

NCL 绘图参考手册

根据 NCL Mini Graphics Manual 翻译

有用的链接:

NCL 主页: <http://www.ncl.ucar.edu/>

本手册下载: <http://www.ncl.ucar.edu/Document/Manuals/>

脚本例子和样本图形: <http://www.ncl.ucar.edu/Applications/>
http://www.ncl.ucar.edu/Document/Manuals/Getting_Started

关键字	courier-bold
内置函数	courier-bold blue
用户贡献函数	courier-bold green
实用函数	courier-bold purple
符号	bold
操作符	bold
绘图模板	courier-bold green
绘图资源	courier-bold
用户变量	<i>italics</i>
WWW 链接	<u>underline</u>

致谢

衷心感谢上海海洋大学海洋科学学院胡松老师的悉心指导;感谢南京信息工程大学大气科学学院海洋科学系程军老师领我入门;感谢刘畅对本手册耐心细致的修订工作。

本人在学习 NCL 过程中,对手册进行了翻译,仅供方便学习查找使用。由于本人水平有限,本手册还存在许多不足,对一些专业名词略有模糊,望大家指正,可发邮件至:

l_n.2006@yahoo.com.cn

崔琳琳

2010 年 7 月 31 日 于上海海洋大学

目 录

第一章 引言	4
1.1 脚本样本	4
第二章 高级绘图界面	4
2.1 gsn 一般界面	5
2.2 gsn_csm 界面	5
2.3 我该用什么界面?	5
2.4 gsn_special 界面	6
2.5 下载界面	6
2.6 gsn_csm 期望	6
第三章 入门指南	6
3.1 \$NCARG_ROOT	6
3.2 .hluresfile	7
3.3 运行 NCL	7
第四章 工作台	8
第五章 通过源代码修改图形	8
5.1 源代码类型	8
5.2 设置源代码	8
5.3 常用的源代码	9
5.4 画图和 gsnDraw	9
5.5 改进框架和 gsnFrame	9
5.6 特殊字符串源代码 gsnLeftString,gsnCenterString,gsnRightString	9
第六章 颜色	10
6.1 开始颜色	10
6.2 默认色表	10
6.3 内置色表	10
6.4 使用 RGB 三色	11
6.5 命名颜色	11
6.6 gsnSpreadColors	12
6.7 CMYK	12
第七章 矢量图	12
7.1 矢量的类型	12
7.2 控制矢量	13
7.3 通过标量场或在标量场给矢量着色	13
第八章 地图标号	15
第九章 页面最大化	15
第十章 等值线图	16
10.1 手动设置等值线阶	16
10.2 等值线效果	16
10.3 明确设置等值线阶	17

10.4 等值线标签	17
第十一章 二维 Lat/Lon 数组	18
11.1 原始网格投影	18
11.2 不规则网格	18
第十二章 改变纵横比	19
第十三章 面板图	19
13.1 脚本样本	19
13.2 定位图形	20
13.3 重要的面板源代码	20
13.4 不同大小的面板图形	21
第十四章 字体高度	22
第十五章 标题	22
第十六章 插图说明	22
第十七章 标签条	22
第十八章 函数编码	22
18.1 上标/下标	23
18.2 回车	23
18.3 希腊/数学字符	23
第十九章 基元	23
19.1 多边形	23
19.2 多义线	24
19.3 Polymarkers	25
第二十章 添加文本	25
第二十一章 x-y 图	26
第二十二章 指明标签	27
附录 A: 命令源代码	28
附录 B: 高级图形接口	34
附录 C: 命名的颜色列表	36
附录 D: 常见错误信息	38
附录 E: 术语	39

第一章 引言

本文描述了如何使用高级图形界面绘制图形。下面一节将介绍一个脚本例子。

1.1 脚本样本

一般情况下，脚本具有下面几个特征：（1）用 **load** 命令下载包含高级图形界面的函数库，按照惯例，这通常写在 **begin** 之前；（2）读取数据；（3）处理数据（可选）；（4）打开工作站；（5）选择颜色表（可选）；（6）创建源变量，各种图形选项作为属性分配给它；（7）调用恰当的图形界面。

```
*****
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
*****
begin
*****
    in = addfile("myfile.nc","r")           ; pointer to file
    t = in->T                                 ; read in data
*****
; create plot
*****
    wks = gsn_open_wks("ps","ce")           ; open ps file
                                           ; choose colormap

    gsn_define_colormap(wks,"BlAqGrYeOrRe")

    res                = True               ; resource varb
    res@cnFillOn        = True               ; turn on color
    res@cnLinesOn       = False              ; no cn lines
    res@cnLevelSpacingF = 0.5                ; cn spacing
    res@gsnSpreadColors = True               ; full colors
    res@lbAutoLabelStride = True             ; nice lb labels

    plot = gsn_csm_contour_map_ce(wks,t,res
    end
```

图形界面的默认行为是画图和提出框架，用户可以改变默认行为。

脚本例子的完整函数库见：

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

第二章 高级绘图界面

NCL 的绘图是基于面向对象的方法。这种方法提供很大的灵活性，但是很单调。为了帮助用户，我们开发了两套高级绘图界面。这两种界面便于可视化操作。由于历史原因，所有的图形界面都以 **gsn_** 开始，

