Introduction to NCL Graphics

Primitives

Part V in the series

November 19, 2014

Mary Haley

Sponsored by the National Science Foundation
About this series

- Fifth in a series:
  - The basics and XY plots
  - Creating contours plots
  - Vector plots and overlays
  - Panel plots
  - Primitives

- Webinars are expanded versions of NCL workshop lectures . . . with more time for demos!

- To see other webinars in this series:

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

• Get you comfortable with creating NCL graphics . . . won’t make you an expert!

• Sneak in tips for more advanced users

• Answer your burning questions
Assumptions

Familiar with:

- basic NCL language features
- NCL array syntax
- reading data off a NetCDF file using NCL
- basic structure of an NCL graphics script
- paneling (previous webinar)
- NCL/NetCDF data model
- UNIX
Outline

- What are primitives?
- Gallery
- Functions for drawing primitives
- Demos
- Tips
What are primitives?

- Primitives include:
  - lines
  - markers
  - polygons
  - text

- You can draw primitives on:
  - any existing plot that has a “data space”
    (contour, vector, streamline, XY, map plot)
  - the NDC unit square

- Primitives have their own set of resources
Outline

• What are primitives?
• Primitives gallery
• Functions for drawing primitives
• Demos
• Tips
Primitives gallery

- Many of these images can be found with full scripts on many of our examples page:

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

- If you see one you like, make a note of it. I can point you to the full script, or help you create it later.
Primitives are markers, lines, polygons, and text used to further annotate a plot.
Map plot with red polyline boxes

The little map was added with "gsn_add_annotation"
XY plots with filled polygons
Filled polygons with text
PROBABILITY THAT CENTER OF ISABEL WILL PASS WITHIN 75 STATUTE MILES DURING THE 72 HOURS STARTING AT 8:00 PM EDT TUE SEP 16 2003

0-4%  5-9%  10-19%  20-49%  50-79%  80-100%
Outline

- What are primitives?
- Primitives gallery
- Functions for drawing primitives
- Demos
- Tips
Three ways to draw primitives

1. By attaching them to an existing plot
2. By drawing them directly on an existing plot
3. By drawing them directly on the “NDC” canvas
1. Attaching primitives to an existing plot

• **Advantage**
  • If plot is resized, primitives are resized too; especially important for paneled plots

• **Disadvantage**
  • Requires a unique id
  • Can be memory intensive if you have a lot of primitives
2. Drawing primitives directly on plot

• **Advantage**
  - Doesn’t require an id
  - Can run faster (less memory intensive)

• **Disadvantage**
  - Primitives don’t “stay” with the plot
3. Drawing primitives directly on “NDC” canvas

- The only method available if need to draw primitives outside of the plot area
- No easy way to attach these types of primitives
- If trying to add a “legend” to a plot, see:
  - gsn_create_labelbar
  - gsn_create_legend
  - gsn_create_text
<table>
<thead>
<tr>
<th>1. Primitive drawing functions (attaching to a plot)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gsn_add_polyline</code></td>
</tr>
<tr>
<td><code>gsn_add_polymarker</code></td>
</tr>
<tr>
<td><code>gsn_add_polygon</code></td>
</tr>
<tr>
<td><code>gsn_add_text</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Primitive drawing procedures (drawing directly on a plot)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gsn_polyline</code></td>
</tr>
<tr>
<td><code>gsn_polymarker</code></td>
</tr>
<tr>
<td><code>gsn_polygon</code></td>
</tr>
<tr>
<td><code>gsn_text</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Primitive drawing procedures (outside a plot, using NDC coordinates)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gsn_polyline_ndc</code></td>
</tr>
<tr>
<td><code>gsn_polymarker_ndc</code></td>
</tr>
<tr>
<td><code>gsn_polygon_ndc</code></td>
</tr>
<tr>
<td><code>gsn_text_ndc</code></td>
</tr>
</tbody>
</table>
Other useful functions and procedures

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drawNDCGrid</td>
<td>Draws NDC grid (for use with NDC primitives)</td>
</tr>
<tr>
<td>gsn_create_text</td>
<td>Create a text string</td>
</tr>
<tr>
<td>gsn_create_labelbar</td>
<td>Create a labelbar</td>
</tr>
<tr>
<td>gsn_create_legend</td>
<td>Create a legend</td>
</tr>
<tr>
<td>gsn_add_annotation</td>
<td>Attach a graphical object to another – can be quite powerful!</td>
</tr>
</tbody>
</table>

