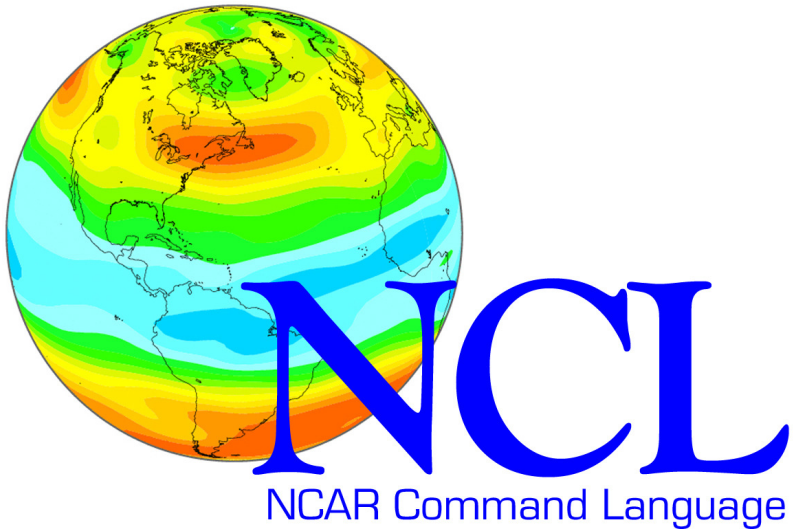
Introduction to NCL Graphics



Part 1 in a series

September 26, 2014

Mary Haley

Sponsored by the National Science Foundation

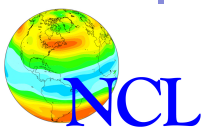
CISL

NESL

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

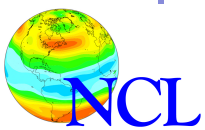
Notes

- First in a series of lectures on NCL Graphics
- Don't know yet how many in the series ☺
- Lectures based on slides used in our NCL Workshops
- The second lecture will focus on contouring
- Geared towards new users of NCL
- Assumptions:
  - You are familiar with basic NCL language features
  - You know about the NCL/NetCDF data model (attributes, coordinate arrays, dimension names, missing values)



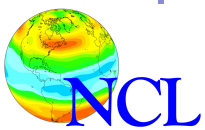
Goals for this series of lectures

- Get you comfortable with the basics of NCL Graphics – you won't be an expert!
- 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

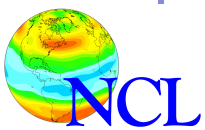


NCL Graphics topics for this lecture

- Types of graphics you can create with NCL
- The basics
- Line-by-line examples of XY plots
- Interactive demo
- Customizing NCL environment (if there's time)

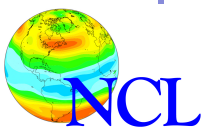


<Suite of gallery slides>



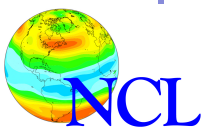
NCL Graphics topics

- Types of graphics you can create with NCL
- The basics
- Line-by-line examples of XY plots
- Interactive demo
- Customizing NCL environment (if there's time)



Four steps to create an NCL graphic

1. Load the "gsn" scripts
2. Open a workstation
3. Set plot options (resources)
4. Call the appropriate plotting function



Step 1: Load the “gsn” scripts *

* Optional in NCL V6.2.x

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

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","test")  ; ‘test.ps’

  res                   = True          ; plot options
  res@xyLineColor   = "Blue"        ; line color
  res@tiMainString = "This is a title"
  res@gsnMaximize   = True
  plot = gsn_csm_y(wks,y,res)     ; no X values
end
```

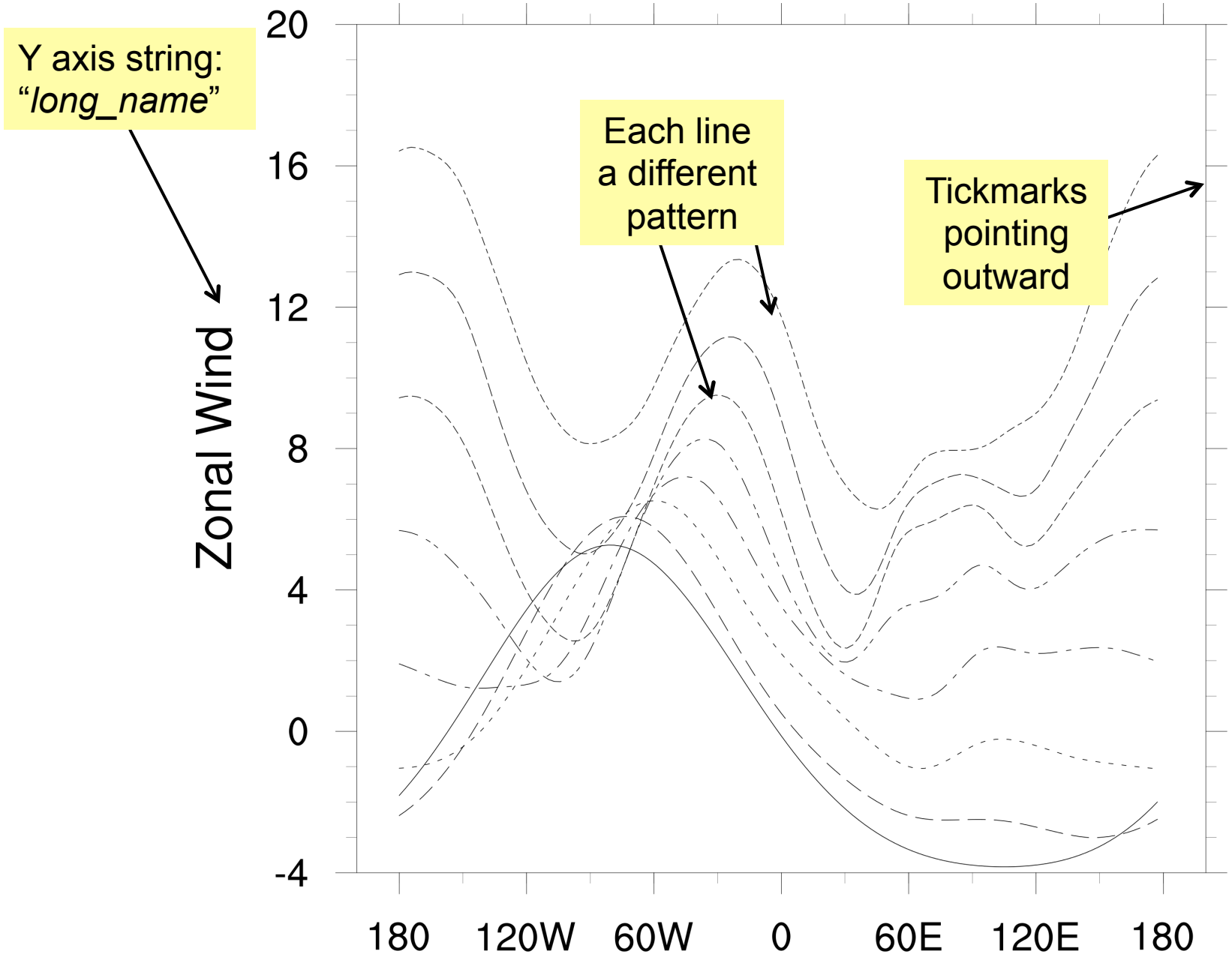
Step 1: Load the "gsn" scripts

- Two ways of doing graphics in NCL:
  1. Using ~~object oriented~~ method
  2. Using high-level graphical interfaces: "gsn" and "gsn_csm" functions
- "gsn" stands for "Getting Started using NCL"
- "csm" stands for "Climate System Model"
- Functions in "gsn_csm.ncl" are more "metadata aware"

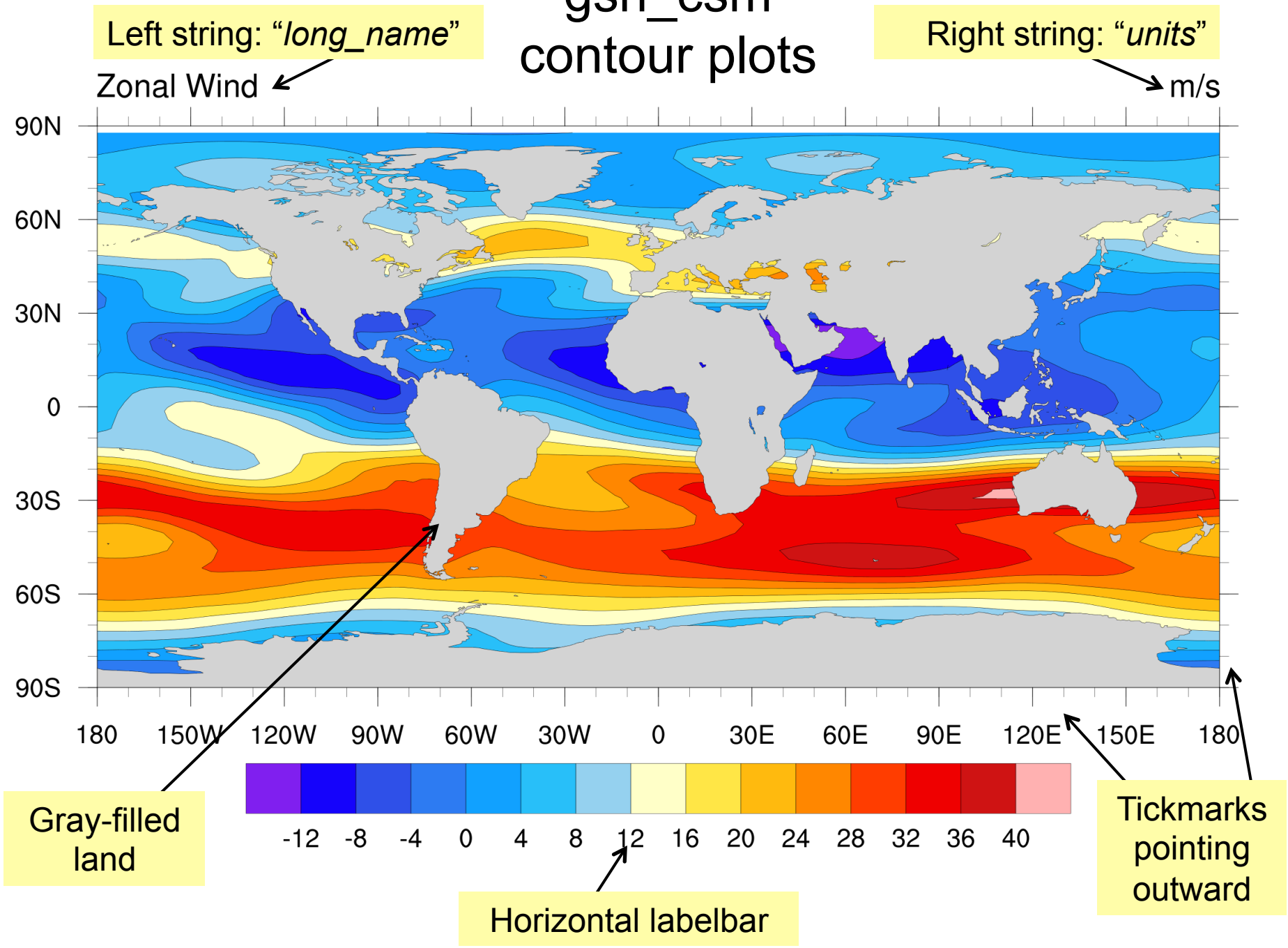
Metadata recognized by gsn_csm scripts

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

“gsn_csm” XY plots



"gsn_csm" contour plots



Step 1.5: Get some data!

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

begin
  y = sin(0.0628*ispan(0,100,1))    ; 101 points

  wks = gsn_open_wks("ps","test")   ; 'test.ps'

  res                   = True          ; plot options
  res@xyLineColor       = "Blue"        ; line color
  res@tiMainString      = "This is a title"
  res@gsnMaximize       = True
  plot = gsn_csm_y(wks,y,res)     ; no X values
end
```