表示“Getting Started with NCL”。

2.1 gsn 一般界面

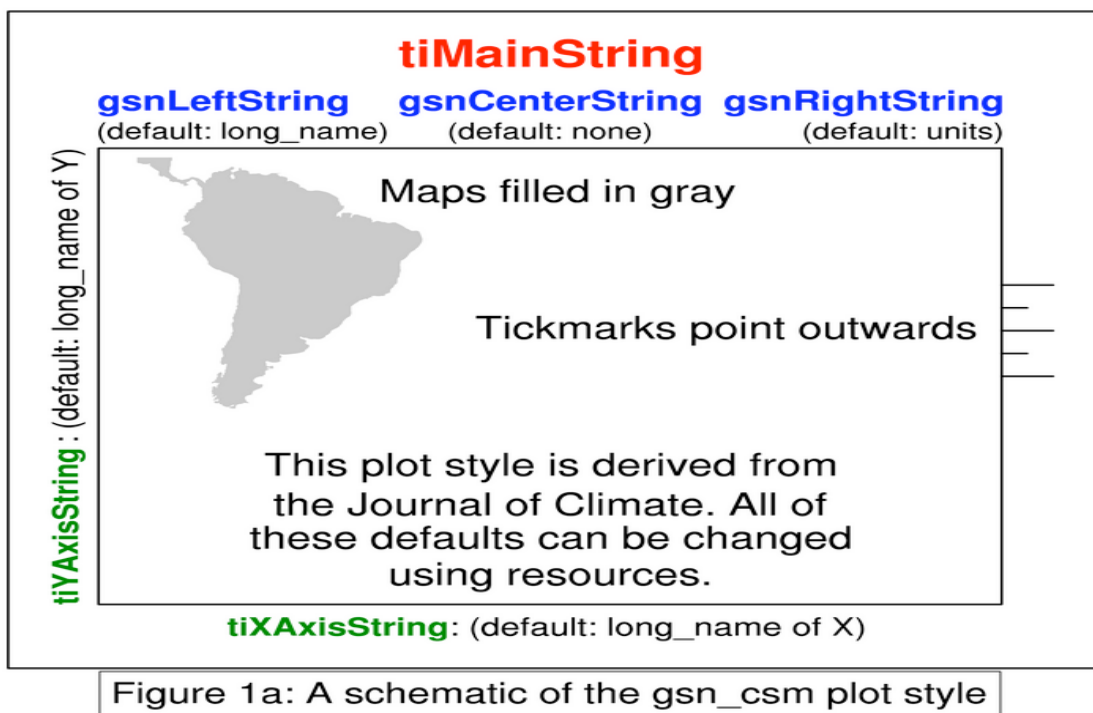
一般界面是函数和程序，它用来绘制基本的 x-y 图，等值线图，流线和矢量图。一般使用默认设置，但是用户可以改变这种设置。附录 B 列出了这些界面。图形例子以及使用指南见：

http://www.ncl.ucar.edu/Document/Manuals/Getting_Started/

2.2 gsn_csm 界面

这些高级界面模仿 *J. of Climate* (June, 1998) 特刊中出现的关于 Climate System Model (CSM) 的图形风格。当 gsn_csm 界面为特定目的而设计时，许多用户喜欢选择它们。原因是它们能自动执行任务，如添加颜色标签，它们也会将变量的 long_name 和单位属性添加到图中（图 1a）。long_name 被放在左上角，单位放在右上角。其它功能包括在圆柱形等距上添加形如 “30N/120E” 的纬度/经度标签，极投影图形，和压力高度图形上的特殊标签。附录 B 列出了这些界面：

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



2.3 我该用什么界面？

通常用户更喜欢选择 gsn_csm 界面，这样做有几个特殊原因：

- 你的数据具有属性，你想图形自动生成标签
- 尽可能为你做同样多的事
- 你喜欢一般风格
- 你想把数据放在地图上，且你的数据具有地球物理坐标

2.4 gsn special 界面

这些接口执行特殊任务，比如绘制标签和文本。不要把它们和 gsn 一般接口混淆。

2.5 下载界面

gsn 一般接口、gsn_csm 图形接口和函数位于两个 NCL 脚本。虽然使用前的任何时候接口可以下载函数库中的函数和程序，但是通常在脚本的上部 begin 语句之前就下载。

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

2.6 gsn_csm 期望

标签:

gsn_csm 图形界面可以通过 CCSM netCDF 惯例获得所需要的信息。如：如果数据具有单位和名称，它将会自动标识在图上。

坐标变量的单位属性:

纬度和经度坐标变量应该具有下面任一个属性:

```
"degrees_north" "degrees_east"  
"degrees-north" "degrees-east"  
"degree_north" "degree_east"  
"degrees north" "degrees east"  
"degrees_N" "degrees_E"  
"Degrees_north" "Degrees_east"
```

如果坐标变量不符合上面的名称，你将会收到错误信息:

```
(0) is_valid_lat_ycoord: Warning: 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.
```

对于不符合的维，最简单的方法是给坐标变量添加属性:

```
x&lat@units = "degrees_north"
```

符号&用来获取坐标变量，@用来获取属性。

第三章 入门指南

3.1 \$NCARG_ROOT

要执行 NCL，你必须设置环境变量。包含 NCARG_ROOT 说明的 UNIX 文件是系统附属物。对于 csh 和 tcsh，可以在 ".cshrc" 文件和 ".login" 文件中设置。在初始化路径变量的文件中设置。如果你用的是

ksh, bsh, bsh, 在文件“.profile”中设置。下面的例子适用于 tcsh, 假定 NCL 位于“/contrib”:

```
setenv NCARG_ROOT /contrib
path = ($NCARG_ROOT/bin $path)
```

NCARG_ROOT 应该设置在包含 ncl 可执行文件的“bin”目录下的父目录里。这根据系统而不同。如果你不确定, 你可以系统管理员安装在什么地方。

3.2 .hluresfile

NCL 有一个默认的图形环境。这要通过 .hluresfile 来完成。执行时, NCL 在用户的主目录下寻找该文件。下面列出该文件最常见的用法:

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

! Color map
*wkColorMap : rainbow+gray

! Font stuff
*Font : helvetica

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

! X11 window size
*wkWidth : 800
*wkHeight: 800
```

将该文件存放在主目录下会产生一个大的 x11 窗口尺寸, 常见的字体, 黑色背景白色前景的图形。 .hluresfile 可以从该网页下载:

<http://www.ncl.ucar.edu/Document/Graphics/>

3.3 运行 NCL

NCL 运行模式有两种: 脚本模式和交互式。后者仅需要按照提示输入 ncl, 交互式需要单独输入每个命令。脚本模式中, 用户可以创建一个单独的 NCL 脚本, 按照下面的方法将脚本发送给编译器:

```
prompt> ncl < script
prompt> ncl [space] script
```

NCL 脚本不需要扩展名, 但是习惯用“.ncl”作为后缀。NCL 脚本必须有 **begin** 和 **end** 语句, **end** 语句后回车。

第四章 工作台

创建图形之前需要打开工作台，它接受图形命令。工作台有一个名称，这个名称称为输出文件的一部分或 X11 窗口的标题。用户可以同时打开多个工作台，每个工作台只有一个颜色地图。

有六种类型的工作台：`ncgm` (NCAR computer graphics metafile), `ps` (postscript), `eps` (encapsulated postscript, contains a bounding box), `epsi` (encapsulated postscript with a

`bitmap preview`), `pdf`, and `X11 window`。例如：

```
wks = gsn_open_wks("pdf", "34_x_45")
wks_2 = gsn_open_wks("ps", "myfile")
```

第五章 通过源代码修改图形

源代码是用来修改默认图形的，它们可以是字符串、浮点型、整型、双精度型等。前两个(或三个)字母小写，剩下的字母第一个大写（如：`cnFill10n`）。如果源代码希望是浮点型数值，源代码名称要加上 `F`（如：`txFontHeightF`, `mpMinLatF`）。

5.1 源代码类型

源代码的前两个字母表示类型：

am	annotation manager	pm	plot manager	vf	vector field(矢量场)
cn	contour(等值线)	pr	primitive (原型)	vc	vectors (矢量)
ca	coordinate arrays (坐标数组)	sf	scalar field (标量场)	vp	viewport (视口)
gs	graphical style (图形风格)	ti	Title (标题)	wk	workstation (工作台)
lb	labelbar (标签条)	tm	Tickmarks (刻度线)	ws	workspace (工作空间)
lg	legend (图例)	tx	Text (文本)	xy	xy plot (xy 图)
mp	maps (地图)	Tr	Transform (转换)		

详见：

<http://www.ncl.ucar.edu/Document/Graphics/Resources/>

5.2 设置源代码

源代码作为属性传递给高级图形接口，用符号@分配属性。注意，`res` 是用户定义的变量。最好给每个源代码建立一个独立的变量，方便传给不同类型的接口。`Vetor` 源代码发送给等值线函数将会出错。下面的例子是建立一个逻辑性的 `res` 变量：

```
res = True
res@tiMainString = "my title"
```



```
res@cnFillOn = True
```

源代码变量是图形接口调用序列参数中的最后一个：

```
plot = gsn_csm_contour(wks,data,res)
```

```
plot2 = gsn_xy(wks,data,res)
```

等号左边的变量是图形类型，名称可以任意定义。

5.3 常用的源代码

附录 A 中给出了常用的源代码。

5.4 画图和 gsnDraw

默认情况下，高级图形接口创建和画出图形对象。设置 `gsnDraw = False` 可以改变默认设置。

5.5 改进框架和 gsnFrame

默认情况下，高级图形界面在画出图形对象后对框架进行改进。框架可以比喻成书的一页。工作台就像一本书，改进框架就等于翻书。工作台可以有多个框架。设置 `gsnFrame = False` 可以改变默认设置。

5.6 特殊字符串源代码

gsnLeftString,gsnCenterString,gsnRightString

gsn_csm图形接口的默认行为是将数据的全称放在图形的左上角，数据单位放在右上角。可以通过 gsnLeftString 和 gsnRightString 来改变默认行为。下面的例子将会删除字符串：

