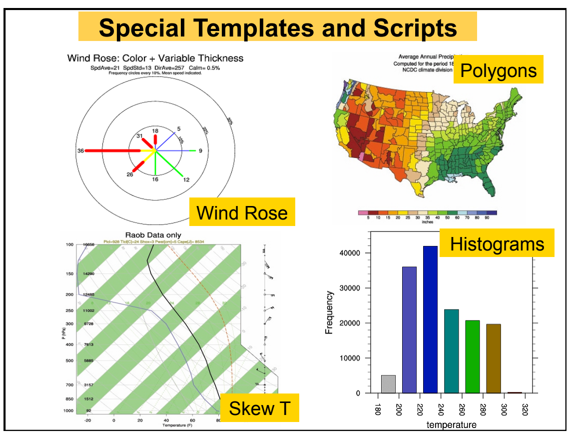


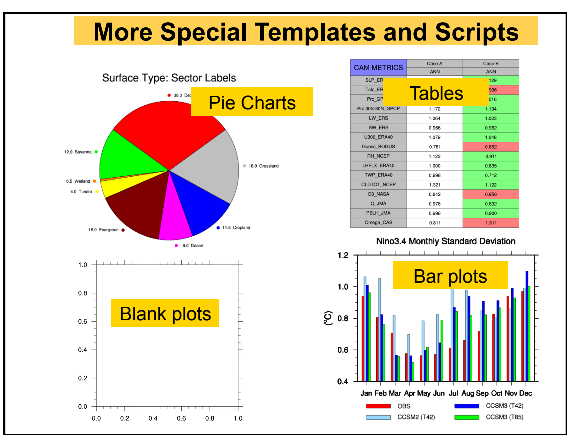
Streamline Plots

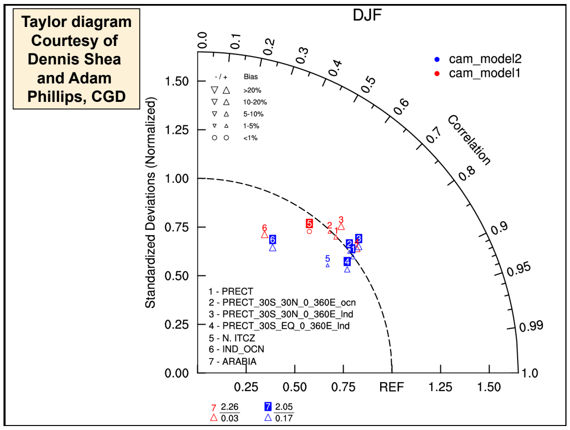
- Streamlines are "continuous" vectors
- Not used much, vectors have more options




Hovmueller plot
(really, just a
contour plot)
Time versus latitude
contour plot
Velocity Potential
m2/s
180
150
120
90
60
30
0
90E
180
90W
0
90E
Surface temperature
K
9000
8500
8000
7500
7000
6500
75S
45S
15S
15N
45N
75N


Special Templates and Scripts
Wind Rose: Color + Variable Thickness
Polygons
Wind Rose
Raob Data only
Histograms
40000
30000
20000
10000
0
Frequency
180
200
220
240
260
280
300
320
temperature
Skew T


More Special Templates and Scripts
Surface Type: Sector Labels
Pie Charts
CAM METRICS
Tables
Nino3.4 Monthly Standard Deviation
Bar plots
Blank plots
1.2
1.0
0.8
0.6
0.4
(°C)
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
OBS
CCSM2 (T42)
CCSM3 (T42)
CCSM3 (T85)



What new (graphical) things you can do in V6.1.0

- Transparency
- Read in existing images and overlay NCL graphics on top
- Use named colors without adding them to your color map
- Use more than one color map (color table) per frame
- Use more than 256 colors per frame

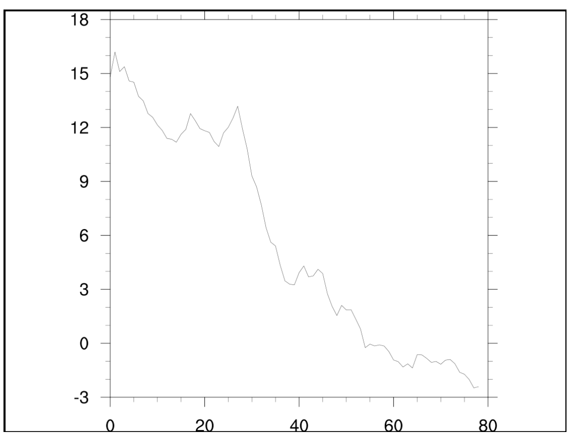
Topics

- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the 5 steps
- XY plot demo
- Contour/map demo
- Tips
- Other special topics
- Python

Metadata conventions recognized by gsn_csm scripts

- _FillValue attribute recognized as a missing value ("missing_value" is NOT)
- Data attributes such as "long_name" and "units" may be used for plot titles
- Coordinate variables used for axes values
- If data has 1D coordinate arrays and you are plotting over a map, then "units" attribute of "degrees_east" or "degrees_north" expected


```
; Step 1. Load the necessary NCL scripts
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f        = addfile("PLPNM50-1.75.M2006.nc", "r")
  tsur     = f->tsur              ; (time, y, x ) (12 x 79 x 117)
  tsur_avg = dim_avg(tsur)        ; calculate average

; Step 2. Open a PNG file
  wks = gsn_open_wks("png", "ALADIN_tsur_avg_1")

; Step 3. Set a color map (not doing that here)
; gsn_define_colormap(wks, "rainbow")

; Step 4. Create a resource list (not doing anything with it yet)
  res = True

; Step 5. Call the plot function
  plot = gsn_csm_y(wks,tsur_avg(0,:), res)
end
```



NCL Graphics 5 steps – the details

1. Load the necessary NCL scripts
2. Open a workstation (where to send graphics)
3. Set a color map (optional)
4. Create a resource list (most crucial step)
5. Call the plot function (XY, contour, vector, etc)

Step 1: Load necessary scripts

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

- Order is important!
- Best to put at very top of script
- Can load other scripts, including your own in the same manner:

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


Step 2: Open graphics "workstation"

- Can be:
  - PostScript ("ps") ("eps" - only one image)
  - PDF ("pdf")
  - X11 window ("x11" – good for debugging)
  - PNG ("png")
  - NCGM ("ncgm") – rarely used
- Has a default color map associated with it (which was changed in V6.1.0-beta!)
- A "frame" means a "page"

Step 2: Open graphics "workstation"

Some samples:

```
wks = gsn_open_wks("x11","test") ; X11 window - good
                                 ; for debugging!
wks = gsn_open_wks("ps","test")  ; "test.ps"
wks = gsn_open_wks("png","wrf")  ; "wrf.png"
wks = gsn_open_wks("pdf","slp")  ; "slp.pdf"
wks = gsn_open_wks("eps","cn")   ; "cn.eps"
```

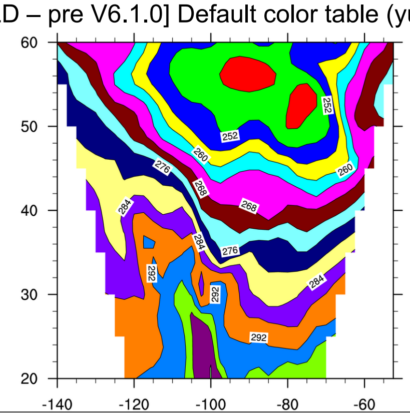

Step 3: Change the color map

- This step is optional
- Do this before drawing any graphics.