Shapefile functions (these use primitives under the hood)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsn_add_shapefile_polylines</td>
<td>Add shapefile outlines to a map</td>
</tr>
<tr>
<td>gsn_add_shapefile_polymarkers</td>
<td>Add shapefile points to a map</td>
</tr>
<tr>
<td>gsn_add_shapefile_polygons</td>
<td>Add shapefile polygons to a map</td>
</tr>
</tbody>
</table>
### Resources useful with **polyline** drawing routines

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsLineColor</td>
<td>Line color (default foreground color)</td>
</tr>
<tr>
<td>gsLineThicknessF</td>
<td>Line thickness (default 1.0)</td>
</tr>
<tr>
<td>gsLineDashPattern</td>
<td>Line dash pattern (default solid)</td>
</tr>
</tbody>
</table>

### Resources useful with **polymarker** drawing routines

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsMarkerColor</td>
<td>Marker color (default foreground color)</td>
</tr>
<tr>
<td>gsMarkerIndex</td>
<td>Marker index (default asterisk)</td>
</tr>
<tr>
<td>gsMarkerSizeF</td>
<td>Marker size (default ?)</td>
</tr>
<tr>
<td>gsMarkerThicknessF</td>
<td>Marker thickness (useful for non-filled markers)</td>
</tr>
</tbody>
</table>

### Resources useful with **polygon** drawing routines

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsFillColor</td>
<td>Fill color (default transparent)</td>
</tr>
<tr>
<td>gsFillOpacityF</td>
<td>Fill opacity (default 1.0)</td>
</tr>
<tr>
<td>gsFillIndex</td>
<td>Fill index (for pattern fill)</td>
</tr>
<tr>
<td>gsEdgesOn</td>
<td>Turn on polygon outline</td>
</tr>
<tr>
<td>gsEdgeColor</td>
<td>Color of polygon outline, if turned on</td>
</tr>
</tbody>
</table>

### Resources useful with **text** drawing routines

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>txFontColor</td>
<td>Font color (default foreground)</td>
</tr>
<tr>
<td>txFontHeightF</td>
<td>Font height</td>
</tr>
<tr>
<td>txPerimOn</td>
<td>Turn on text box perimeter</td>
</tr>
</tbody>
</table>

---

*Lines, markers, and polygons all use “GraphicStyle” (“gs”) resources. Text uses “TextItem” (“tx”) resources.*
Outline

• What are primitives?
• Primitives gallery
• Functions for drawing primitives
• Demos
• Tips
Demo #1

- Start with “prim1a.ncl” script
- Scripts (prim1x.ncl) and data file available from:

  NCL Home Page -> Support -> Webinars -> NCL Graphics

  [Link](http://www.ncl.ucar.edu/Training/Webinars/NCL_Graphics/PrimitivesDemo)
f = addfile ("uv300.nc","r")
u = f->U ; read U

wks = gsn_open_wks ("x11","prim1a")

res = True
res@xyMonoDashPattern = True ; use solid lines only
res@xyLineColors = ("NavyBlue","Brown3")
res@xyLineThicknessF = 2.0

lon = 82 ; Which longitude to select
plot = gsn_csm_xy (wks,u&lat,u(:,:,{lon}),res)
- Draw a text string at (-30,45)
f = addfile("uv300.nc","r")
u = f->U ; read U

wks = gsn_open_wks("x11","primlf")
res = True
res@gsnDraw = False ; Don't draw plot yet
res@gsnFrame = False ; Don't advance frame
res@xyMonoDashPattern = True ; Use solid lines only
res@xyLineColors = ("NavyBlue","Brown3")
res@xyLineThicknessF = 2.0
res@trYMinF = min(u)
res@trYMaxF = max(u)

lon1 = 70
plot1 = gsn_csm_xy(wks,u&lat,u(:,:,{lon1}),res) ; Create plots at different longitudes
lon2 = 0
plot2 = gsn_csm_xy(wks,u&lat,u(:,:,{lon2}),res)

lonstr1 = "lon = " + lon1 ; Create longitude strings to add to plot
lonstr2 = "lon = " + lon2

txres = True
txres@txFontHeightF = 0.02
txres@txJust = "TopRight"
id = gsn_add_text(wks,plot1,lonstr1,max(u&lat),max(u)-1,txres) ; Attach longitude strings
id = gsn_add_text(wks,plot2,lonstr2,max(u&lat),max(u)-1,txres)

txres@txJust = "CenterLeft"
txres@txFontColor = "NavyBlue"
id = gsn_add_text(wks,plot1,"January",-60,-5,txres) ; Attach month strings
id = gsn_add_text(wks,plot2,"January",-60,-5,txres)

txres@txFontColor = "Brown3"
id = gsn_add_text(wks,plot1,"July",-60,0,txres)
id = gsn_add_text(wks,plot2,"July",-60,0,txres)

pres = True
gsn_panel(wks,(/plot1,plot2/),(/1,2/),pres) ; Panel plots
Demo #2

• Start with “prim2c.ncl” script

• Scripts (prim2x.ncl) and data file available from:

  NCL Home Page -> Support -> Webinars -> NCL Graphics

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

• Australia administrative shapefiles available from:

  http://www.gadm.org/country
wks = gsn_open_wks("x11","prim2c")
res = True
res@gsnDraw = False
res@gsnFrame = False

res@mpDataBaseVersion = "MediumRes" ; Medium resolution
res@mpDataSetName = "Earth..4" ; Better outlines

res@mpLimitMode = "LatLon" ; Zoom in on Australia
res@mpMinLatF = -45 ; Zoom in on Australia
res@mpMaxLatF = -10
res@mpMinLonF = 112
res@mpMaxLonF = 155
res@mpFillOn = False ; Turn off map fill

map = gsn_csm_map(wks,res) ; Create map

cities = ("Adelaide", "Perth", "Brisbane", "Melbourne", "Sydney")
latt = (-34.93, -31.95, -27.47, -37.81, -33.86)
lon = (138.60, 115.86, 153.03, 144.96, 151.21)

;---Attach markers to map.
mres = True
mres@gsMarkerSizeF = 10. ; Make markers larger
mres@gsMarkerColor = "blue"
mres@gsMarkerIndex = 16 ; Filled dot
dum1 = gsn_add_polymarker(wks,map,lon,lat,mres)

;---Attach text to map.
tres = True
tres@txFontHeightF = 0.015 ; Default is large
tres@txJust = "CenterLeft" ; default is "CenterCenter"

dum2 = gsn_add_text(wks,map,cities,lon+0.5,lat,tres)
draw(map) ; Drawing the map draws the markers
frame(wks) ; Advance the frame
Outline

• What are primitives?
• Primitives gallery
• Functions for drawing primitives
• Demos
• Tips
Tips for drawing primitives

• `gsn_add_polyxxx` generally preferred over `gsn_polyxxx`

• Be sure to use unique id’s if using `gsn_add_polyxxx`

• If drawing a lot of primitives, then `gsn_polyxxx` may be better. Check out the new “gsSegments” resource

• Use “txJust” resource to control positioning of text

• Use “drawNDCGrid” procedure to help with drawing primitives on the canvas
Illustrates use of “txJust”

Example “text_13.ncl” on “text” examples page

Nine possible vaues for resource ’txJust’

Red box denotes bounding box, used to determine location of justification point.

Blue dot w/label shows location and value of txPosXF/txPosYF.

(txPosXF, txPosYF)

TopLeft: (0.75, 0.82)

TopCenter: (0.75, 0.55)

TopRight: (0.75, 0.28)

CenterLeft: (0.5, 0.73)

Center: (0.5, 0.46)

CenterRight: (0.5, 0.19)

BottomLeft: (0.25, 0.64)

BottomCenter: (0.25, 0.37)

BottomRight: (0.25, 0.1)
If primitives not showing up

• Are x,y are in correct data space?

• Are x,y location arguments in correct order?
  Common mistake is using “lat,lon” order for map coordinates; it should be “lon,lat”.

• Make sure you are drawing things in the right order (i.e. drawing the primitives after you draw the plot).

• May need to use one of the “DrawOrder” resources mpOutlineDrawOrder, xyCurveDrawOrder, cnFillDrawOrder, etc
If primitives taking a long time to draw

• If using gsn_add_polyxxx, then this can be expensive memory-wise. Consider using gsn_polyxxx.

• If drawing lines or polygons, then “gsSegments” can significantly speed things up.
To create legends / labelbars

• Use `gsn_create_labelbar` or `gsn_create_legend` with `gsn_add_annotation` to attach to an existing plot

• Use `gsn_labelbar_ndc` or `gsn_legend_ndc` to draw on the canvas somewhere

• Use primitive drawing routines if need more customization
Where to learn more

• Examples page:
  http://www.ncl.ucar.edu/Applications/

• In particular:
  – “Polygons”, “Polylines”, “Polymarkers”
  – “Annotations”
  – “Text”
  – “Legends”, “Labelbars”
  – “Draw order”

• List of functions and every example they appear in:
  http://www.ncl.ucar.edu/Applications/func_list.shtml
Questions?
nci-talk@ucar.edu
http://mailman.ucar.edu/mailman/listinfo/nci-talk

NCAR Mesa Lab
November 13, 2014