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

Y axis string: “long_name”

Each line a different pattern

Tickmarks pointing outward

Zonal Wind

180 120W 60W 0 60E 120E 180
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
More examples of graphical objects

January

Subtitles

Temperature

Degrees K

Map

Tickmarks (both the ticks and the labels)

Labelbar and its labels

Contours
Description of resources

• Starts with 2 or 3 lower-case letters based on object it is associated with. Some examples:
  – “xy” - XY Plot    “cn” - Contour plot
  – “vc” - Vector plot “ti” - Title
  – “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: “xyLineThicknessF”

• “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:

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

• Most have default values.

• There are many types:

```ncl
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, 3.0)
```
• Resources across objects are similarly named for easier recollection:

- \texttt{xyLineColor}, \texttt{cnLineColor}, \texttt{gsLineColor}, \texttt{mpGridLineColor}, \texttt{tmBorderLineColor}

- \texttt{tiMainFontHeightF}, \texttt{tmXBLLabelFontHeightF}, \texttt{lbLabelFontHeightF}, \texttt{cnLineLabelFontHeightF}

- \texttt{xyDashPattern}, \texttt{mpPerimLineDashPattern}, \texttt{lbBoxLineDashPattern}, \texttt{cnLineDashPattern}

and so on...
Step 4: Call appropriate plotting function

```ncl
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 \texttt{gsn_csm_xxxxx} plotting functions.
• Some examples:

\begin{verbatim}
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)
\end{verbatim}

These are all functions. You can use whatever name you like for variable name on left side of “=” sign.
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
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","xylb") ; "xylb.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
<table>
<thead>
<tr>
<th>Index 0 is the background color</th>
<th>Index 1 is the foreground color</th>
</tr>
</thead>
</table>

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

- Color index 2 is purple, so...
- `res@xyLineColor = 2` will give you a purple curve.

**Default color table**

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
</tr>
<tr>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
</tr>
<tr>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
</tr>
<tr>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
</tr>
<tr>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
</tr>
<tr>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
<td>223</td>
</tr>
<tr>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
<td>228</td>
<td>229</td>
<td>230</td>
<td>231</td>
<td>232</td>
<td>233</td>
<td>234</td>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
</tr>
<tr>
<td>240</td>
<td>241</td>
<td>242</td>
<td>243</td>
<td>244</td>
<td>245</td>
<td>246</td>
<td>247</td>
<td>248</td>
<td>249</td>
<td>250</td>
<td>251</td>
<td>252</td>
<td>253</td>
<td>254</td>
<td>255</td>
</tr>
</tbody>
</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


<table>
<thead>
<tr>
<th>Sample</th>
<th>Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>seashell3</td>
<td>slategrey</td>
</tr>
<tr>
<td>seashell4</td>
<td>snow</td>
</tr>
<tr>
<td>sienna</td>
<td>snow1</td>
</tr>
<tr>
<td>sienna1</td>
<td>snow2</td>
</tr>
<tr>
<td>sienna2</td>
<td>snow3</td>
</tr>
<tr>
<td>sienna3</td>
<td>snow4</td>
</tr>
<tr>
<td>sienna4</td>
<td>springgreen</td>
</tr>
<tr>
<td>skyblue</td>
<td>steelblue1</td>
</tr>
<tr>
<td>skyblue1</td>
<td>steelblue2</td>
</tr>
<tr>
<td>skyblue2</td>
<td>steelblue3</td>
</tr>
<tr>
<td>skyblue3</td>
<td>steelblue4</td>
</tr>
<tr>
<td>skyblue4</td>
<td>tan</td>
</tr>
<tr>
<td>slateblue</td>
<td>tan1</td>
</tr>
<tr>
<td>slateblue1</td>
<td>tan2</td>
</tr>
<tr>
<td>slateblue2</td>
<td>tan3</td>
</tr>
<tr>
<td>slateblue3</td>
<td>tan4</td>
</tr>
<tr>
<td>slateblue4</td>
<td>lightblue1</td>
</tr>
<tr>
<td>linen</td>
<td>mediumpurple4</td>
</tr>
<tr>
<td>magenta</td>
<td>mediumseagreen</td>
</tr>
<tr>
<td>magenta1</td>
<td>mediumslateblue</td>
</tr>
<tr>
<td>magenta2</td>
<td>mediumspringgreen</td>
</tr>
<tr>
<td>magenta3</td>
<td>mediumturquoise</td>
</tr>
<tr>
<td>magenta4</td>
<td>mediumvioletred</td>
</tr>
<tr>
<td>maroon</td>
<td>midnightblue</td>
</tr>
<tr>
<td>maroon1</td>
<td>mintcream</td>
</tr>
<tr>
<td>maroon2</td>
<td>mistyrose</td>
</tr>
<tr>
<td>maroon3</td>
<td>mistyrose1</td>
</tr>
<tr>
<td>maroon4</td>
<td>mistyrose2</td>
</tr>
<tr>
<td>mediumaquamarine</td>
<td>mistyrose3</td>
</tr>
<tr>
<td>mediumblue</td>
<td>mistyrose4</td>
</tr>
<tr>
<td>mediumorchid</td>
<td>moccasin</td>
</tr>
<tr>
<td>mediumorchid1</td>
<td>navajowhite</td>