Step 2: Open a workstation

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

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","test")  ; 'test.ps'

  res                  = True          ; plot options
  res@xyLineColor      = "Blue"        ; line color
  res@tiMainString     = "This is a title"
  res@gsnMaximize      = True
  plot = gsn_csm_y(wks,y,res)      ; no X values
end
```

Step 2: Open a workstation

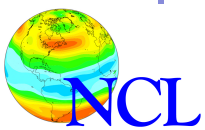
- "Workstation" is where to send the graphics
- Can be:
  - PostScript ("ps") ("eps" - only one image)
  - PDF ("pdf")
  - X11 window ("x11" – good for debugging)
  - PNG ("png")
  - SVG (new in NCL V6.2.0)
  - NCGM ("ncgm") – rarely used
- Has a default color map associated with it

Step 2: Open a workstation

Some samples:

```
wks = gsn_open_wks("x11","test") ; X11 window
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"
```

"gsn_open_wks" is a function. "wks" is a variable name.
You can use whatever name you like for "wks".

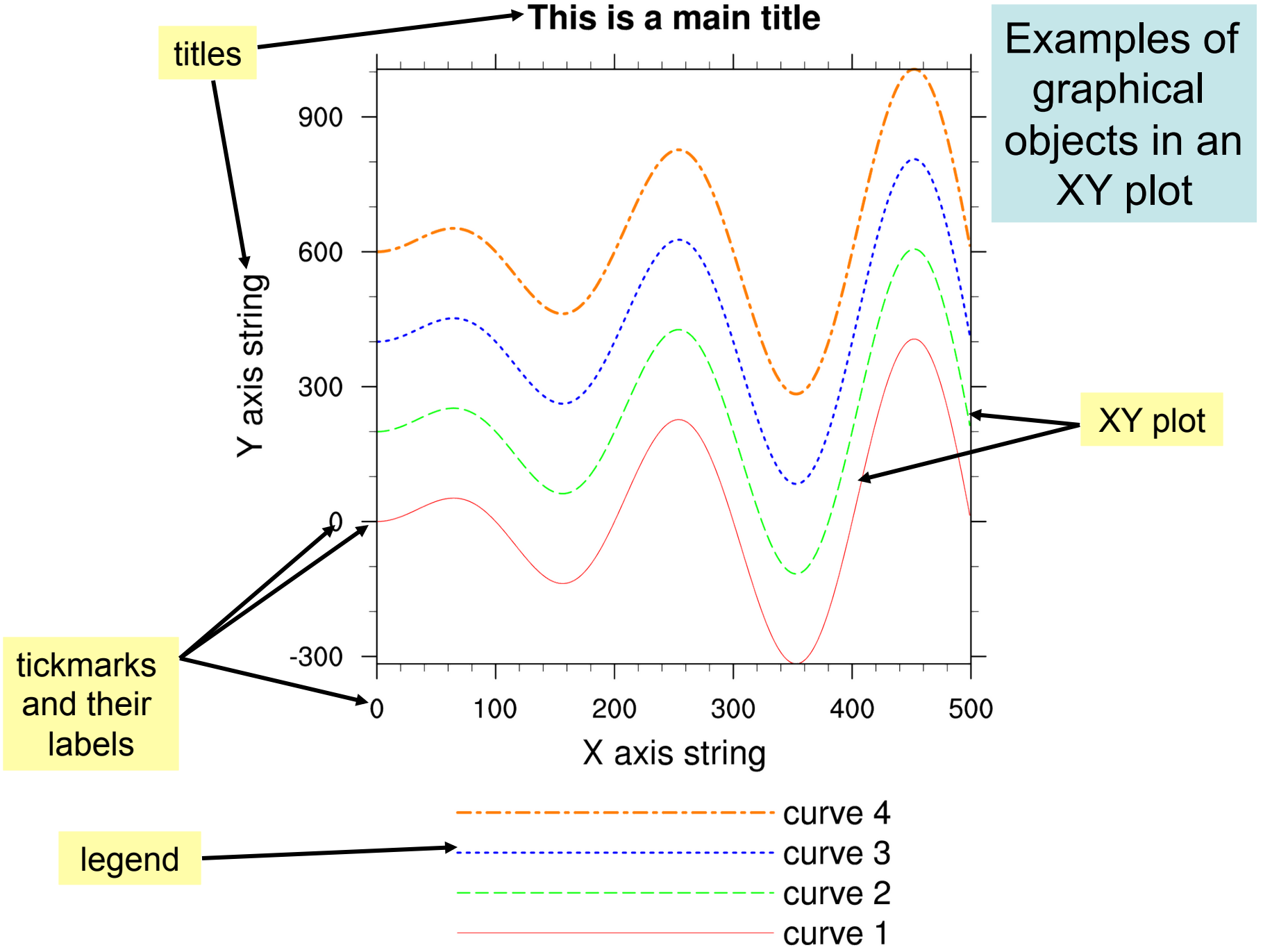


Step 3: Set plot options (resources)

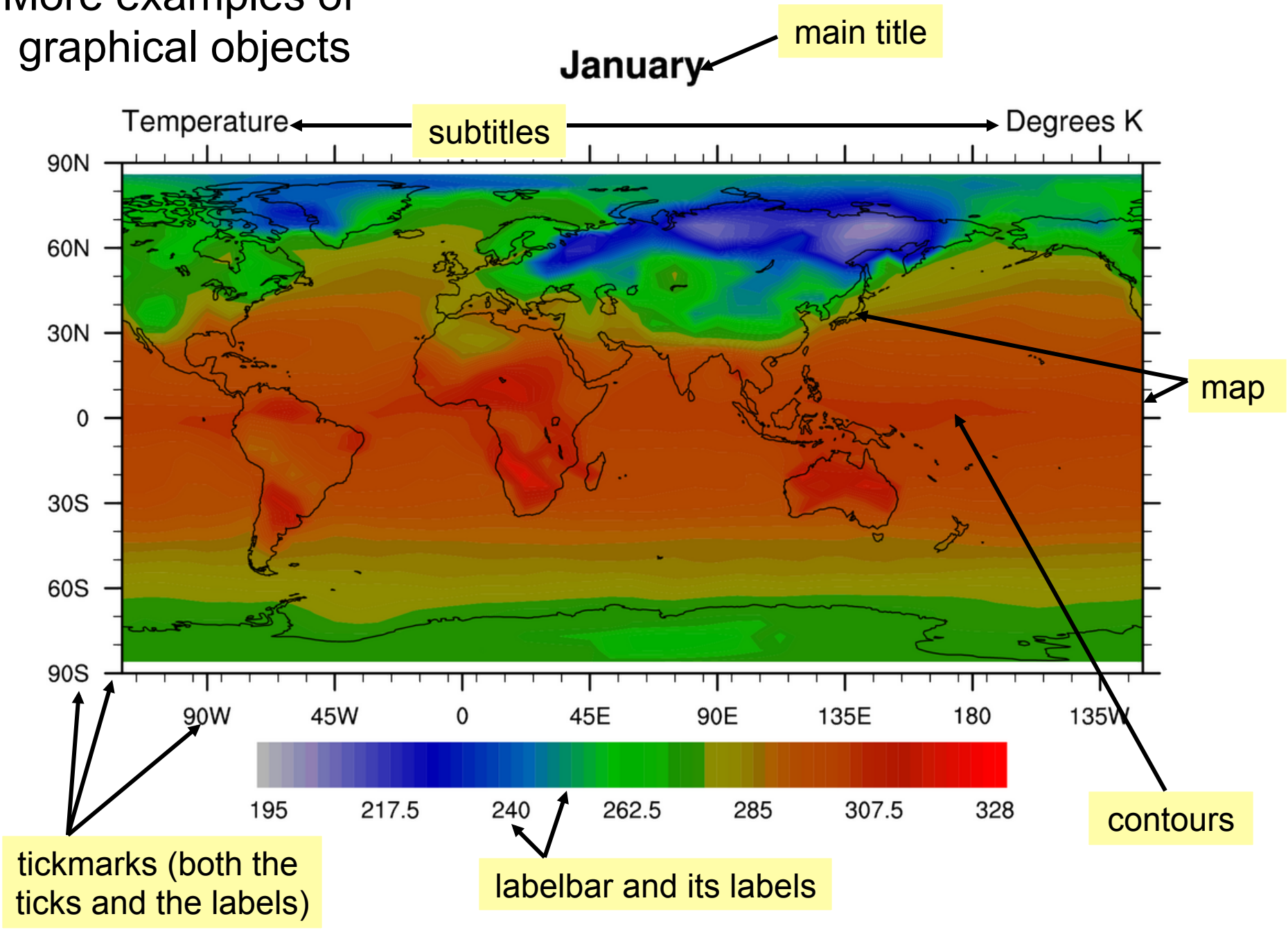
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","test")  ; 'test.ps'

  res                   = True          ; plot options
  res@xyLineColor  = "Blue"       ; line color
  res@tiMainString = "This is a title"
  res@gsnMaximize  = True  ; make plot larger
  plot = gsn_csm_y(wks,y,res)     ; no X values
end
```