```
res = True
```

```
res@gsnLeftString = ""
```

下面的例子将右边的字符串设置为用户指定的，而且设置了中间字符串：

```
res = True
```

```
res@gsnRightString = "my string"
```

```
res@gsnCenterString = "center"
```

第六章 颜色

6.1 开始颜色

`cnFillOn = True`将会打开存在的等值线填充颜色。另外，`cnFillMode = "RasterFill"`会打开试映图模式。

一个工作台只有一个色表，而不是一个单独的图形。这就意味着在同一个工作台上你不能用不同的色表绘制图形，除非你合并色表。该程序见：

<http://www.ncl.ucar.edu/Applications/color.shtml>

6.2 默认色表

NCL的默认色表由一系列不同的颜色组成，但是用户可以发现这些和科学应用不太相称。有三种方法可以改变色表：选择内置色标表，指明RGB三种颜色，或者指明一个命名的数组。



Figure 5a: Default colormap

6.3 内置色表

NCL中有许多预先定义好的色表，列表可以从下面网站找到：

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

用户可以通过下面程序来选择特定的色表：

`gsn_define_colormap(wks, "gui_default")`

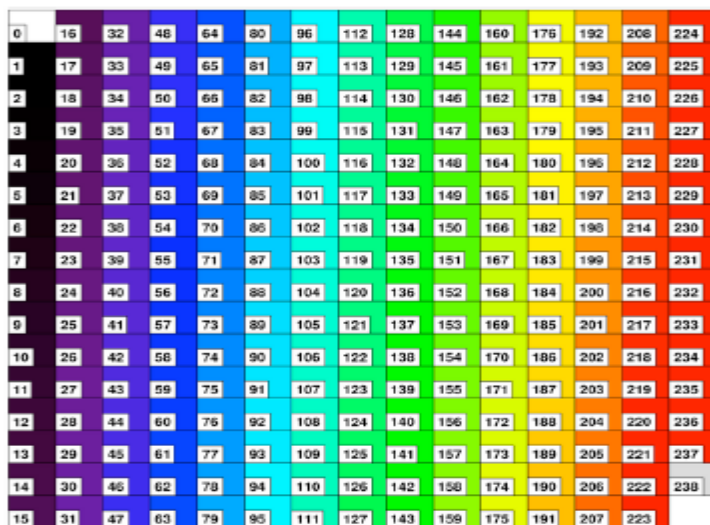


Figure 5b: Example of a built-in colormap (rainbow+gray) showing the index values for each color.

6.4 使用 RGB 三色

用户可以使用RGB (red, green, blue) 三色自定义色表。用下面的代码段来说明：

```
;divide by 225.0 (type float) to
;normalize and convert colors to a
;float variable.
    colors = (/ (/255,255,255/),\
               (/0,0,0/),\
               (/255,255,255/),\
               (/244,255,244/), \
               (/217,255,217/), \
               (/163,255,163/),\
               (/106,255,106/),
               (/43,255,106/),\
               (/255,127,0/) /) /255.0
```

; generate new color map

```
gsn_define_colormap(wks,colors)
```

三色的前两种颜色是黑色和白色，这两种颜色用来表示前景和背景。如果你想创建自己的色表，就必须确保这些颜色存在于前两种颜色中。

6.5 命名颜色

有许多命名的颜色和制定的RGB值相当。详见目录C或：

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

下面的代码段是利用一组命名的颜色创建色表：

```

colors = (/ "white", "black", "white", "RoyalBlue",
           "LightSkyBlue", "PowderBlue", "LightGreen",\
           "PaleGreen", "wheat", "brown", "pink"/)
gsn_define_colormap(wks, colors)

```

6.6 gsnSpreadColors

当创建一个等值线图或矢量图时，NCL选择色表的前N个连续颜色为默认值，这里N是指等值线或矢量阶的个数。假如一个色表包含200个颜色，跨越从深蓝色到深红色。默认行为是使用前N个颜色。设置 `gsnSpreadColors = True` 可以改变默认行为。这个gsn代码迫使使用色表中的全部颜色。如，如果等值线阶为10, 颜色为200，那么使用的颜色是每隔20个。用户也可以通过设置 `gsnSpreadColorStart` 和 `gsnSpreadColorEnd` 来使用色表中的一部分。

6.7 CMYK

一些科学杂志要求提供的图形为CMYK格式。CMYK是一种可选择颜色模式。下面的代码段生成一个CMYK图形：

```

; create variable to hold info
type                = "ps"
type@wkColorModel   = "cmyk"
; pass this to the workstation
wks = gsn_open_wks(type, "color")
; select a colormap
gsn_define_colormap(wks, "BlWhRe")

```

注意，色表不能改成CMYK格式，附言文件在输出时被转换。

第七章 矢量图

7.1 矢量的类型

NCL中矢量有四种类型，设置 `vcGlyphStyle = "type"` 可以改变类型：

```

"LineArrow":    多段线和箭头，默认
"FillArrow":    加填的多边形和箭头
"WindBarb":     使用天气图上的标准风标图示
"CurlyVector": 卷曲矢量

```

7.2 控制矢量

多数矢量图需要三个源代码，它们通过参考矢量控制矢量的大小，细化矢量（图5c）：

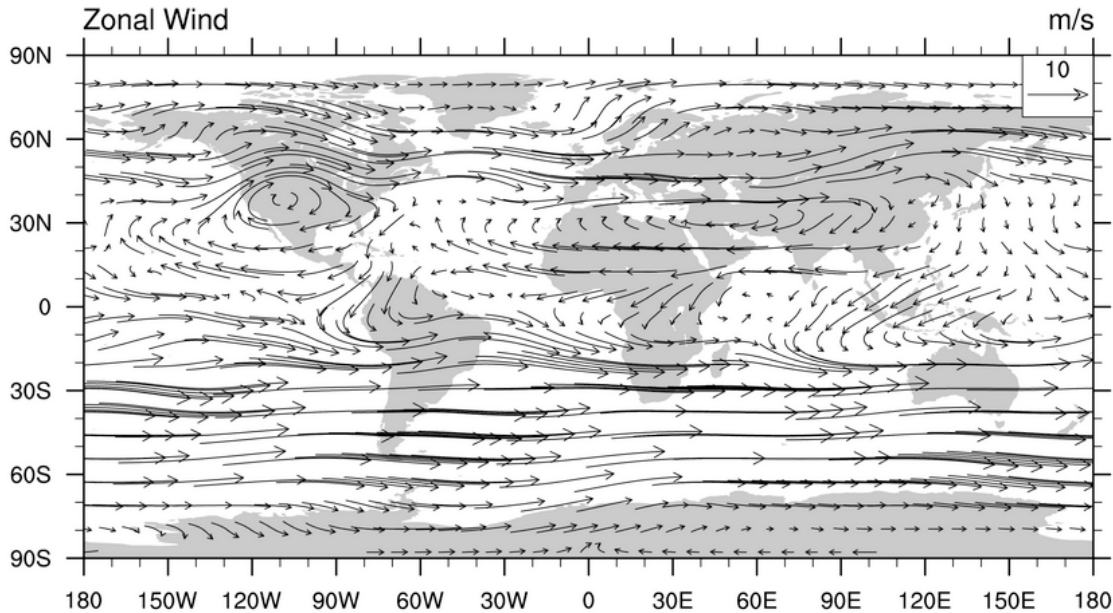


Figure 5c: Example vector plot. Reference vector moved to the upper right corner, curly vectors selected, and the vectors thinned using **vcMinDistanceF**.

下面的代码段说明了源代码的使用：