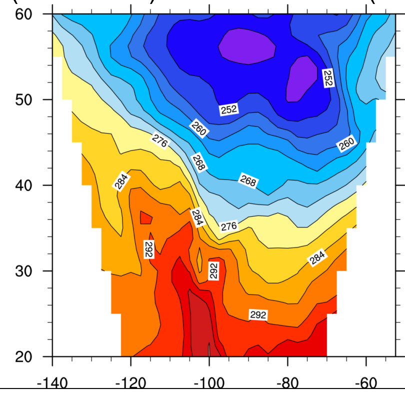
```
gsn_define_colormap(wks,"rainbow")
```

- If you use the same color map a lot, can put in ".hluresfile" file (more later)
- Can use one of the other 90+ color maps, or create your own.
- If you don't change the color map, here's what you'll get…

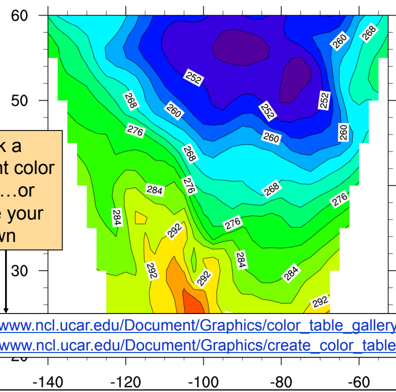
[OLD – pre V6.1.0] Default color table (yuck)



New (V6.1.0-beta) default color table (better?)



“rainbow”

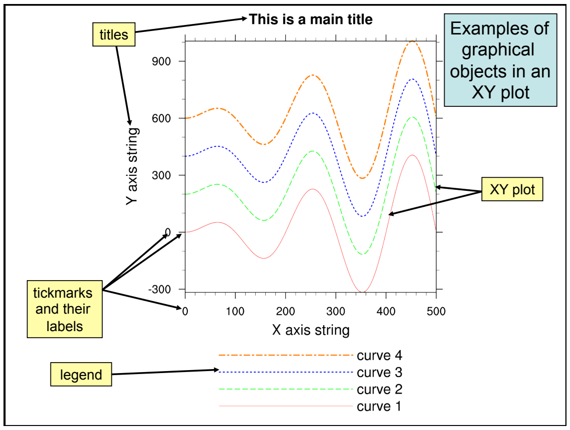


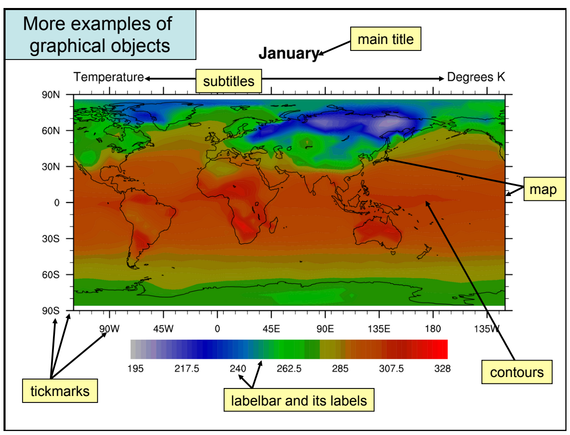
Pick a different color table…or create your own

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

Step 4: Set optional resources

- Resources are the heart of your NCL graphics code.
- There are over 1,400 resources!
- Resources are grouped by object type.
- There are 11 “graphical” objects: contours, labelbars, legends, maps, primitives, streamlines, text strings, tickmarks, titles, vectors, XY plots
- Most common resources listed in the very back of your book





How a resource is constructed

- Starts with 2 or 3 lower-case letters based on object it is associated with. Some examples:
  - "xy" - XY Plot "cn"- Contour plot "mp" - Map
  - "vc" - Vector plot "ti" – Title "gsn" - Special
  - "tm" - Tickmark "lb" - Labelbar
- Made up of full words; first letter capitalized:
  - xyLineColor cnFillOn tiMainString
  - vcRefMagnitudeF gsnMaximize
- Some have an "**F**" on the end to indicate a floating point resource: "xyLineThickness**F**"
- "gsn" – special resources recognized by gsn scripts

How a resource is constructed (cont'd)

- Resources are set by attaching them as attributes to an NCL *logical* variable:

```
res = True      ; can name it whatever you want
res@mpMinLatF = 30    ; decimal not necessary
```

- Most have default values.
- There are many types:

```
res@tiMainString  = "This is a title"
res@tmXBLabelFontHeightF = 0.01
res@cnLineLabelsOn = True
res@xyLineColors = (/5,7,11/)
res@xyLineColors = (/"red", "green", "blue"/)
res@lgLineThicknesses = (/ 1.0, 2.0, 3/)
```


How a resource is constructed (cont'd)

- Resources across objects are similarly named for easier recollection:
  - xyLineColor, cnLineColor, gsLineColor, mpGridLineColor, tmBorderLineColor
  - tiMainFontHeightF, tmXBLabelFontHeightF, lbLabelFontHeightF, cnLineLabelFontHeightF
  - xyDashPattern, mpPerimLineDashPattern, lbBoxLineDashPattern, cnLineDashPattern
  - and so on…