Step 3: Set plot options (resources)

- Resources are the heart of your NCL graphics code. There are over 1,400 resources!
- Use resources to change the look of a plot
- They are grouped by object type.
- There are 11 "graphical" objects:
  - contours, labelbars, legends, maps, primitives, streamlines, text strings, tickmarks, titles, vectors, XY plots

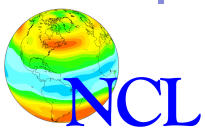

Examples of graphical objects in an XY plot
titles
This is a main title
Y axis string
900
600
300
0
-300
0
100
200
300
400
500
X axis string
XY plot
tickmarks and their labels
legend
curve 4
curve 3
curve 2
curve 1

More examples of graphical objects



Description of resources

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



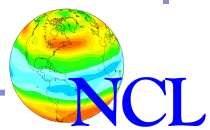
Description of resources (cont'd)

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

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

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

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

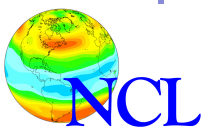



Description of resources (cont'd)

- 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 4: Call appropriate plotting function

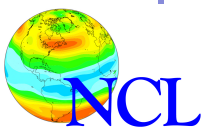
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","test")  ; 'test.ps'

  res                   = True         ; plot options
  res@xyLineColor  = "Blue"       ; line color
  res@tiMainString = "This is a title"
  res@gsnMaximize  = True
  plot = gsn_csm_y(wks,y,res)     ; no X values
end
```

Step 4: Call appropriate plotting function

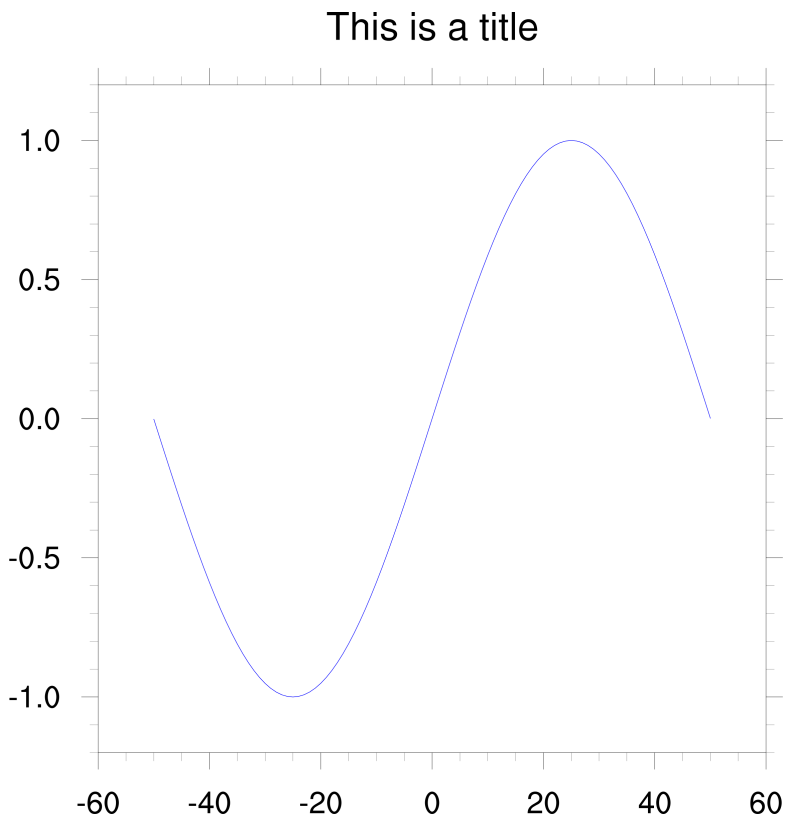
- Call one of the gsn_csm_xxxxx plotting functions.
- Some examples:

```
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)
```

These are all functions. You can use whatever name you like for variable name on left side of "=" sign.

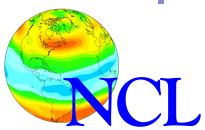


```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","test")  ; 'test.ps'

  res                  = True         ; plot options
  res@xyLineColor      = "Blue"       ; line color
  res@tiMainString     = "This is a title"
  res@gsnMaximize      = True
  plot = gsn_csm_y(wks,y,res)
end
```



NCL Graphics topics

- Types of graphics you can create with NCL
- The basics
- Line-by-line examples of XY plots
- Interactive demo
- Customizing NCL environment (if there's time)



Line-by-line example scripts

Most NCL scripts that follow can be viewed and downloaded from the web:

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

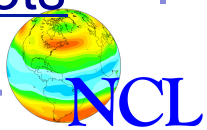
Scripts have names like *xy1a.ncl, xy1b.ncl, …*

The first one is usually one with no resources set, and each subsequent script adds a few more resources.

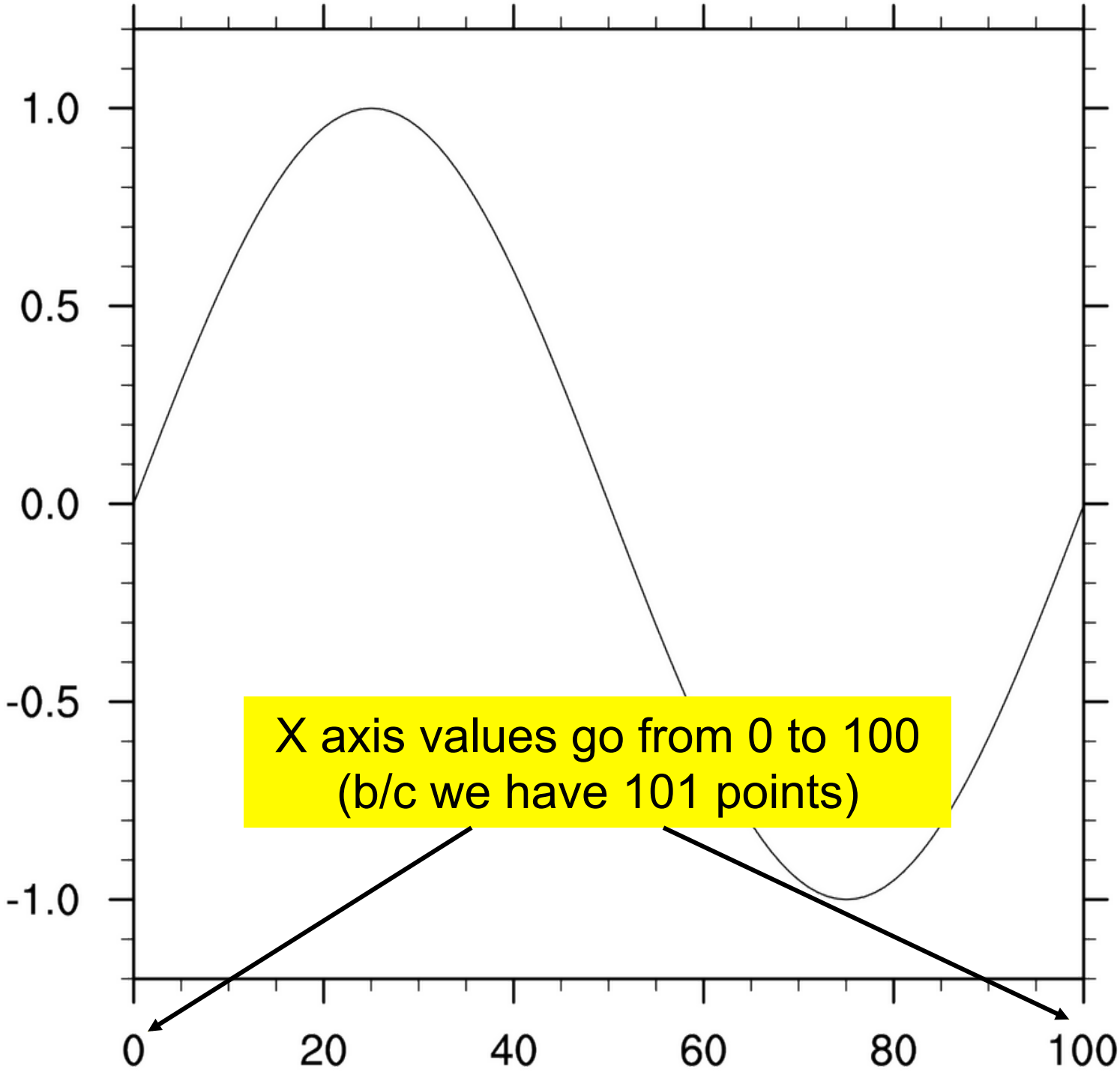
Example *xy1a.ncl*

- gsn_csm_y
- Dummy data generated in script
- Simple XY plot with a single curve, index values are used on X axis
- No resources (plot options) set
- Scripts "xy1b.ncl", "xy1c.ncl", etc. show the same example as we add more resources to further customize it.

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



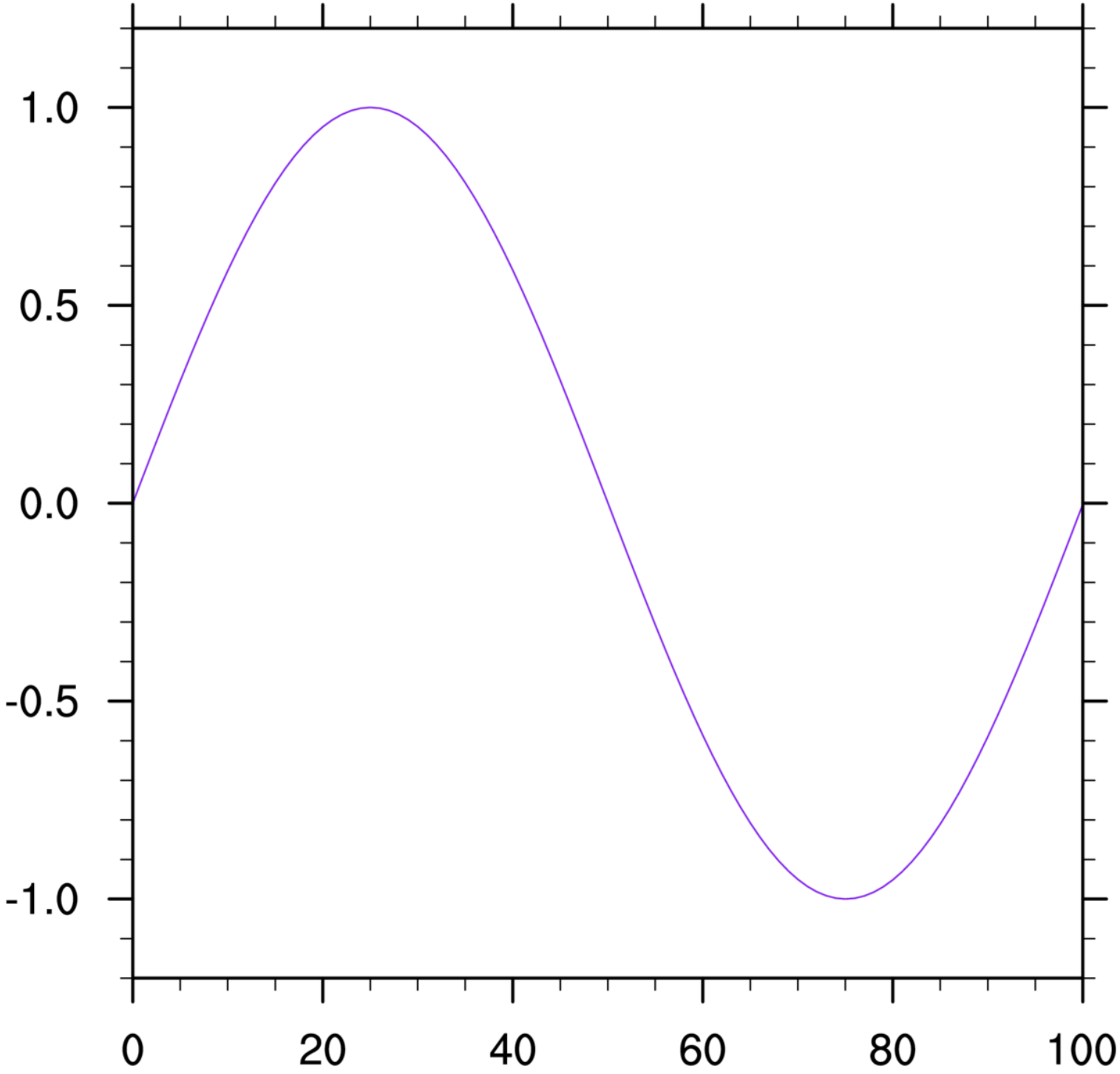
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  y = sin(0.0628*ispan(0,100,1))   ; 101 points

  wks = gsn_open_wks("ps","xy1a")  ; “xy1a.ps”

  res = True      ; no plot options desired yet
  plot = gsn_csm_y(wks,y,res)     ; no X values
end
```