```
res                                = True
; set the reference vector mag and size

res@vcRefMagnitudeF                = 10.0
res@vcRefLengthF                   = 0.045

; thin the vectors
res@vcMinDistanceF                  = 0.017

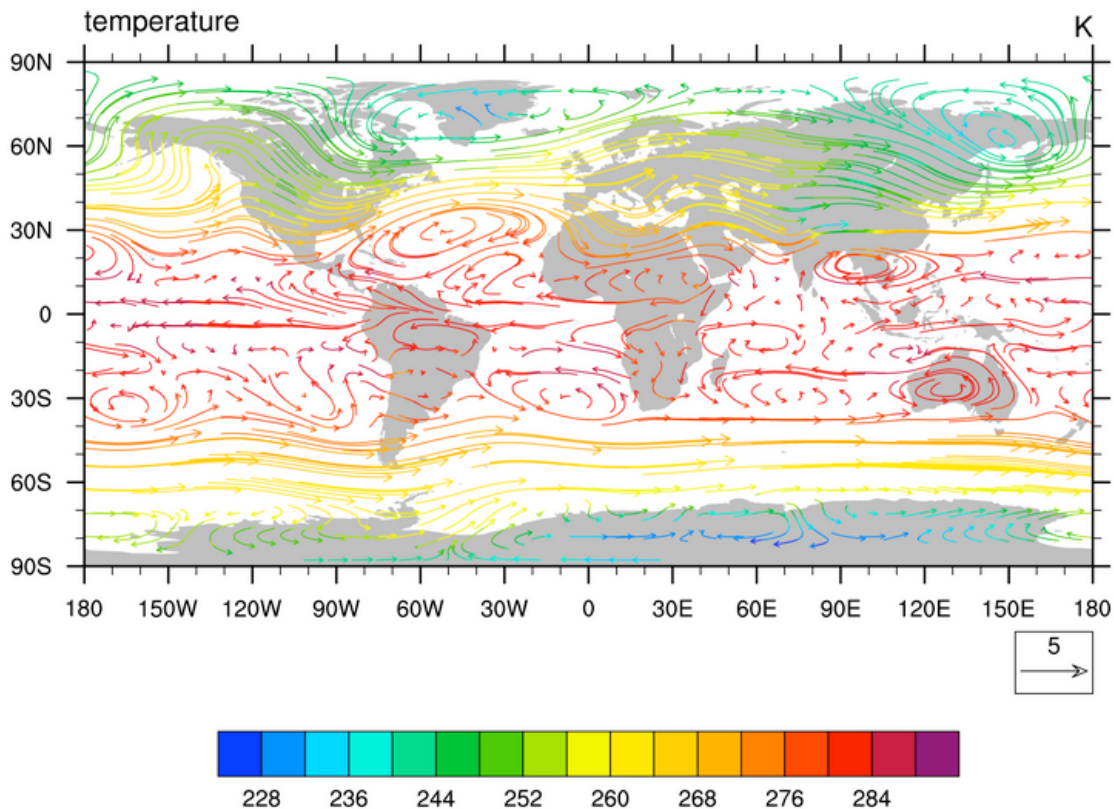
plot = gsn_csm_vector(wks,u,v,res)
```

7.3 通过标量场或在标量场给矢量着色

在同一张图上画矢量场和标量场有四个接口：

- `gsn_csm_vector_scalar_map_ce`
- `gsn_csm_vector_scalar_map_polar`
- `gsn_csm_vector_scalar_map`
- `gsn_csm_pres_hgt_vector`

这些接口的默认情况是通过标量场的大小给矢量定义颜色（图5d），设置 `gsnScalarContour = True` 可以改变默认行为。



如果想要一个不同类型的图形，需要创建单独的等值线图 and 矢量图，然后使用 **overlay** 程序连接两者。

下面是一个生成两个图形对象的脚本。 **Overlay** 用来把它们连接成一个对象。注意 **gsnDraw** 和 **gsnFrame** 设置为 **False**。生成图形的顺序不重要。 **Overlay** 有两个参数，每一个是图形对象。 **Overlay** 程序将第二个对象添加到第一个对象下。 **Overlay** 完成后，需要手动画出连接对象且改进框架。

在一个图形资源列表中，应该关闭 **gsnLeftString** 和 **gsnRightString**。否则两幅图的标签字符串将会重叠。

```
; create vector plot
    res                                = True
    res@vcRefMagnitudeF                 = 30.0
    res@vcRefLengthF                   = 0.045
    res@vcMinDistanceF                 = .019
    res@vcGlyphStyle                    = "CurlyVector"

    res@gsnDraw = False
    res@gsnFrame = False

    res@gsnLeftString = ""
    res@gsnRightString = ""

    plot = gsn_csm_vector(wks,u,v,res)
; create contour plot
    resCN = True
    resCN@cnFillOn = True
```