Step 5: Draw the graphics

- Call one of the gsn_csm_xxxxx functions from the second library we loaded.
- Some examples:

```
xy   = gsn_csm_xy(wks,x,y,res)
plot = gsn_csm_contour(wks,data,res)
plot = gsn_csm_contour_map(wks,data,res)
plot = gsn_csm_vector(wks,u,v,res)
map  = gsn_csm_vector_map(wks,u,v,res)
phgt = gsn_csm_pres_hgt(wks,data,res)
```

http://www.ncl.ucar.edu/Document/Graphics/Interfaces/

Topics

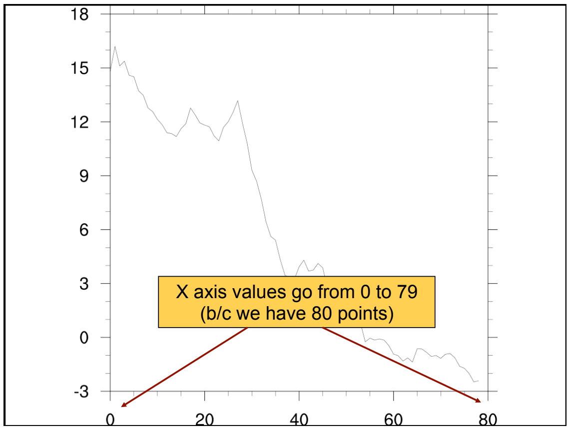
- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the basics
- XY plot demo
- Contour/map demo
- Tips
- Other special topics
- Python

XY plot demo

- Examples of plotting ALADIN data
- Data file provided by Pierre Nabat, Météo-France
- Full scripts can be found on this page:

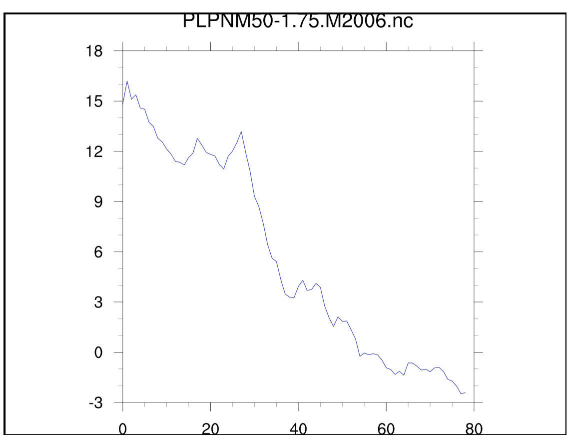
http://www.ncl.ucar.edu/Training/Workshops/CERFACS/Scripts/ALADIN/


```
; Step 1. Load the necessary NCL scripts
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f        = addfile("PLPNM50-1.75.M2006.nc", "r")
  tsur     = f->tsur              ; (time, y, x ) (12 x 79 x 117)
  tsur_avg = dim_avg(tsur)        ; calculate average

; Step 2. Open a PNG file
  wks = gsn_open_wks("png", "ALADIN_tsur_avg_1")

; Step 3. Set a color map (not doing that here)
; gsn_define_colormap(wks, "rainbow")

; Step 4. Create a resource list (not doing anything with it yet)
  res = True

; Step 5. Call the plot function
  plot = gsn_csm_y(wks,tsur_avg(0,:), res)
end
```



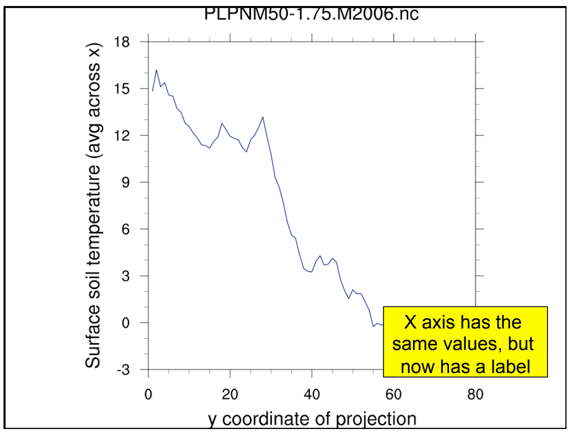

```
; Step 1. Load the necessary NCL scripts
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f        = addfile("PLPNM50-1.75.M2006.nc", "r")
  tsur     = f->tsur               ; (time, y, x ) (12 x 79 x 117)
  tsur_avg = dim_avg(tsur)         ; calculate average

; Step 2. Open a PNG file
  wks = gsn_open_wks("png", "ALADIN_tsur_avg_2")

; Step 4. Create a resource list
  res                    = True
  res@tiMainString       = filename    ; add a main title
  res@xyLineColor        = "NavyBlue"  ; change line color
  res@xyLineThicknessF   = 2.5         ; default is 1

; Step 5. Call the plot function
  plot = gsn_csm_y(wks,tsur_avg(0,:), res)
End
```

Order of resources not important!

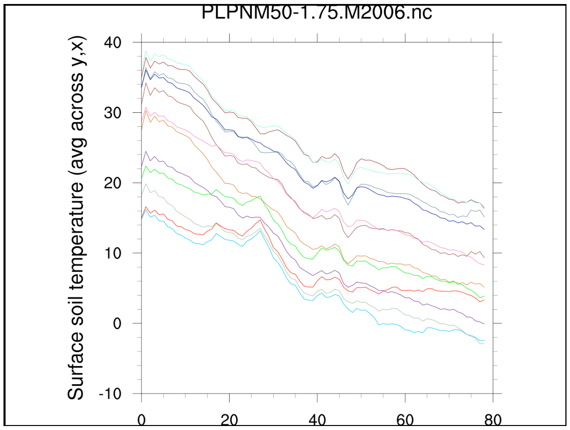


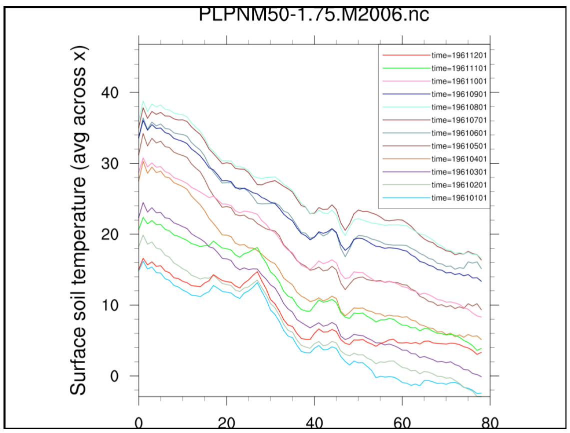
```
; Step 1. Load the necessary NCL scripts
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f        = addfile("PLPNM50-1.75.M2006.nc", "r")
  tsur     = f->tsur             ; (time, y, x ) (12 x 79 x 117)
  tsur_avg = dim_avg_Wrap(tsur)  ; calculate average (retain metadata)

; Step 2. Open a PNG file
  wks = gsn_open_wks("png", "ALADIN_tsur_avg_2")

; Step 4. Create a resource list
  res = True
  res@tiMainString      = filename    ; add a main title
  res@tiYAxisString     = tsur_avg@long_name + " (avg across x)"
  res@xyLineColor       = "NavyBlue"  ; change line color
  res@xyLineThicknessF = 2.5          ; default is 1

; Step 5. Call the plot function
  plot = gsn_csm_xy(wks,tsur&y,tsur_avg(0,:), res)
end
```

Can set this variable to False to quickly turn off all plot options.