1.0
0.5
0.0
-0.5
-1.0
0
20
40
60
80
100
X axis values go from 0 to 100
(b/c we have 101 points)

Example *xy1b.ncl*

- Line color changed
  (using color index values)
- Default color map discussed
- Resource introduced:
  - xyLineColor - sets line color

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

begin
  y = sin(0.0628*ispan(0,100,1))

  wks = gsn_open_wks("ps","xy1b")    ; “xy1b.ps”

; Set an XY plot resource.

  res                  = True   ; We plan to set options
  res@xyLineColor      = 2      ; Line color (old way
                                ;    of setting color)

  plot = gsn_csm_y(wks,y,res)
end
```

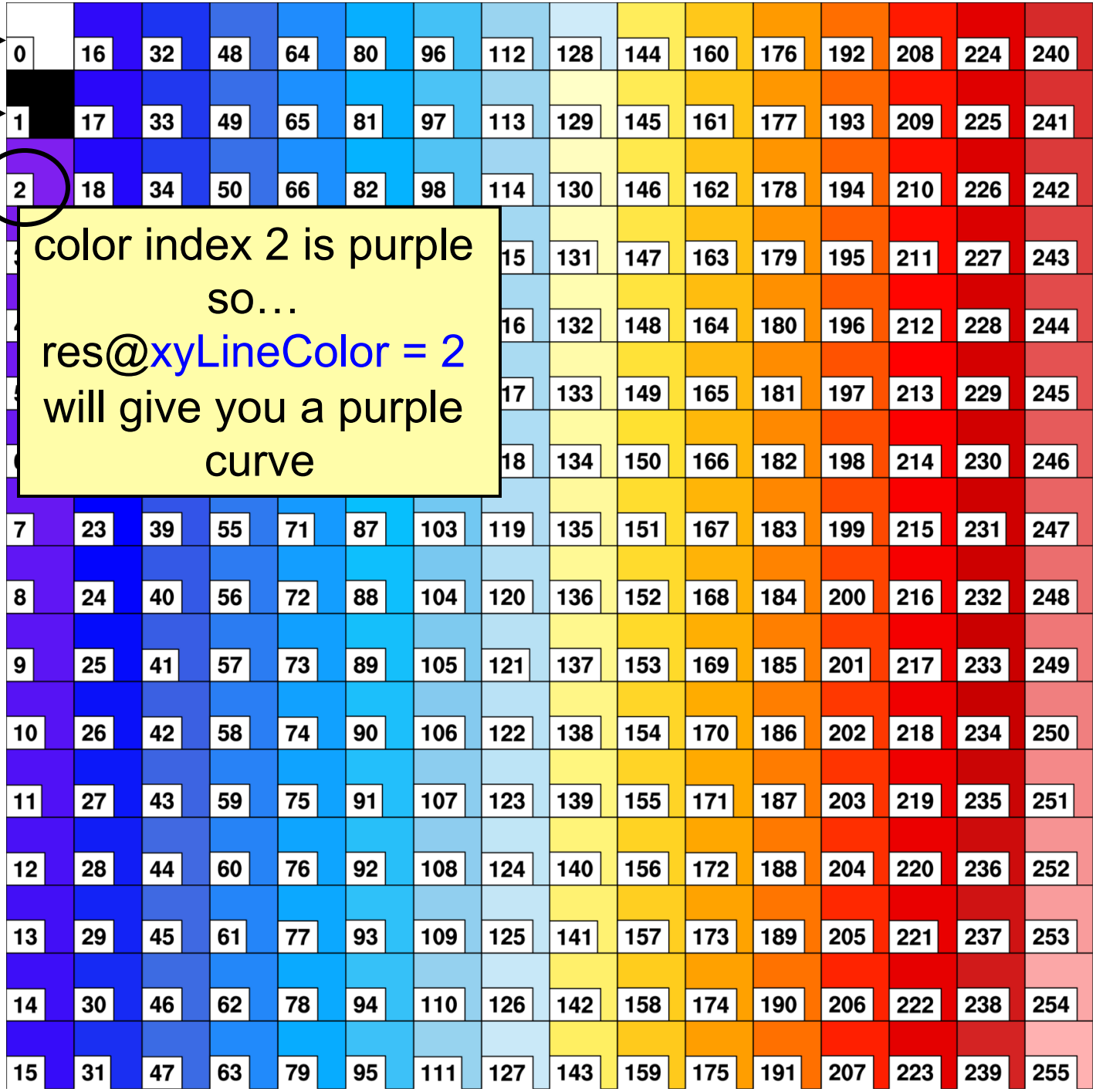

1.0
0.5
0.0
-0.5
-1.0
0
20
40
60
80
100

Index 0 is the background color

Index 1 is the foreground color

color index 2 is purple so…
res@xyLineColor = 2
will give you a purple curve

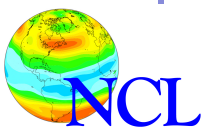
Integer values used with color resources will be indexes into the current color table



Default color table

Example *xy1c.ncl*

- gsn_csm_xy
- Better way of setting line color (using "named" color)
- Line thickness increased
- "long_name" attributes set
- Resource introduced:
  - xyLineThicknessF - sets line thickness

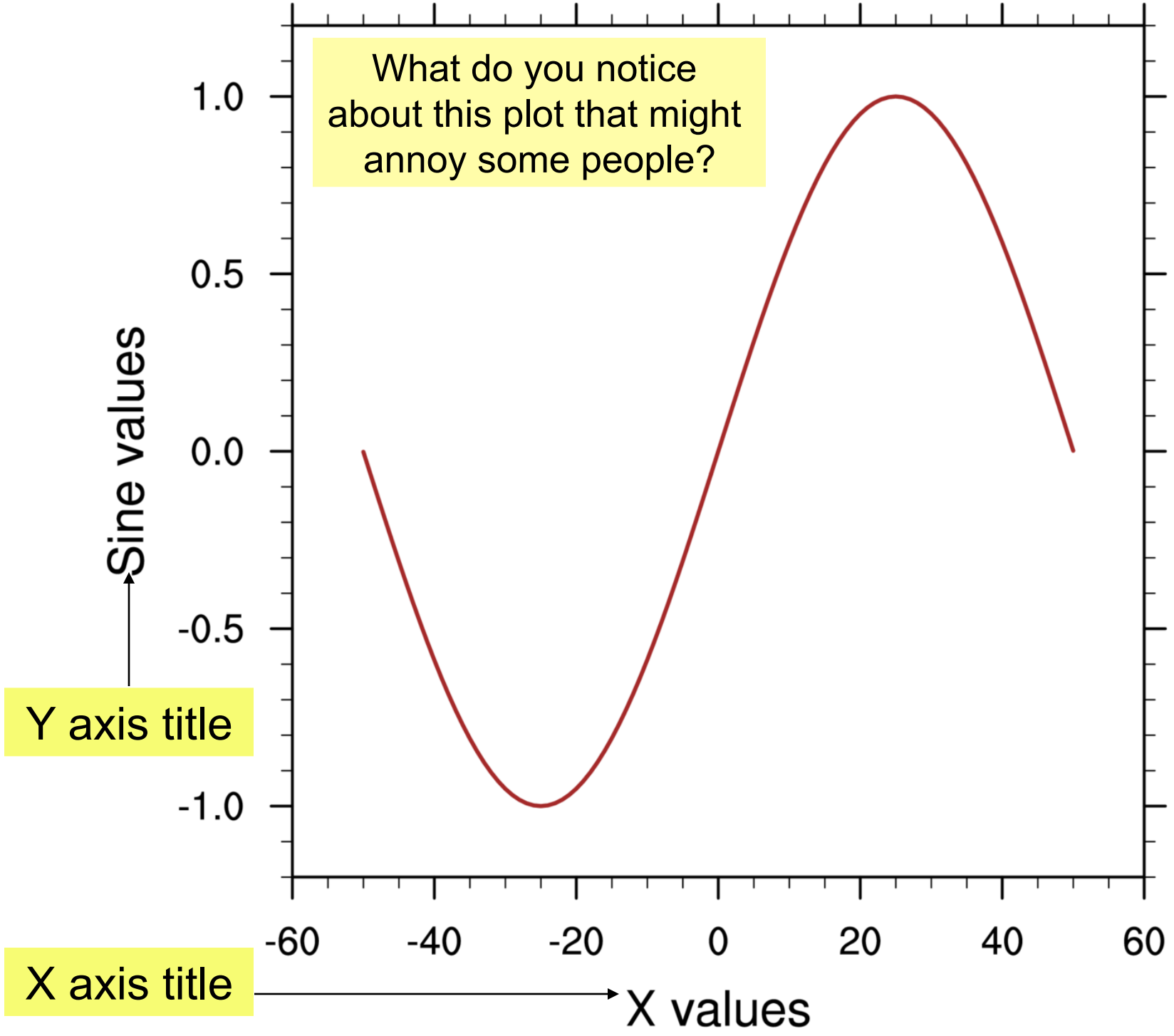


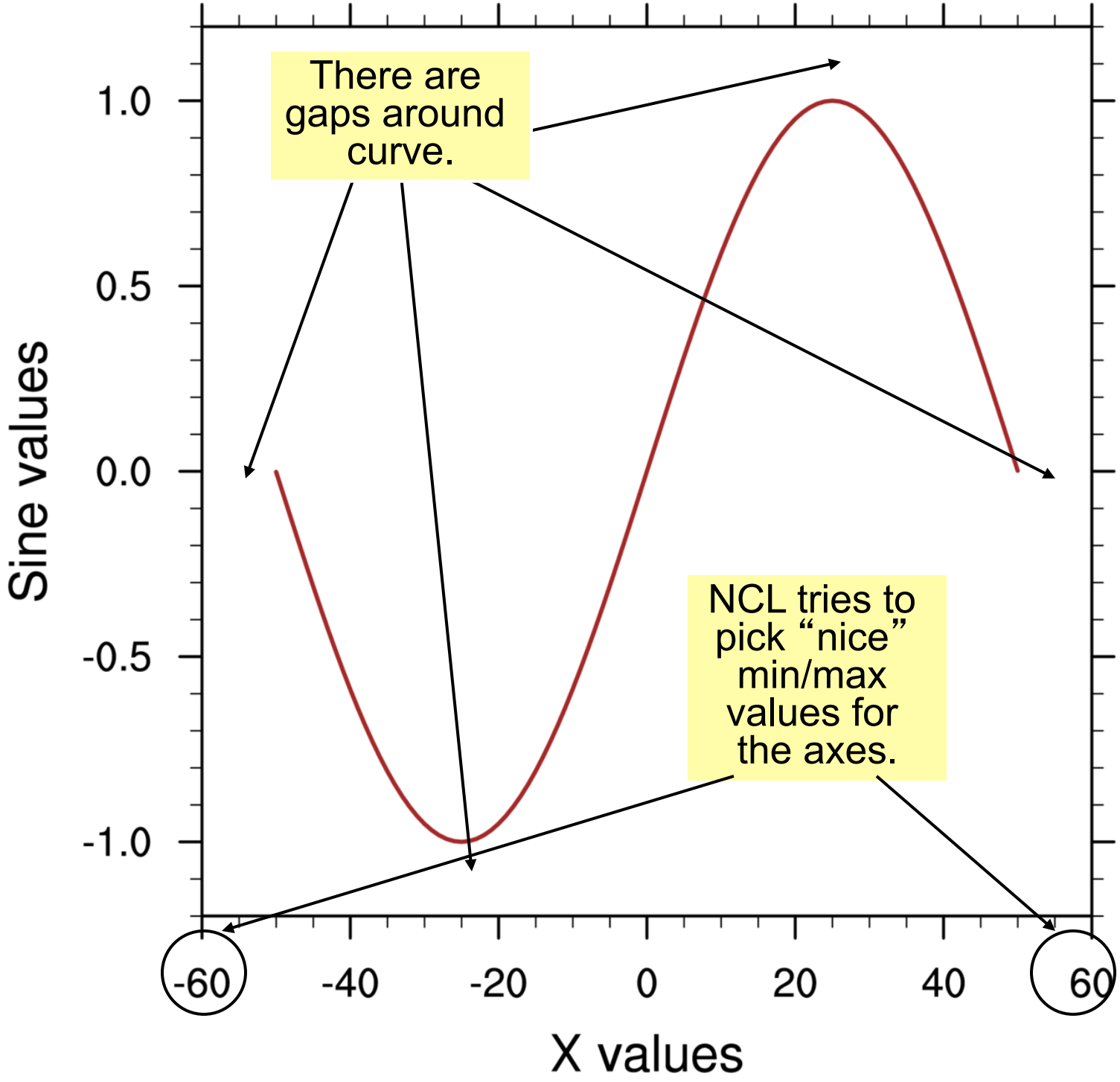
Samples of named colors

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

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

begin
  x = ispan(-50,50,1)              ; Create some X and
  y = sin(0.0628*x)                ; Y data.
  x@long_name = "X values"         ; Add long_name attributes to
  y@long_name = "Sine values"      ; see what happens to plot.

  wks = gsn_open_wks("ps","xy1c")      ; “xy1c.ps”

;---Set some XY plot resources.

  res                      = True
  res@xyLineColor          = “brown” ; “new” way of setting color
  res@xyLineThicknessF     = 3       ; 3 times thicker
                                     ; default is 1

  plot = gsn_csm_xy(wks,x,y,res)
end
```