```

resCN@cnLinesOn = False
resCN@gsnSpreadColors = True

resCN@gsnDraw = False

base = gsn_csm_contour(wks,data,resCN)

```

```

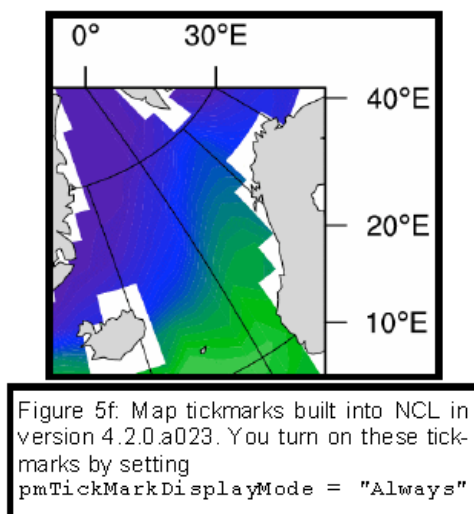
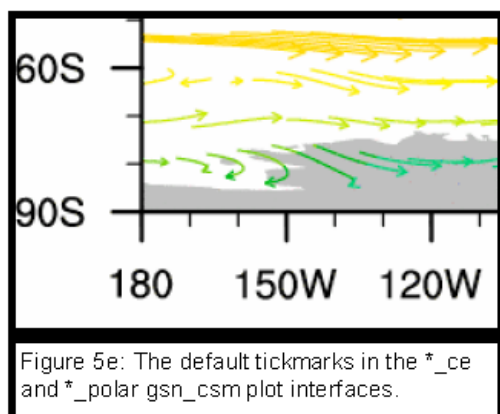
;overlay vector plot onto contour plot
  overlay(base,plot)
  draw(base) ; draw the combined obj
  frame(wks) ; advance the frame

```

成功创建一个`overlay`的关键是要确保两个图形中数据的坐标变量相同。

第八章 地图标号

NCL中有两种地图标号，第一种（图5e）是默认的：`*_ce` and `*_polar`图形接口；第二种（图5f）是4.2.0.a023版本加入的。



第九章 页面最大化

`gsnMaximize`将会自动调整图形大小，且旋转图形使之填满整个页面。注意，如果你要生成一个平板图形（第13章），需要设置变量`gsn_panel`，不是设置每个图形的资源变量。

另一个资源`gsnPaperOrientation`可以设置为"automatic"(default), "landscape"或者

"portrait".

第十章 等值线图

10.1 手动设置等值线阶

需要四个源代码：

```
res@cnLevelSelectionMode      = "ManualLevels"  
res@cnMinLevelValF            = -30  
res@cnMaxLevelValF            = 30  
res@cnLevelSpacingF           = 5
```

10.2 等值线效果

NCL已经开发了许多函数和gsn源代码来创建特殊的等值线效果（图5g）。

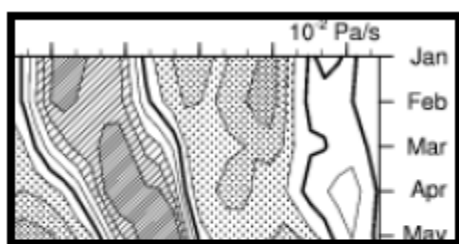


Figure 5g: Example of a special contour effect. This plot uses the resource **gsnZeroLineContourThicknessF** to make the zero line thicker, and then uses various contour fill patterns to shade different areas.

例如，有一个源代码要求你指定零线的宽度（**gsnContourZeroLineThicknessF**），负值用虚线表示（**gsnContourNegLineDashPattern**）。有些函数用来在不同的地方描影（如：**ShadeLtContour**）。所有这些函数都位于特定函数库**shea_util.ncl**，它们和NCL捆绑在一起。脚本库必须在**begin**语句前下载：

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

shea_util.ncl函数的列表见：

http://www.ncl.ucar.edu/Document/Functions/list_shea_util.shtml

下面的一段代码用来调用函数：

```
res                                = True  
res@gsnDraw                        = False
```



```

res@gsnFrame          = False
res@cnFillOn          = True
res@gsnSpreadColors    = True
plot = gsn_csm_hov(wks,chi,res)
plot = ShadeLtContour(plot,0.,3)

```

10.3 明确设置等值线阶

两个代码:

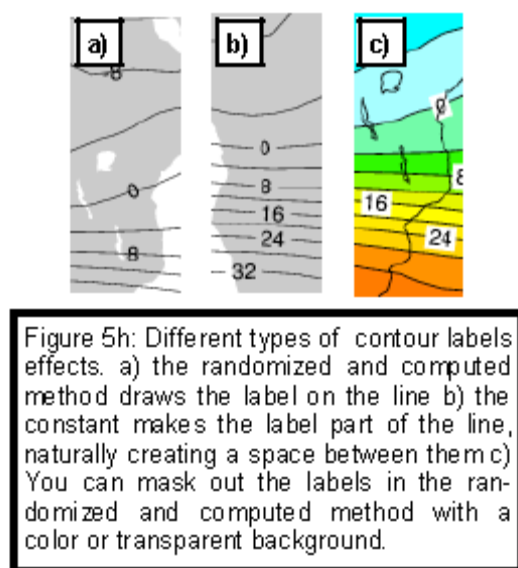
```