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

begin
  f        = addfile("PLPNM50-1.75.M2006.nc", "r")
  tsur     = f->tsur             ; (time, y, x ) (12 x 79 x 117)
  tsur_avg = dim_avg_Wrap(tsur)  ; calculate average (retain metadata)

; Open a PNG file
  wks = gsn_open_wks("png", "ALADIN_tsur_avg_5")

; Create a resource list
  res = True
  res@tiMainString      = filename    ; add a main title
  res@tiYAxisString     = tsur_avg@long_name + " (avg across x)"
  res@xyLineColor       = "NavyBlue"  ; change line color
  res@xyLineThicknessF = 2.5          ; default is 1
  res@xyMonoDashPattern = True        ; Use one pattern (solid)
  res@xyLineColors = (/"DeepSkyBlue", "DarkSeaGreen","DarkOrchid4",\
                     "Chocolate","coral4","cadetblue4","brown4",\
                     "aquamarine","blue4","hotpink","green","red"/)
; Call the plot function
  plot = gsn_csm_xy(wks,tsur&y,tsur_avg, res) ; 12 curves x 79 points
end
```





Advanced scripts for XY plots

- More examples of the previous scripts:
  http://www.ncl.ucar.edu/Training/Workshops/CERFACS/ALADIN/
- Two ways to add data to an existing XY plot
  http://www.ncl.ucar.edu/Applications/xy.shtml#ex25
  http://www.ncl.ucar.edu/Applications/xy.shtml#ex26
- Filling the area between two curves
  http://www.ncl.ucar.edu/Applications/xy.shtml#ex24
- Turning XY curves into individual bars
  http://www.ncl.ucar.edu/Applications/bar.shtml
- Changing an axis (log, irregular, linear)
  http://www.ncl.ucar.edu/Applications/axes.shtml#ex3

Very basic XY plot exercises

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

Click on:

- Basic graphical exercises
- XY Plot Exercises (set 1)
- XY Plot Exercises (set 2)

More complex (real world) examples:
http://www.ncl.ucar.edu/Applications/xy.shtml

Topics

- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the basics
- XY plot demo
- Contour/map demo
- Tips
- Other special topics
- Python

Contour/map plot demo

- Examples of plotting MOCAGE data
- Data provided by Andrea Piacentini, CERFACS
- Data is GLOBAL
- More scripts can be found on this page:

http://www.ncl.ucar.edu/Training/Workshops/CERFACS/Scripts/MOCAGE/

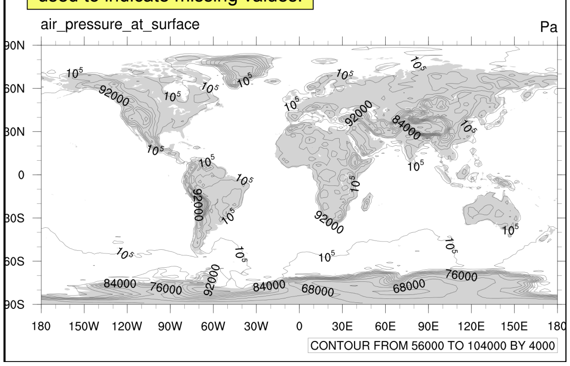
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  filename = "DXGLOB22+2003070106+000.nc"
  f = addfile (filename,"r")

;---Read data
  apas = f->air_pressure_at_surface ; time x lat x lon
  printVarSummary(apas)     ; Use for debugging!

;---Start the graphics
  wks = gsn_open_wks("png" ,"MOCAGE_DX_apas_1")

;---Set some resources
  res = True

  nt = 0              ; time index
  plot = gsn_csm_contour_map(wks,apas(nt,:,:),res)
end
```

Output from "printVarSummary" call

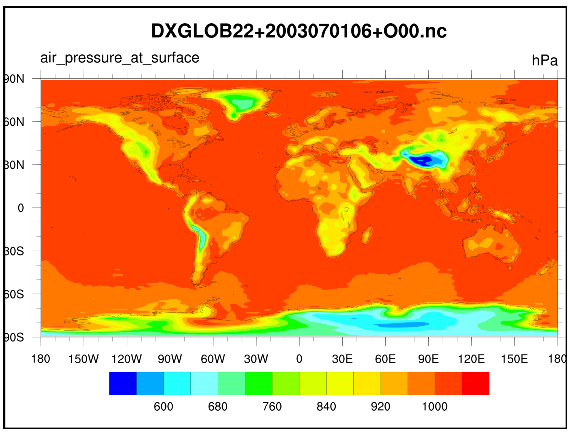
```
Variable: apas
Type: float
Total Size: 64800 bytes
            16200 values
Number of Dimensions: 3
Dimensions and sizes:      [time | 1] x [lat | 90] x [lon | 180]
Coordinates:
            time: [21600..21600]
            lat: [-89..89]
            lon: [-179..179]
Number Of Attributes: 2
  standard_name :    air_pressure_at_surface
  units :     Pa
```

We have lat/lon coordinate arrays

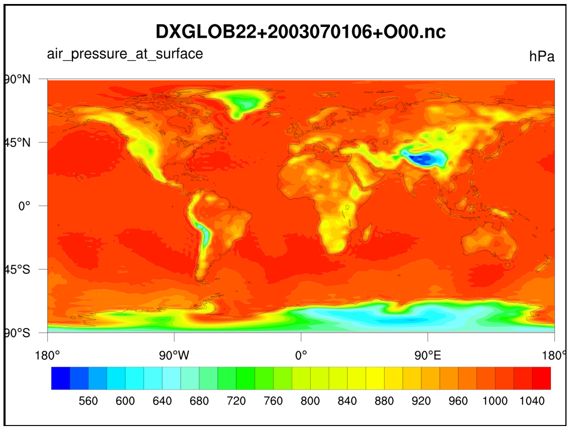
"_FillValue" attribute is being used to indicate missing values.