Order of resources not important

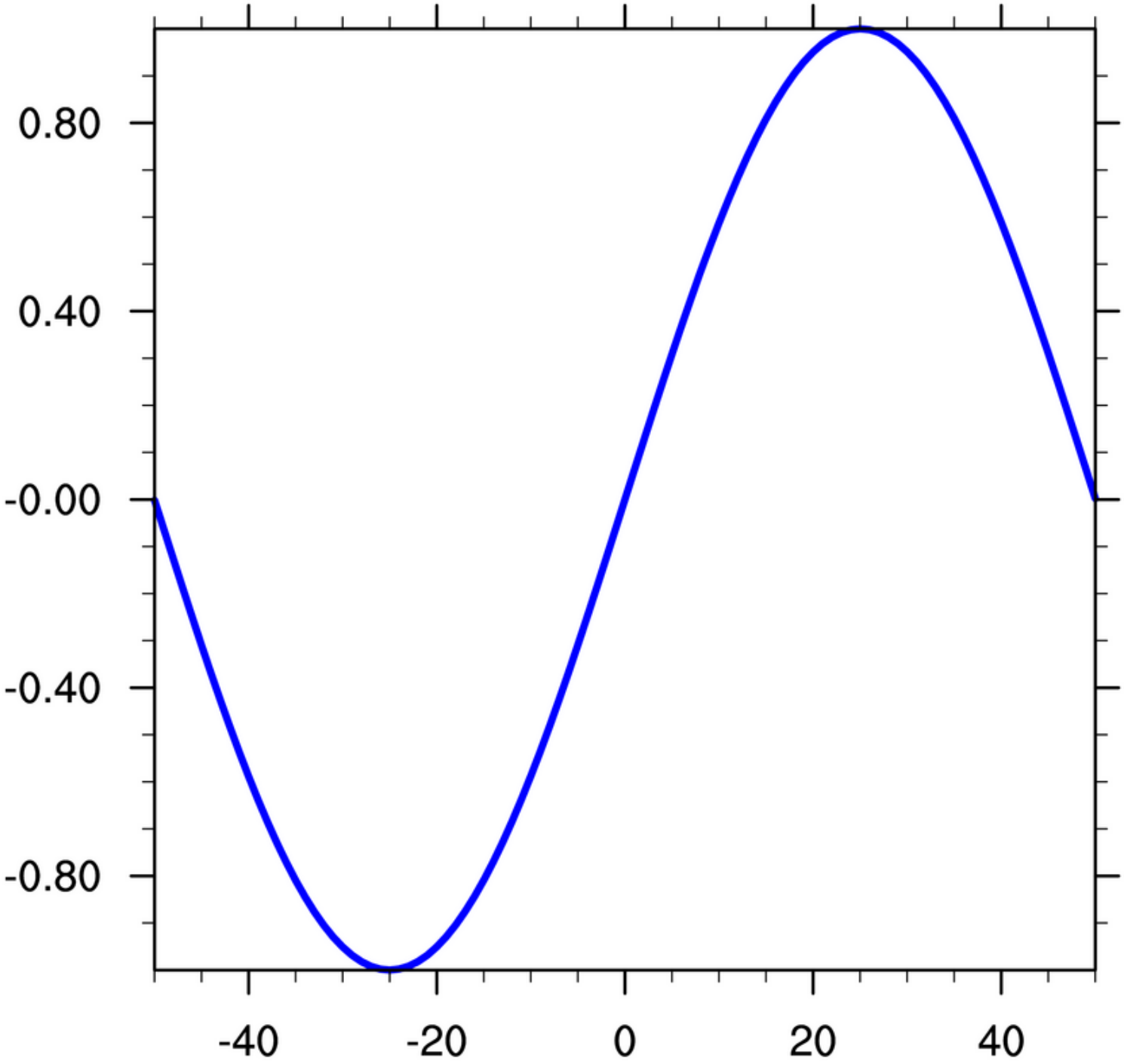

What do you notice about this plot that might annoy some people?
1.0
0.5
0.0
-0.5
-1.0
Sine values
Y axis title
-60
-40
-20
0
20
40
60
X axis title
X values


There are gaps around curve.
NCL tries to pick “nice” min/max values for the axes.
Sine values
1.0
0.5
0.0
-0.5
-1.0
-60
-40
-20
0
20
40
60
X values