res@cnLevelSelectionMode = "Explicit"
res@cnLevels              = (/ .01, 4, 7.2 /)

```

10.4 等值线标签

有三种等值线标签分布模式: randomized (default), computed, and constant. 只有常量模式使标签成为线的一部分 (图5h. b), 其它两种方法穿过线 (图5h. a)。为了避免这种情况, 需要设置 **cnLabelMasking = True**。使用 **cnLineLabelBackgroundColor** 可以选择背景色或透明的背景 (图5h. c)。



如果常量模式有效, 标签的密度由 **cnLineDashSegLenF** 控制; 如果另外两种方法有效, 标签的密度将由 **cnLineLabelDensityF** 控制。

第十一章 二维 Lat/Lon 数组

二维lat/lon数据产生两种类型:

第一种: 数据已经被投影到地球范围内, 称之为原始网格投影。常见例子是原始兰伯特正形投影。

第二种: 来自不规则网格的数据, 它的每个格点必须通过独特的存储单元来表示, 但不是预先投影好的。许多网格属于这一类, 包括曲线网格和有限元网格。

每种情况要求各自特殊的技术。如果你有二维lat/lon数据, 你必须提前知道它是否是原始网格投影。

11.1 原始网格投影

用户怎样知道是否由原始网格投影呢? 一些GRIB和netCDF文件包含“grid type”属性, 它们表明特定的投影。对于没有信息的文件, 用户需要询问数据源来获得更多的信息。

原始网格投影要求的第一种方法是关闭转换数据到地球投影, 设置:

```
res@tfDoNDCOverlay = True
```

第二种方法是利用转角方法限制地图。这种方法要求用户知道网格的左下角和右上角的内角。

```
res@mpLimitMode      = "Corners"
res@mpLeftCornerLatF  = 16
res@mpLeftCornerLonF  = 135
res@mpRightCornerLatF = 54
res@mpRightCornerLonF = 79
```

一些netCDF和GRIB文件包含数组“corners”, 它包含这些信息。

最后一种方法是网格特性。网格本身需要被详细说明, 且要设置定义网格的代码。朗伯特正型投影需要设置两个纬度值一个经度值:

```
res@mpProjection      = "LambertConformal"
res@mpLambertParallel1F = 30.
res@mpLambertParallel2F = 55.
res@mpLambertMeridianF = 45.
```

一些文件包含这些信息:

<http://www.ncl.ucar.edu/Applications/native.shtml>

<http://www.ncl.ucar.edu/Applications/lcnative.shtml>

11.2 不规则网格

lon2d 和 lat2d用来在地图上画出这种类型的图。

假定一个数据文件具有下列特征:

```
time = 1
nlat = 345
nlon = 567
```

```
float TLONG (nlat,nlon)
```

```
float TLAT (nlat,nlon)
float ROFF (time,nlat,nlon)
```

下面一段代码说明了这种方法:

```
tlat      = f->TLAT
tlong     = f->TLONG
roff      = f->ROFF
roff@lon2d = tlat
roff@lat2d = tlon
```

虽然tlat和tlong是变量名,可以任意定,但是属性**lon2d** 和 **lat2d**要保留不能改变。

只有利用高级图形接口绘制二维 lat/lon数据才要明确设置这两个属性。对于等值线图,用户需要用**cnFillMode**源代码改变填充模式。

第十二章 改变纵横比

vpWidthF 和 **vpHeightF**用来改变图形的纵横比。注意,图形是地图时,需要添加源代码
mpShapeMode = "FreeAspect"。

第十三章 面板图

面板图是在同一页上显示两个或多个图形对象。最常用的方法是通过gsn接口 **gsn_panel**。这个接口假定所有图形尺寸相同。它根据第一幅图的大小和形状决定面板的定位。

13.1 脚本样本

下面的代码段说明了基本面板图的方法:

```
plot = new(2,graphic)

res                                     = True
res@cnFillOn                           = True
res@gsnSpreadColors                    = True

res@gsnDraw                             = False
res@gsnFrame                           = False

plot(0) = gsn_csm_contour(wks,u,res)
plot(1) = gsn_csm_contour(wks,v,res)
```

```

;*****
; create panel plot
;*****

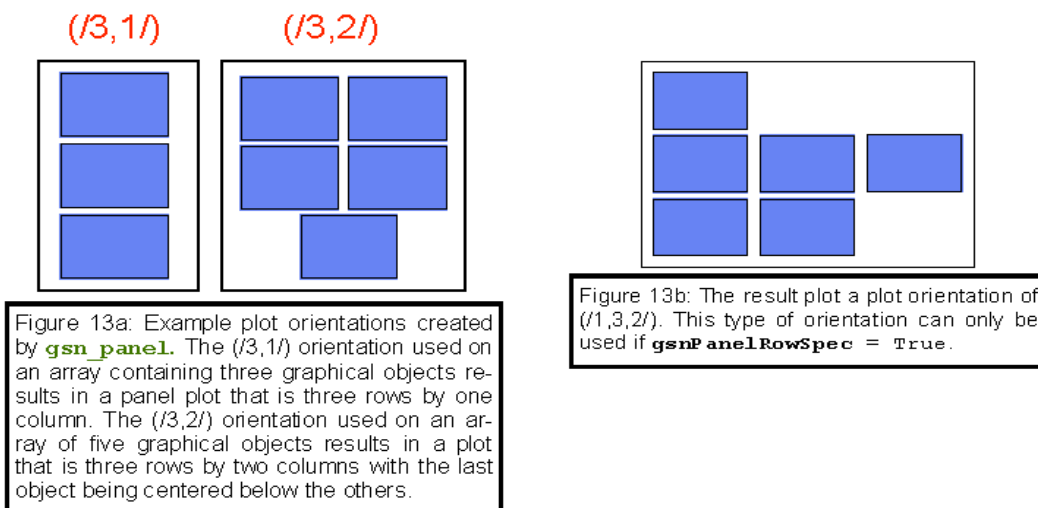
resP          = True
resP@txString = "common title"
gsn_panel(wks,plot,(/3,1/),resP)

```

在这个脚本中，（1）生成一个图形数组，图形是一个包含多个图形对象的变量；（2）将`gsnDraw`设置成`False`，因此生成图形不是独自出现；（3）将`gsnFrame`设置为`False`以便阻止自动生成`frame/page`；（4）生成各个图形对象；（5）生成一个独立的面板代码变量；（6）调用`gsn_panel`。参数是工作台、图形数组、页面上的图形和面板代码变量。

13.2 定位图形

`gsn_panel`中的第三个参数是指明图形应该如何定位的数组。这个数组最简单的指示是指明行和列（图13a）。



设置`gsnPanelRowSpec = True`，图形分布是每行的个数。

13.3 重要的面板源代码

下面的源代码是用来生成面板图形的常用的源代码。它们被传递给`gsn_panel`，而不是给图形数组的每个图形。

- `txString`: 公用标题
- `gsnPanelLabelBar`: 公用标签
- `gsnPanelBottom`: 下面加上空格
- `gsnPanelTop`: 上部加入空格
- `gsnPanelFigureStrings`: 将你选择的字符串放在每个图形的左上角。每个放置字符串的地方能够受控制。

该源代码的用法及效果见：

<http://www.ncl.ucar.edu/Applications/panel.shtml>

13.4 不同大小的面板图形

用户可以通过指定图形的位置和大小来创建包含不同尺寸的图形。下面一段代码说明了该方法（图13a）：