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

begin
  filename = "DXGLOB22+2003070106+O00.nc”
  f = addfile (filename,"r”)

;---Read data
  apas = f->air_pressure_at_surface ; time x lat x lon
  apas       = apas * 0.01                        ; Pa to hPa
  apas@units = "hPa"

;---Start the graphics
  wks = gsn_open_wks("png" ,"MOCAGE_DX_apas_2”)
  gsn_define_colormap(wks,"BlAqGrYeOrRe")    ; Change color map

;---Set some resources
  res                 = True
  res@gsnMaximize  = True
  res@cnFillOn       = True          ; Turn on contour fill
  res@cnLinesOn      = False         ; Turn off contour lines
  res@tiMainString = filename      ; Use filename as title

  nt = 0             ; time index
  plot = gsn_csm_contour_map(wks,apas(nt,:,:),res)
end
```




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

begin
  filename = "DXGLOB22+2003070106+O00.nc”
  f = addfile (filename,"r”)

;---Read data
  apas = f->air_pressure_at_surface ; time x lat x lon
  apas       = apas * 0.01                        ; Pa to hPa
  apas@units = "hPa"

;---Start the graphics
  wks = gsn_open_wks("png" ,"MOCAGE_DX_apas_3”)
  gsn_define_colormap(wks,"BlAqGrYeOrRe")    ; Change color map

;---Set some resources
  res                 = True
  res@gsnMaximize  = True
  res@cnFillOn       = True          ; Turn on contour fill
  res@cnLinesOn      = False         ; Turn off contour lines
  res@tiMainString = filename      ; Use filename as title
  res@cnLevelSpacingF = 20 ; Default was 40

  res@tiMainOffsetYF         = -0.04      ; Move closer to plot
  res@pmLabelBarWidthF       = 0.8        ; Make labelbar longer
  res@pmTickMarkDisplayMode = "Always"   ; Nicer map tickmarks
  nt = 0             ; time index
  plot = gsn_csm_contour_map(wks,apas(nt,:,:),res)
end
```



Another contour/map plot demo

- Examples of plotting SAFRAN data
- Data provided by Clotilde Dubois, Météo-France
- Data is REGIONAL (France)
- More scripts can be found on this page:

http://www.ncl.ucar.edu/Training/Workshops/CERFACS/Scripts/SAFRAN/

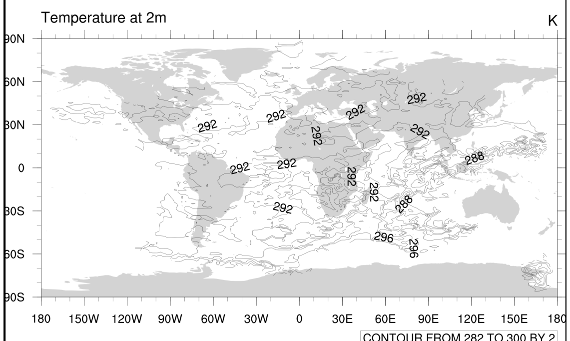
Introduction to NCL Graphics

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

Begin
;---Open file and read data
  f = addfile ("ForcT.DAT_france_0001.nc", "r")
  t = f->T              ; (time, y, x) (8760 x 134 x 143)
  printVarSummary(t)

;---Open workstation
  wks = gsn_open_wks("png","SAFRAN_temperature_1")

;---Set some resources
  res = True
  res@tiMainString = "Note: this plot is incorrect"

;---Draw the plot
  plot = gsn_csm_contour_map(wks,t(0,:,:),res)
end
```

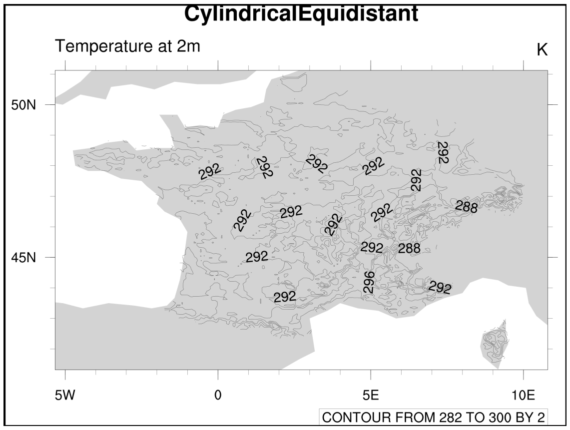
```
Variable: t
Type: float
Total Size: 671436480 bytes
            167859120 values
Number of Dimensions: 3
Dimensions and sizes:      [time | 8760] x [y | 134] x [x | 143]
Coordinates:
            time: [881688..0]
            y: [1617000..2681000]
            x: [60000..1196000]
Number Of Attributes: 6
  long_name : Temperature at 2m
  units :     K
  grid_mapping :    Lambert_Conformal
  coordinates :     lon lat
  missing_value :   -9999
  _FillValue :      -9999
```

We have NO lat/lon coordinate arrays

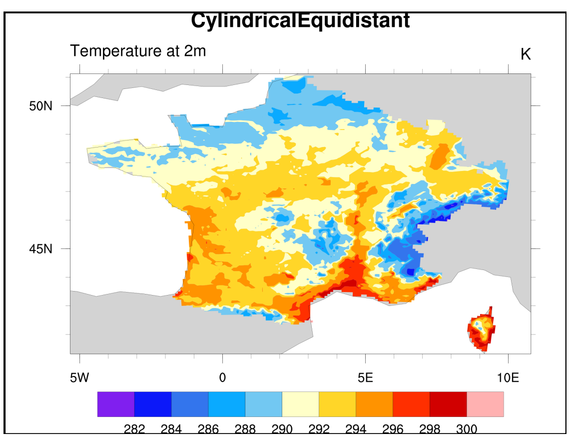
Note: this plot is incorrect




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

begin
  f = addfile ("ForcT.DAT_france_0001.nc", "r")
  t = f->T ; (time, y, x) (8760 x 134 x 143)

  wks = gsn_open_wks("png","SAFRAN_temperature_3")

  res = True
  res@tiMainString = "CylindricalEquidistant"

;---This will position data correctly on map.
  res@sfXArray = lon2d
  res@sfYArray = lat2d
  res@gsnAddCyclic = False ; Data not global, don't add lon cyclic pt

;---Zoom in on map
  res@mpMinLatF = min(lat2d)
  res@mpMaxLatF = max(lat2d)
  res@mpMinLonF = min(lon2d)
  res@mpMaxLonF = max(lon2d)

;---Draw the plot
  plot = gsn_csm_contour_map(wks,t(0,:,:),res)
end
```