Example *xy1d.ncl*

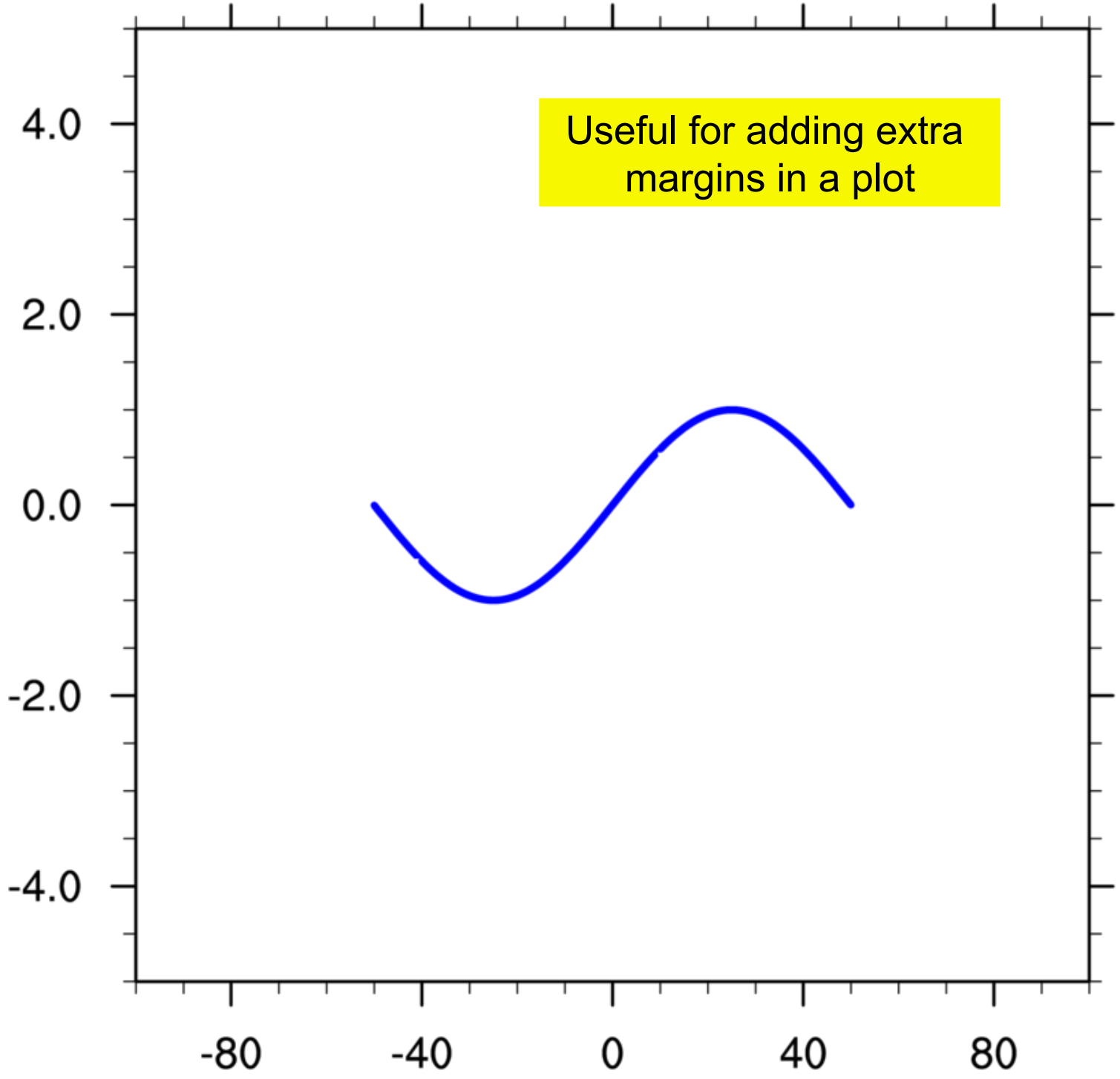
- Axes limits changed
- Special "tr" resources introduced:
  - trYMinF, trYMaxF, trXMinF, trXMaxF - sets mins and maxes for X and Y axes (**tr**ansformation resources)
- These resources can also apply to contour, vector, and streamline plots

```
...
begin
  x = ispan(-50,50,1)
  y = sin(0.0628*x)

  wks = gsn_open_wks("ps","xy1d")

  res                      = True
  res@xyLineColor          = “Blue”
  res@xyLineThicknessF = 5        ; 5x as thick

  res@trYMinF      = min(y)  ; Set axes limits.
  res@trYMaxF      = max(y)
  res@trXMinF      = min(x)
  res@trXMaxF      = max(x)

  plot = gsn_csm_xy(wks,x,y,res)
end
```


0.80
0.40
-0.00
-0.40
-0.80
-40
-20
0
20
40

```
...
begin
  x = ispan(-50,50,1)
  y = sin(0.0628*x)

  wks = gsn_open_wks("ps","xy1d_mod2")

  res                      = True
  res@xyLineColor          = “Blue”
  res@xyLineThicknessF = 5        ; 5x as thick

  res@trYMinF      =   -5  ; Set axes limits.
  res@trYMaxF      =    5
  res@trXMinF      = -100
  res@trXMaxF      =  100

  plot = gsn_csm_xy(wks,x,y,res)
end
```

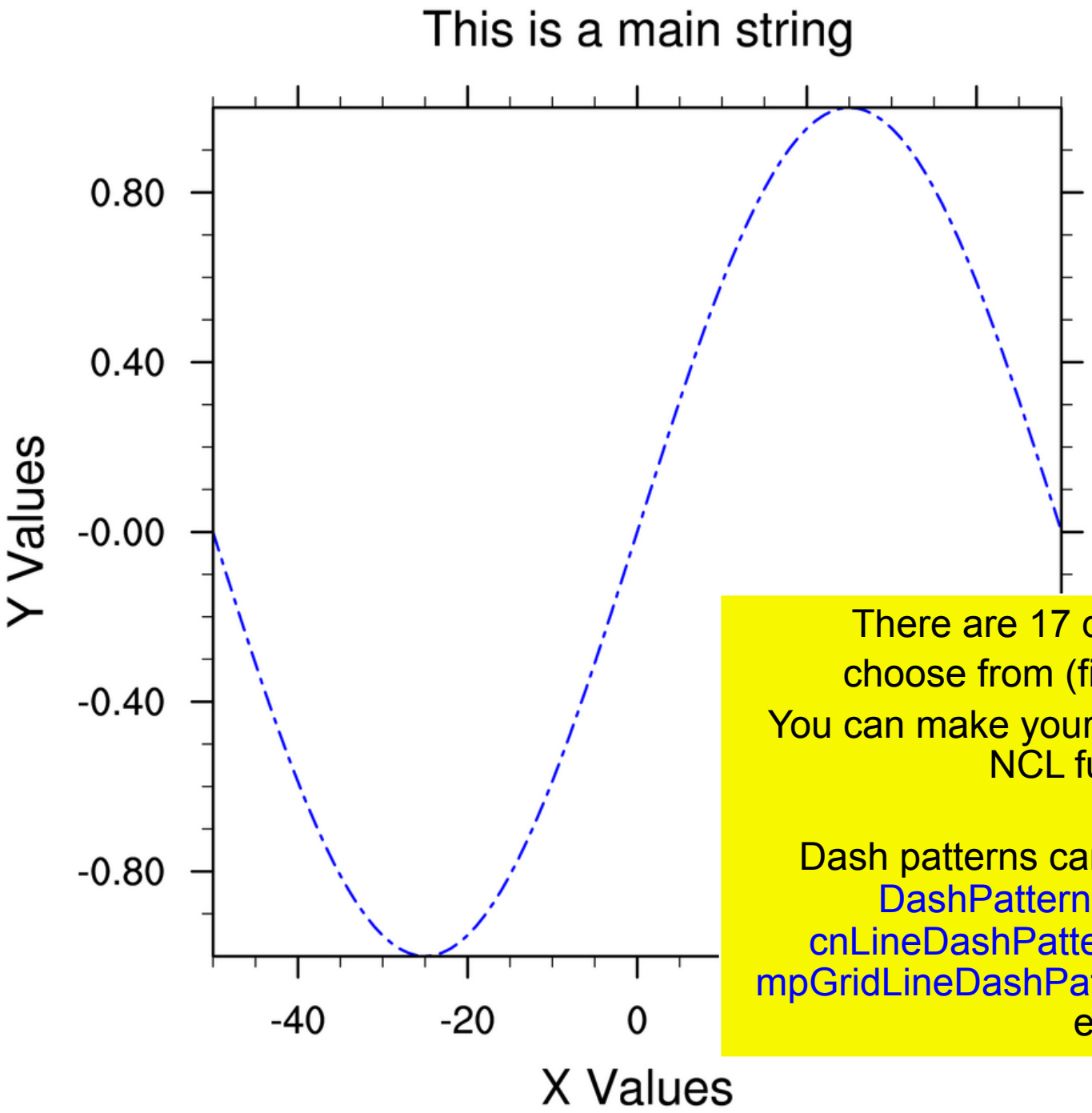
Using different values for tr* resources


Useful for adding extra margins in a plot
4.0
2.0
0.0
-2.0
-4.0
-80
-40
0
40
80

Example *xy1e.ncl*

- Dash pattern set for curve
- Title resources set
- Resources introduced:
  - xyDashPattern - sets dash pattern for curve
  - tiMainString, tiXAxisString, tiYAxisString - sets strings for axes and main title, can also be used for contour, vector, etc, plots
- tiXAxisString and tiYAxisString will override "long_name" attributes

```
...
begin
  x = ispan(-50,50,1)
  y = sin(0.0628*x)

  wks = gsn_open_wks("ps","xy1e")

  res                     = True
  res@xyLineColor         = “Blue”
  res@xyDashPattern       = 2          ; 0 is default (solid)
  res@xyLineThicknessF    = 3          ; 3x as thick

  res@trYMinF = min(y)       ; Set axes limits.
  res@trYMaxF = max(y)
  res@trXMinF = min(x)
  res@trXMaxF = max(x)

  res@tiMainString        = "This is a main string"
  res@tiXAxisString       = "X Values"
  res@tiYAxisString       = "Y Values"

  plot = gsn_csm_xy(wks,x,y,res)
end
```