</tr>
<tr>
<td>mediumorchid2</td>
<td>navajowhite1</td>
</tr>
<tr>
<td>mediumorchid3</td>
<td>navajowhite2</td>
</tr>
<tr>
<td>mediumorchid4</td>
<td>navajowhite3</td>
</tr>
<tr>
<td>mediumpurple</td>
<td>navajowhite4</td>
</tr>
<tr>
<td>mediumpurple1</td>
<td>navy</td>
</tr>
<tr>
<td>mediumpurple2</td>
<td>navyblue</td>
</tr>
<tr>
<td>mediumpurple3</td>
<td>oldlace</td>
</tr>
<tr>
<td>grey99</td>
<td>khaki1</td>
</tr>
<tr>
<td>honeydew</td>
<td>khaki2</td>
</tr>
<tr>
<td>honeydew1</td>
<td>khaki3</td>
</tr>
<tr>
<td>honeydew2</td>
<td>khaki4</td>
</tr>
<tr>
<td>honeydew3</td>
<td>lavender</td>
</tr>
<tr>
<td>lavenderblue</td>
<td>lavenderblush1</td>
</tr>
<tr>
<td>hotpink</td>
<td>lavenderblush2</td>
</tr>
<tr>
<td>hotpink1</td>
<td>lavenderblush3</td>
</tr>
<tr>
<td>hotpink2</td>
<td>lavenderblush4</td>
</tr>
<tr>
<td>hotpink3</td>
<td>lawngreen</td>
</tr>
<tr>
<td>indianred</td>
<td>lemonchiffon</td>
</tr>
<tr>
<td>indianred1</td>
<td>lemonchiffon1</td>
</tr>
<tr>
<td>indianred2</td>
<td>lemonchiffon2</td>
</tr>
<tr>
<td>indianred3</td>
<td>lemonchiffon3</td>
</tr>
<tr>
<td>indianred4</td>
<td>lemonchiffon4</td>
</tr>
<tr>
<td>ivory</td>
<td>lightblue</td>
</tr>
<tr>
<td>ivory1</td>
<td>lightblue1</td>
</tr>
<tr>
<td>ivory2</td>
<td>lightblue2</td>
</tr>
<tr>
<td>ivory3</td>
<td>lightblue3</td>
</tr>
<tr>
<td>ivory4</td>
<td>lightblue4</td>
</tr>
<tr>
<td>darkorchid</td>
<td>deeppink2</td>
</tr>
<tr>
<td>darkorchid1</td>
<td>deeppink3</td>
</tr>
<tr>
<td>darkorchid2</td>
<td>deeppink4</td>
</tr>
<tr>
<td>darkorchid3</td>
<td>deepskyblue</td>
</tr>
<tr>
<td>darkorchid4</td>
<td>deepskyblue1</td>
</tr>
<tr>
<td>darksalmon</td>
<td>deepskyblue2</td>
</tr>
<tr>
<td>darkseagreen</td>
<td>deepskyblue3</td>
</tr>
<tr>
<td>darkseagreen1</td>
<td>deepskyblue4</td>
</tr>
<tr>
<td>darkseagreen2</td>
<td>dimgray</td>
</tr>
<tr>
<td>darkseagreen3</td>
<td>dimgrey</td>
</tr>
<tr>
<td>darkseagreen4</td>
<td>dodgerblue</td>
</tr>
<tr>
<td>darkslategray</td>
<td>dodgerblue1</td>
</tr>
<tr>
<td>darkslategray1</td>
<td>dodgerblue2</td>
</tr>
<tr>
<td>darkslategray2</td>
<td>dodgerblue3</td>
</tr>
<tr>
<td>darkslategray3</td>
<td>dodgerblue4</td>
</tr>
<tr>
<td>darkslategray4</td>
<td>firebrick</td>
</tr>
<tr>
<td>darkslategrey</td>
<td>firebrick1</td>
</tr>
<tr>
<td>darkslategrey1</td>
<td>firebrick2</td>
</tr>
<tr>
<td>darkslategrey2</td>
<td>firebrick3</td>
</tr>
<tr>
<td>darkviolet</td>
<td>firebrick4</td>
</tr>
<tr>
<td>deepink</td>
<td>floralwhite</td>
</tr>
<tr>
<td>deepink1</td>
<td>forestgreen</td>
</tr>
</tbody>
</table>
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","xylc")     ; “xylc.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
What do you notice about this plot that might annoy some people?
NCL tries to pick “nice” min/max values for the axes. There are gaps around the curve.
Example *xy1d.ncl*

- Axes limits changed
- Special “tr” resources introduced:
  - trYMinF, trYMaxF, trXMinF, trXMaxF - sets mins and maxes for X and Y axes (*transformation 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
... 

begin 

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

wks = gsn_open_wks("ps","xyld_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 \texttt{tr*} resources
Useful for adding extra margins in a plot
Introduction to NCL Graphics

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","xyle")
    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:
Example \texttt{xy1f.ncl}

- Explicitly set labels for bottom tickmarks
- Resources introduced:
  - \texttt{tmXBM}ode, \texttt{tmX}BValues, \texttt{tmX}BLabels – sets locations and strings for bottom tickmarks
- Note about tickmark resources: all four axes have their own set of tickmarks, for example:
  - \texttt{tmYLM}ode (Y left), \texttt{tmYRM}ode (Y right), \texttt{tmXBM}ode (X bottom), \texttt{tmXTM}ode (X top)
begin
  x = ispan(-50,50,1)
  y = sin(0.0628*x)

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

  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

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
This is a main string

Add labels to curve

xy1h.ncl
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

; no coordinate information is available so create on our own

dim_ndvi = dim_sizes(ndvi)

nlat = dim_ndvi(0)
lon = latGlobeFo (nlat, "lat", "latitude", "degrees_north")
lon = 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", "ahrr")
gsn_define_colormap(wks, "BlaqInYelOrReVi200")

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@llLabel@AutoStride = True

res@tMainString = "AVHRR data" ; title

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

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?