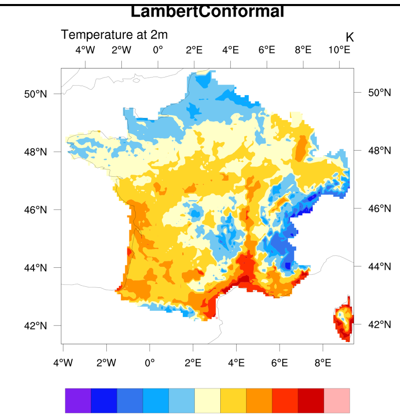



```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile ("ForcT.DAT_france_0001.nc", "r")
  t = f->T ; (time, y, x) (8760 x 134 x 143)

  wks = gsn_open_wks("png","SAFRAN_temperature_2")

  res = True
  res@tiMainString = ”CylindricalEquidistant”

;---This will position data correctly on map.
  res@sfXArray = lon2d
  res@sfYArray = lat2d
  res@gsnAddCyclic = False ; Data not global, don't add lon cyclic pt

;---Zoom in on map
  res@mpMinLatF = min(lat2d)
  res@mpMaxLatF = max(lat2d)
  res@mpMinLonF = min(lon2d)
  res@mpMaxLonF = max(lon2d)

  res@cnFillOn = True      ; Turn on contour fill
  res@cnLinesOn = False    ; Turn off contour lines

;---Draw the plot
  plot = gsn_csm_contour_map(wks,t(0,:,:),res)
end
```

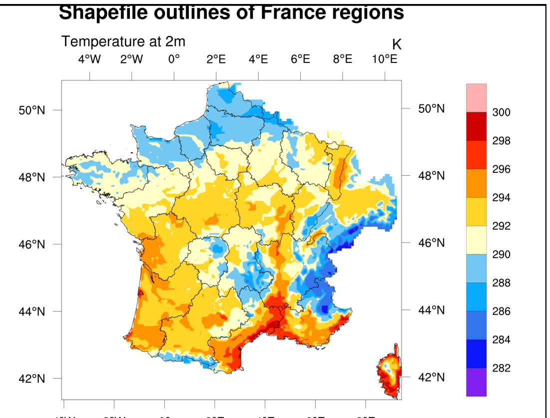


```
. . .
  f    = addfile ("ForcT.DAT_france_0001.nc", "r")
  t    = f->T ; (time, y, x) (8760 x 134 x 143)

  lc   = f->Lambert_Conformal ; contains map projection information
  nlat = dimsizes(lat2d(:,0)) ; Get lat dimension size
  mlon = dimsizes(lon2d(0,:)) ; Get lon dimension size
. . .
;---Use projection information on file
  res@mpProjection         = "LambertConformal"
  res@mpLambertParallel1F = lc@standard_parallel(0)
  res@mpLambertParallel2F = lc@standard_parallel(1)
  res@mpLambertMeridianF  = lc@longitude_of_central_meridian

;---Zoom in on map res@mpLimitMode = "Corners"
  res@mpLeftCornerLatF  = lat2d(0,0)
  res@mpLeftCornerLonF  = lon2d(0,0)
  res@mpRightCornerLatF = lat2d(nlat-1,mlon-1)
  res@mpRightCornerLonF = lon2d(nlat-1,mlon-1)
  res@mpDataBaseVersion = "MediumRes" ; Better map outlines

;---Draw the plot
  plot = gsn_csm_contour_map(wks,t(0,:,:),res)
end
```

LambertConformal



Shapefile outlines of France regions



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

- More examples of the previous scripts:
  http://www.ncl.ucar.edu/Training/Workshops/CERFACS/ALADIN/
- Changing the labeling style of labelbars:
  http://www.ncl.ucar.edu/Applications/labelbar.shtml#ex14
- 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

No vector plot demo

- Be aware of these resources:

```
res@vcRefMagnitudeF = 20    ; Vector magnitude
res@vcRefLengthF    = 0.09  ; Size of reference vector
res@vcMinDistanceF  = 0.02  ; Thins number of arrows

res@vcGlyphStyle = "CurlyVector"   ; "LineVector","WindBarb"
```

Vector plot exercises

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

Click on “vector plot exercises”

More examples: http://www.ncl.ucar.edu/Applications/vector.shtml
http://www.ncl.ucar.edu/Applications/veceff.shtml

In review…

- Five main steps to create a plot
- Use X11 window while debugging script; move to PS/PDF later
- Hardest part are the resources: start simple
- Organize resources for easier debugging
- Start with an existing script if possible

Topics

- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the 5 steps
- XY plot demo
- Contour/map demo
- Tips & common mistakes
- Other special topics
- Python

Debugging tips

- Start small, don't set 50 resources all at once
- Start with an existing script, if possible
- Group resources by type
- Don't share resource lists
- Comment out resources and add back slowly to see where problem is
- Use "printVarSummary" to examine variables
  - Missing coordinate arrays
  - No "_FillValue" or wrong "_FillValue"
- Use
  - print(min(x)) and print(max(x)) ; Minimum/maximum of data
  - print(num(ismissing(x))) ; Count number of msg vals

  to further examine data
- Read errors and warnings carefully

Introduction to NCL Graphics

Common mistakes or problems

http://www.ncl.ucar.edu/Document/Graphics/error_msg.shtml

- ~~Forgot .hluresfile (fonts will look wrong)~~
- "xyLineColour" is not a resource in XyPlot at this time"
  - Misspelling a resource, "xyLineColour"
  - Using the wrong resource with the wrong plot (i.e. using "xyLineColor" 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

Introduction to NCL Graphics

More common mistakes or problems

- Data values in plot look off-scale
  - Maybe "_FillValue" attribute not set or not correct.
- Not getting gray-filled lands in map plots. (Version 6.1.0-beta doesn't have this issue)
  - You are using a color map that doesn't have gray in it (V6.0.0 or earlier: use "NhlNewColor" to add gray or change color maps to one that has gray).
- "_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.