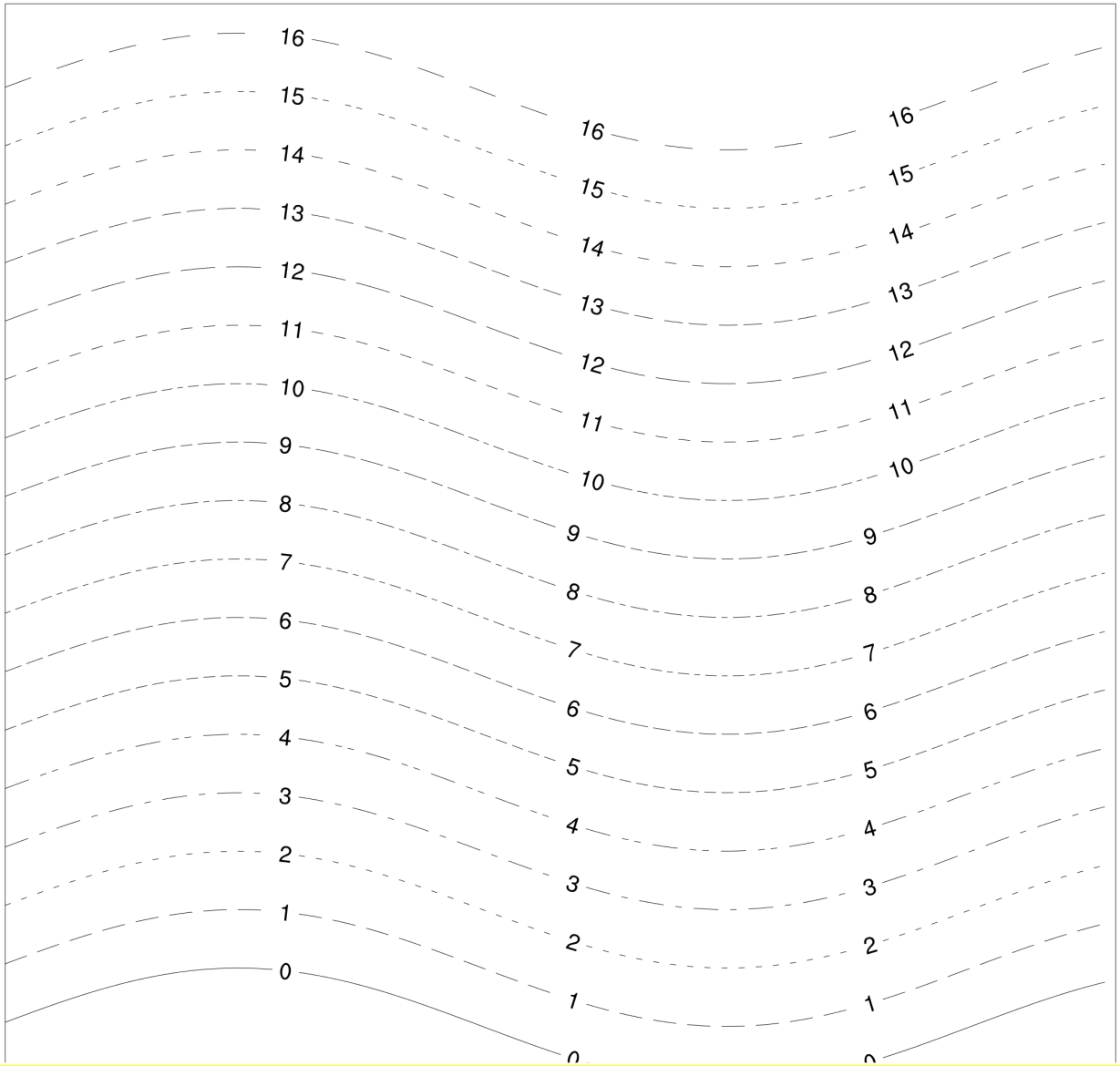


There are 17 dash patterns to choose from (first one is solid).

You can make your own using a special NCL function.

Dash patterns can be used with any DashPattern resource, like cnLineDashPattern (contour lines), mpGridLineDashPattern (map grid lines), etc.

Predefined dash patterns and their index numbers



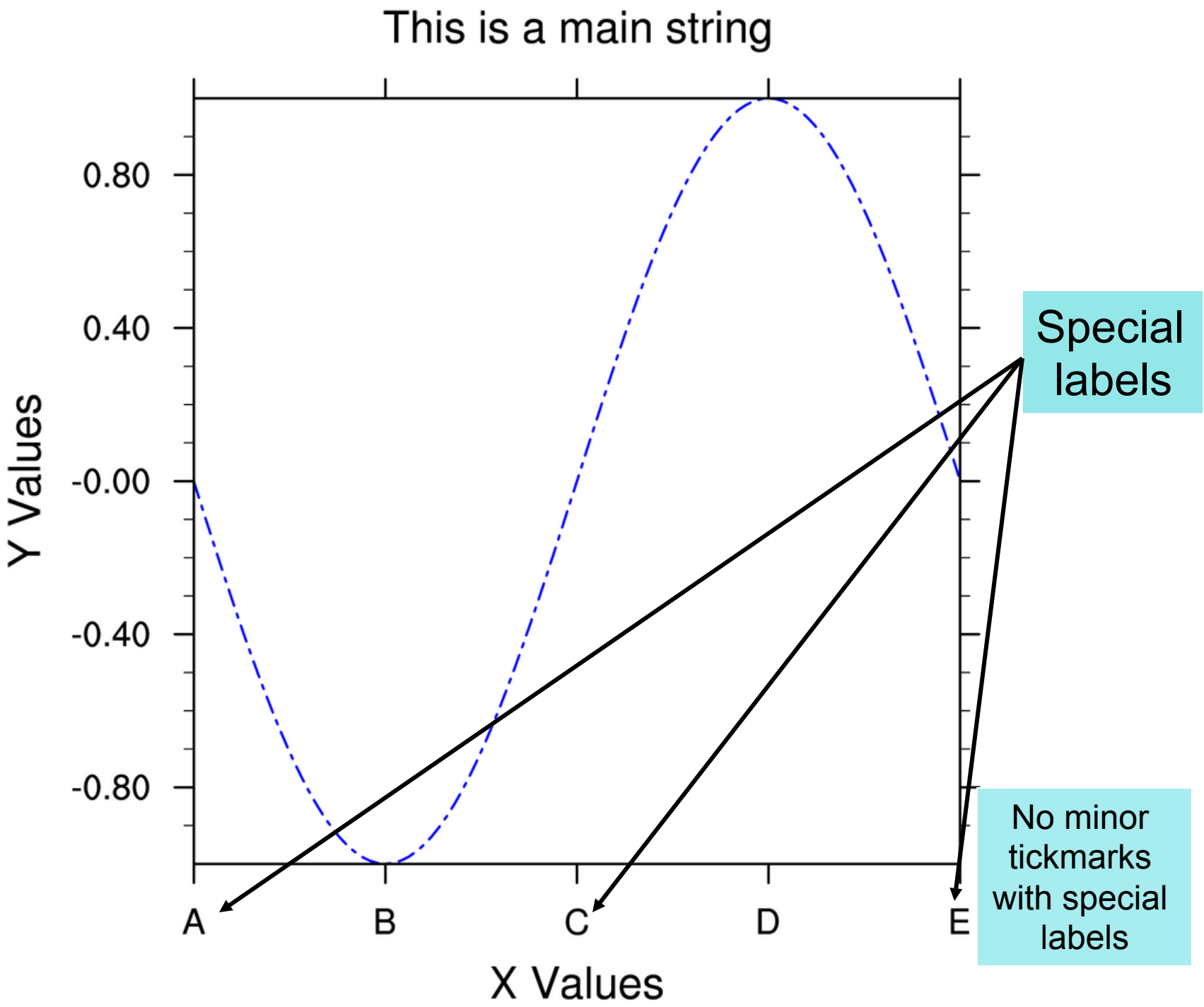
pre-defined dash patterns:

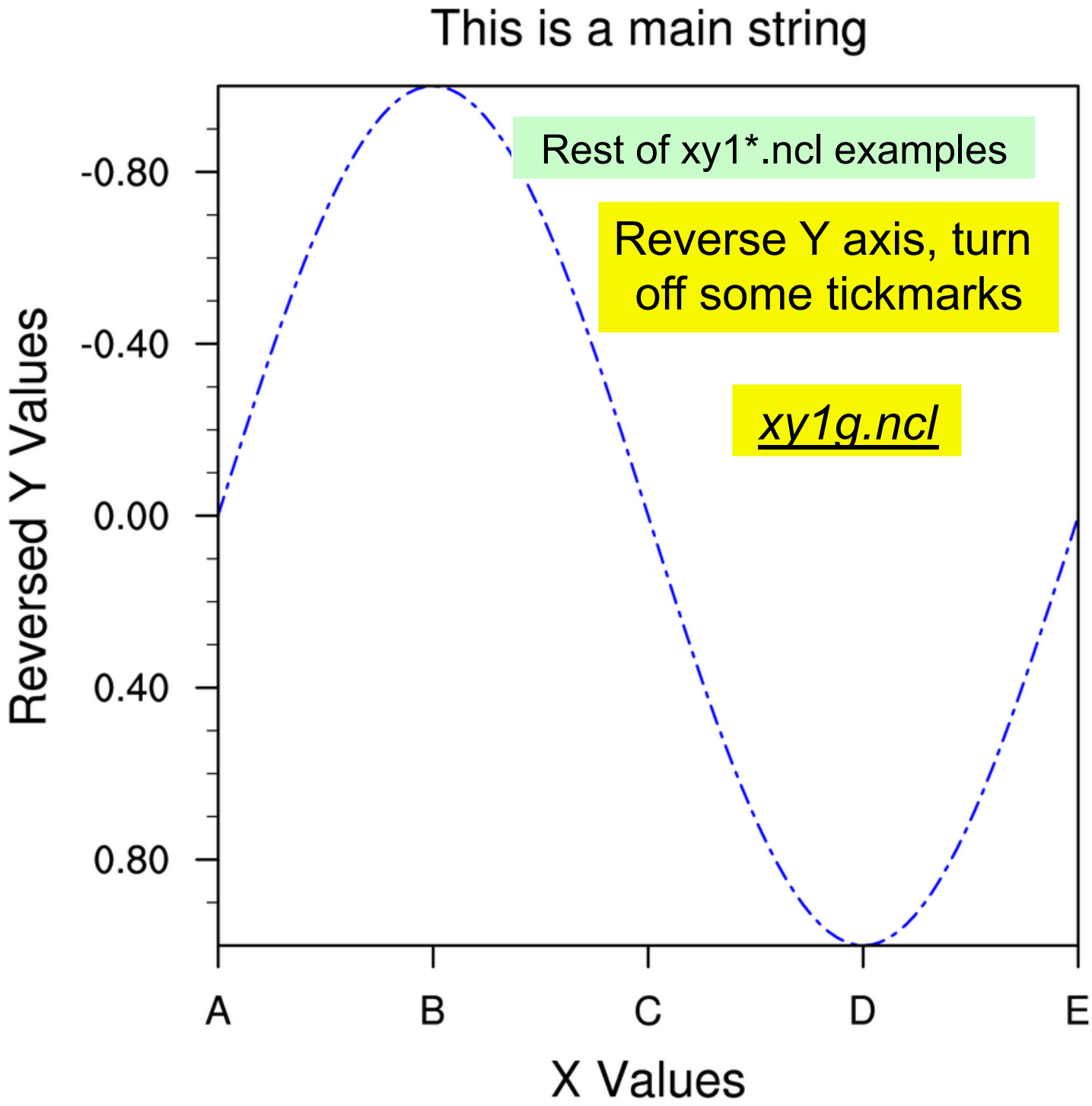
http://www.ncl.ucar.edu/Document/Graphics/dash_patterns.shtml