Introduction to NCL Graphics

Topics

- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the 5 steps
- XY plot demo
- Contour/map demo
- Tips
- Other special topics
- Python

Special topics

- Creating paneled plots
- Using function codes
- Customizing NCL graphics environment
- Creating images for PowerPoint, Keynote, Web

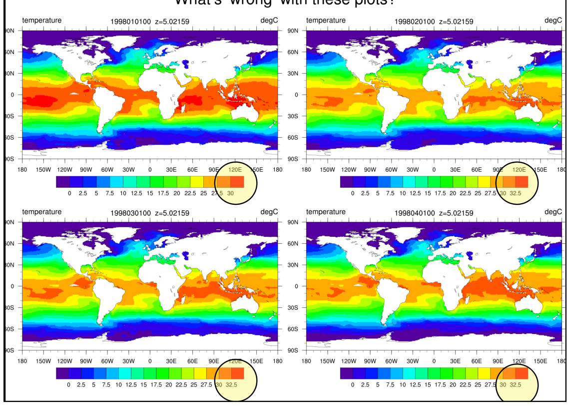
Creating paneled plots

- Using IPSL data from Sophie Valcke, CERFACS
- Plotting first four timesteps
- See scripts at:

http://www.ncl.ucar.edu/Training/Workshops/CERFACS/Scripts/IPSL/

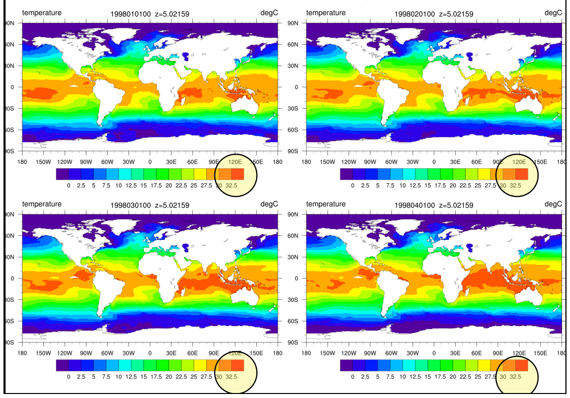
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

begin
  filename = "FRCCORE2_e_1m_19980101_20071231_thetao.nc"
  f        = addfile (filename, "r")
  thetao   = f->thetao              ; (time_counter, deptht, y, x)
  lat2d    = f->nav_lat             ; (y,x)
  lon2d    = f->nav_lon             ; (y,x)
  date     = cd_calendar(thetao&time_counter, -3)   ; YYYYMMDDHH

;---Start the graphics
  wks = gsn_open_wks("png" ,"IPSL_thetao_panel_1")
  gsn_define_colormap(wks,"rainbow")             ; Change color map

 ;---Set some resources
   res                = True
   res@gsnDraw        = False              ; Don't draw plot, b/c will
   res@gsnFrame       = False              ; panel it later.
   res@gsnMaximize    = True               ; Maximize size of plot

   res@sfXArray      = lon2d         ; This will position data
   res@sfYArray      = lat2d         ; correctly on map.
   res@gsnAddCyclic = False          ; Cyclic point already added
   res@cnFillOn      = True          ; Turn on contour fill
   res@cnLinesOn     = False         ; Turn off contour lines
   res@cnFillMode    = "RasterFill"  ; Faster than "AreaFill"

   nd    = 0
   plots = new(4,graphic)
   do nt=0,3     ;---Loop over four timesteps and create four plots
     res@gsnCenterString = date(nt) + "  z=" + thetao&deptht(nd)
     plots(nt) = gsn_csm_contour_map(wks,thetao(nt,nd,:,:),res)
   end do

 ;---Set up resources for paneling
   pres               = True
   pres@gsnMaximize = True                ; Maximize paneled plots
   pres@txString    = "What's 'wrong' with these plots?"

  gsn_panel(wks,plots,(/2,2/),pres)     ; Draw 2 rows x 2 columns
end
```

What's 'wrong' with these plots?

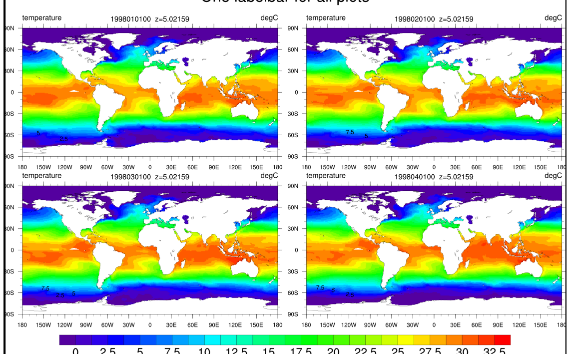



```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

begin
  filename = "FRCCORE2_e_1m_19980101_20071231_thetao.nc"
  f        = addfile (filename, "r")
  thetao   = f->thetao              ; (time_counter, deptht, y, x)
  lat2d    = f->nav_lat             ; (y,x)
  lon2d    = f->nav_lon             ; (y,x)
  date     = cd_calendar(thetao&time_counter, -3)   ; YYYYMMDDHH
. . .

;---Fix the contour levels for all plots
   nd    = 0
   mnmxspac = nice_mnmxintvl(min(thetao(:,nd,:,:)),max(thetao(:,nd,:,:)),16,0)
   res@cnLevelSelectionMode = "ManualLevels" ; Default is "automaticlevels"
   res@cnMinLevelValF       = mnmxspac(0)
   res@cnMaxLevelValF       = mnmxspac(1)
   res@cnLevelSpacingF      = mnmxspac(2)

   plots = new(4,graphic)
   do nt=0,3     ;---Loop over four timesteps and create four plots
     res@gsnCenterString = date(nt) + "  z=" + thetao&deptht(nd)
     plots(nt) = gsn_csm_contour_map(wks,thetao(nt,nd,:,:),res)
   end do

 ;---Set up resources for paneling
   pres               = True
   pres@gsnMaximize = True                ; Maximize paneled plots
   pres@txString    = "Contour levels 'fixed' for all plots"

   gsn_panel(wks,plots,(/2,2/),pres)     ; Draw 2 rows x 2 columns
end
```

Contour levels 'fixed' for all plots



One labelbar for all plots



Problems with paneling?

- Are plots the same size? If not, maybe set res@gsnPanelScalePlotIndex
- Set res@gsnPanelDebug = True
  - Prints debug information about size and location of paneled plots
- Set res@gsnPanelBoxes = True
  - Draws bounding boxes around each plot element so you can see true size
- Set res@gsnPanelXF and/or res@gsnPanelYF
  - Can use these to force plots to line up
- Use vpXF/vpYF/vpWidthF/vpHeightF instead of gsn_panel

Paneling exercises

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

Click on “paneling exercises”

More examples: http://www.ncl.ucar.edu/Applications/panel.shtml

Special topics

- Creating paneled plots
- Using function codes
- Customizing NCL graphics environment
- Creating images for PowerPoint, Keynote, Web

Function Codes

“Superscripts: x~S~2~N~ + y~S~2~N~”

→ Superscripts: $x^2 + y^2$

Can be used for any string in a graphic
tiMainString
txString
xyLineLabel
etc

“Subscripts: CH~B~4~N~ + N~B~2~N~O”

→ Subscripts: $CH_4 + N_2O$

"Carriage~C~Return"

→ Carriage
Return

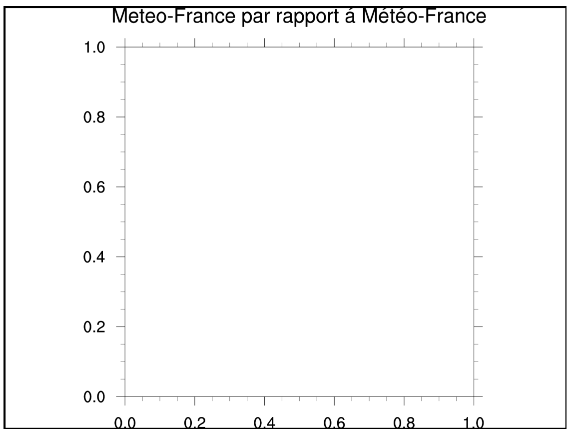
Examples:

```
res@tiMainString = “Happy ~F35~r~F~ Valentine’s Day”
res@tiXAxisString = “m~S~2”
```

”Happy ~F35~r~F~ Valentine’s Day"

→ Happy ♥ Valentine’s Day

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

begin
  eacute  = "e~H-13V2F35~B~FV-2H3~" ; 'B' is a back tick in F35
  aacute  = "a~H-13V2F35~B~FV-2H3~" ; H is for horizontal move;
                                    ; V is for vertical move

  wks = gsn_open_wks("png","text")

  res = True
  res@tiMainString   = "Meteo-France par rapport " + aacute + \
                       " M" + eacute + "t" + eacute + \
                       "o-France"

  plot = gsn_csm_blank_plot(wks,res)
  draw(plot)
  frame(wks)
end
```



Special topics

- Creating paneled plots
- Using function codes
- Customizing NCL graphics environment
- Creating images for PowerPoint, Keynote, Web

Customize your graphics environment

- Optional, but highly recommended if you have V6.0.0 or earlier.
- Download ".hluresfile" file, put in home directory
  - Changes your default background, foreground colors from black/white to white/black
  - Changes font from times-roman to helvetica
  - Changes "function code" (default is a colon)
  - Can be used to change default color map
- May already be available on your lab machines:
  - cat ~/.hluresfile

http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml

Customize your graphics environment

- Optional, but highly recommended if you have V6.0.0 or earlier.
- Download ".hluresfile" file, put in home directory
  - Changes your default background, foreground colors from black/white to white/black
  - Changes font from times-roman to helvetica
  - Changes "function code" (default is a colon)
  - Can be used to change default color map
- May already be available on your lab machines:
  - cat ~/.hluresfile

These are the defaults in V6.1.0

http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml

Sample ".hluresfile"

```
! White background/black foreground
*wkForegroundColor       : (/0.,0.,0./)
*wkBackgroundColor       : (/1.,1.,1./)

! Color map
*wkColorMap              : rainbow+gray

*Font                    : helvetica

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

! Set size of x11 window
*wkWidth                 : 700
*wkHeight                : 700
```

These are the defaults in V6.1.0

Special topics

- Creating paneled plots
- Using function codes
- Customizing NCL graphics environment
- Creating images for PowerPoint, Keynote, Web

Creating images for web or PowerPoint

- Try using direct "png" output:

```
wks = gsn_open_wks("png","example")
```

- If this doesn't produce good results, send output to PS or PDF file

```
wks = gsn_open_wks("pdf","example")
```

- Download "convert", part of free ImageMagick package

  http://www.imagemagick.org/script/index.php

- Mac users can use MacPorts: port install imagemagick
- Linux users: yum install imagemagick
- Use command like:

```
convert -geometry 1000x1000 -density 300 -trim xy.ps
xy.png
```

- The "-density 300" option is what gives you higher-quality images. You can play with this number. For posters, use larger values for both the geometry and density.

Converting images inside NCL script

- Send output to "png" file
- Increase size using wkWidth and wkHeight
- Use NCL's "system" to call "convert" to trim PNG

```
wtype          = "png
wtype@wkWidth  = 2000
wtype@wkHeight = 2000
wks = gsn_open_wks(wtype,"test")
res = True
. . .
plot = gsn_csm_xxxx(wks,data,res)

delete(wks)
system("convert –trim test.png test_trim.png")
```

Converting images inside NCL script

- Send output to "ps" file
- Use "delete(wks)" to force the close of the PS file
- Use NCL's "system" to call "convert"

```
. . .
filename = "test"
psf      = filename + ".ps"    ; PS file name
pngf     = filename + ".png"   ; PNG file name
wks = gsn_open_wks("ps",filename)
res = True
. . .
plot = gsn_csm_xxxx(wks,data,res)
delete(wks)

options = " -geometry 1000x1000 -density 300 -trim "
system("convert" + options + psf + " " + pngf)
```


Topics

- Quick notes & goals for this lecture
- Tour of NCL visualizations
- NCL Graphics – the 5 steps
- XY plot demo
- Contour/map demo
- Tips
- Other special topics
- Python

PyNGL / PyNIO

- PyNIO – Python module to NCL's file I/O
  - Very similar to "addfile"
  - Has some data subsetting capabilities not available in NCL
- PyNGL – Python module to NCL's graphics
  - Same basic 5 steps to create a graphic
  - Same resource names
  - Slightly different syntax

https://www.pyngl.ucar.edu/

https://www.pyngl.ucar.edu/Nio.shtml

Compare PyNGL/PyNIO and NCL/GSUN scripts

PyNGL/PyNIO

```
import Ngl, Nio

#  Open the NetCDF file.
nf = Nio.open_file("mtemp.cdf","r")

# Get lat/lon/temperature variables.
lat = nf.variables["lat"][:]
lon = nf.variables["lon"][:]
T   = nf.variables["t"][0,:,:]

# Open a PS workstation.
wks = Ngl.open_wks("ps","mecca")

# Contour & scalar field resources.
res                = Ngl.Resources()
res.sfXArray       = lon
res.sfYArray       = lat
res.cnFillOn       = True

# Draw contour plot.
contour = Ngl.contour(wks,T,res)

Ngl.end()
```

NCL

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

begin
;  Open the NetCDF file.
  nf = addfile("mtemp.cdf","r")

; Get lat/lon/temperature variables.
  lat  = nf->lat
  lon  = nf->lon
  T    = nf->t(0,:,:)

; Open a PS workstation.
  wks = gsn_open_wks("ps","mecca")

; Contour & scalar field resources.
  res                        = True
  res@sfXArray               = lon
  res@sfYArray               = lat
  res@cnFillOn               = True
  res@pmLabelBarDisplayMode  = "Always"

; Draw contour plot.
  contour = gsn_contour(wks,T,res)

end
```

Questions?

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

Mary Haley

haley@ucar.edu • ncl-talk@ucar.edu