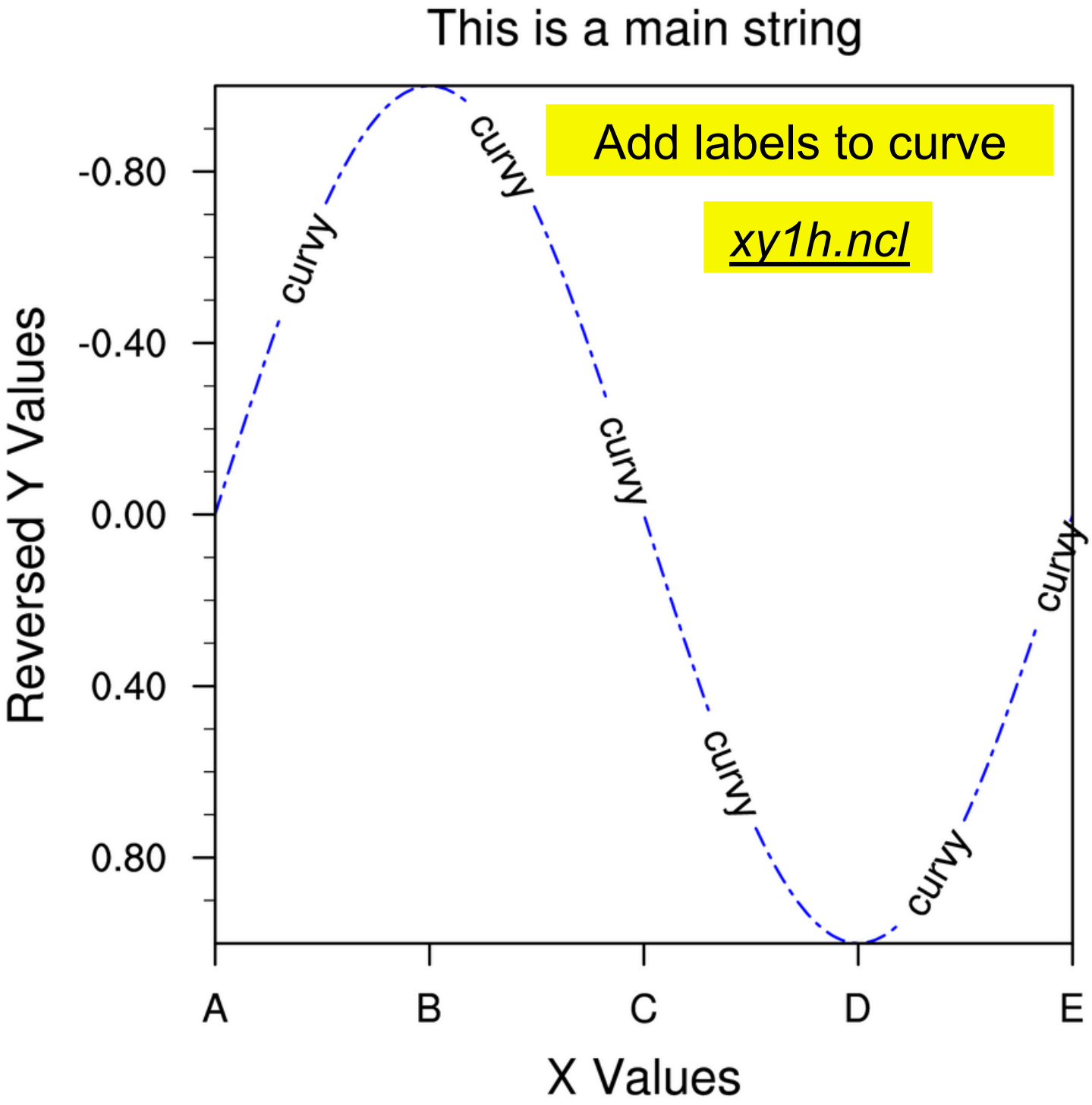
Example *xy1f.ncl*

- Explicitly set labels for bottom tickmarks
- Resources introduced:
  - tmXBMode, tmXBValues, tmXBLabels – sets locations and strings for bottom tickmarks
- Note about tickmark resources: all four axes have their own set of tickmarks, for example:
  - tmYLMode (Y left), tmYRMode (Y right), tmXBMode (X bottom), tmXTMode (X top)

```
...
begin
  x = ispan(-50,50,1)
  y = sin(0.0628*x)
  wks = gsn_open_wks("ps","xy1f")
  res                   = True      ; Set some plot options
  res@xyLineColor       = “Blue”    ; default is foreground color
  res@xyDashPattern     = 2         ; 0 is default (solid)
  res@xyLineThicknessF  = 3         ; 3x as thick
  res@trYMinF           = -0.75     ; Minimum Y axis value
  res@trYMaxF           =  0.80     ; Maximum Y axis value
  res@trXMinF           = -40.0     ; Minimum X axis value
  res@trXMaxF           =  50.0     ; Maximum X axis value
  res@tiMainString      = "This is a main string"
  res@tiXAxisString     = "X Values"
  res@tiYAxisString     = "Y Values"

  res@tmXBMode     = "Explicit"                ; Explicitly set
  res@tmXBValues   = (/-50,-25, 0,  25, 50/) ; bottom tick mark
  res@tmXBLabels   = (/"A","B","C","D","E"/) ; labels.

  plot = gsn_csm_xy(wks,x,y,res)
end
```

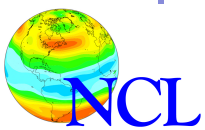

This is a main string
0.80
0.40
-0.00
-0.40
-0.80
Y Values
A
B
C
D
E
X Values
Special labels
No minor tickmarks with special labels


This is a main string
Rest of xy1*.ncl examples
Reverse Y axis, turn off some tickmarks
xy1g.ncl
Reversed Y Values
-0.80
-0.40
0.00
0.40
0.80
A
B
C
D
E
X Values


This is a main string
Add labels to curve
xy1h.ncl
curvy
curvy
curvy
curvy
curvy
curvy
Reversed Y Values
-0.80
-0.40
0.00
0.40
0.80
A
B
C
D
E
X Values

NCL Graphics topics

- Types of graphics you can create with NCL
- The basics
- Line-by-line examples of XY plots
- Interactive demo
- Customizing NCL environment (if there's time)



Advanced topics

- 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

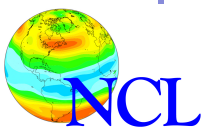
XY plot exercises

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

Click on:

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

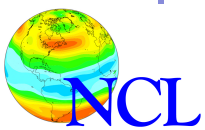
More complex (some real world) examples:
http://www.ncl.ucar.edu/Applications/xy.shtml



In review…

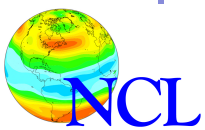
- Four main steps to create a plot

  1. Load the scripts
  2. Open a workstation
  3. Set some resources
  4. Call the desired plotting function



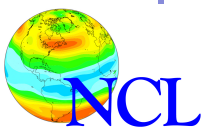
Tips

- 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



NCL Graphics topics

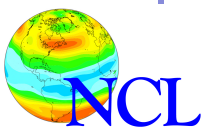
- Types of graphics you can create with NCL
- The basics
- Line-by-line examples of XY plots
- Interactive demo
- Customizing NCL environment (if there's time)



Customize your UNIX editor

- Students have contributed many nice enhanced UNIX editor features specifically for NCL scripts.
- Enhancements available for emacs, nedit, vim, TextMate, Aquamacs, NetBeans, to name a few.

  http://www.ncl.ucar.edu/Applications/editor.shtml
- Editor enhancements will highlight functions, graphical resource, comments, syntax, and other features in different colors.
- Makes debugging a little easier!



Sample editor enhancement

File Edit Options Buffers Tools Help

```
;*************************************************
; no coordinate information is available so create on our own
;*************************************************
  dim_ndvi = dimsizes(ndvi)

  nlat = dim_ndvi(0)
  lat  = latGlobeFo (nlat, "lat", "latitude", "degrees_north")
  lat  = lat(::-1)                ; grid goes from North->South

  mlon = dim_ndvi(1)
  lon  = lonGlobeFo (mlon, "lon", "longitude", "degrees_east")
  lon  = ( lon-180. )             ; grid goes DateLine eastward
;*************************************************
; data does not have named dimensions, coordinate variables or attributes,
; so we must assign
;*************************************************
  ndvi!0   = "lat"                ; name dimensions
  ndvi!1   = "lon"
  ndvi&lat =  lat                 ; assign coord variables
  ndvi&lon =  lon                 ; that were created above

  ndvi@long_name = "NDVI"         ; assign long_name
;*************************************************************************
; create plot
;*************************************************************************
  wks = gsn_open_wks("x11","avhrr")
  gsn_define_colormap(wks,"BlAqGrYeOrReVi200")

  res                     = True                      ; plot mods desired
  res@cnFillOn            = True                      ; turn on color
  res@gsnSpreadColors     = True                      ; use full range of colors
  res@cnLinesOn           = False                     ; Turn off contour lines
  res@cnFillMode          = "RasterFill"              ; Raster Mode
  res@cnLevelSpacingF     = 0.1
  res@lbLabelAutoStride= True

  res@tiMainString        = "AVHRR data"              ; title

  plot = gsn_csm_contour_map_ce(wks,ndvi, res)        ; create plot
```

--:-- avhrr_1.ncl 43% L56 (NCL)

Customize your NCL graphics environment

- Optional, was highly recommended for older versions of NCL
- Download "**.hluresfile**" file, put in home directory
- Can be used to change default color map, font, function code, etc.
- Can be used to change default size of X11 window that pops up, or default size of PNG image

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

Sample “.hluresfile”

```
! Color map
*wkColorMap                    : rainbow

*Font                          : times-roman

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

! Set size of X11 or PNG window
*wkWidth                       : 1000
*wkHeight                      : 1000
```

Questions?