```
; create first plot
res
    = True ; plot mods
res@gsnFrame = False ; don't advance
res@vpXF     = 0.2 ; x location
res@vpYF     = 0.83 ; y location
res@vpWidthF = 0.6 ; width
res@vpHeightF = 0.465 ; height
plot1 = gsn_csm_contour_map_polar(wks,d,res)
; create second plot
sres
    = True
sres@gsnFrame = False
sres@vpXF     = 0.15
sres@vpYF     = 0.3
sres@vpWidthF = 0.7
sres@vpHeightF = 0.18
plot2 = gsn_csm_xy(wks,x,y,sres)
frame(wks)
```

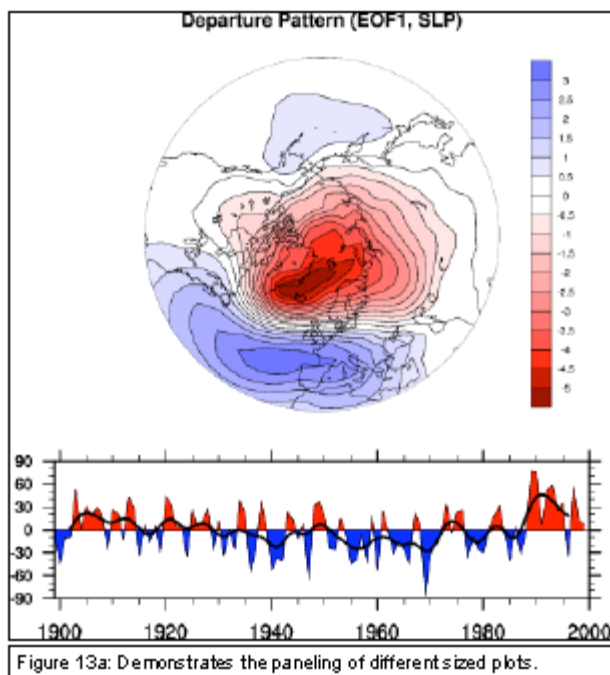


Figure 13a: Demonstrates the paneling of different sized plots.

第十四章 字体高度

NCL有许多标签和标题，每个都由各自的源代码控制。例如，X轴下方的标记字体高度由 `tmXBFontHeightF` 设置，而标签杆的字体高度由 `lbLabelFontHeightF` 设置。附录A中列出了一些常用的字体高度代码。

第十五章 标题

`gsn_csm`高级图形接口有三种主要图形标题和三种附加的标题。主标题由 `tiMainString` 设置，x轴和y轴的标题由 `tiXAxisString` 和 `tiYAxisString` 设置。

第十六章 插图说明

默认情况下，x-y不包括插图说明。设置 `pmLegendDisplayMode = "Always"` 可以开启插图说明。附录A中列出了其他插图说明的源代码。

第十七章 标签条

在 `gsn_csm`高级图形接口中，当用户设置 `cnFillOn = True` 打开颜色时，标签自动生成。标签默认的位置是在图形下面水平放置，标签设在每个颜色的边缘。附录A中包含了能够修改这种默认行为的源代码。对于使用 `gsn_csm`高级图形接口的人，标签条也可以自行创建。具体方法见：

<http://www.ncl.ucar.edu/Applications/labelbar.shtml>

第十八章 函数编码

NCL用函数编码来改变中间文本串的字体，上标、下标等。NCL的默认函数编码是冒号。许多用户喜

欢为字符串保留冒号。你可以在“.hluresfile”文件中改变默认函数编码。这部分的所有例子将会使用函数编码。

18.1 上标/下标

```
"10~S~2~N~x" 102 x
"T~B~K" TK
```

18.2 回车

```
"carriage return~C~here"      carriage return here
```

18.3 希腊/数学字符

```
"~F33~helas~F21~Chars"  ηελασChars
```

第十九章 基元

19.1 多边形

多边形是一个封闭的区域，至少有三个点。最后一个点和第一个点一样才能使多边形封闭。

图形坐标(**gsn_polygon**)，或 page/NDC 坐标 (**gsn_polygon_ndc**)中的图形都可以添加多边形。没有使得多边形成为图形一部分的程序，这就意味着，如果图形是面板形式，多边形不会和图形一起存在。如果想要镶板，要用**gsn_add_polygon**。

下面的代码段用来说明如何在图像上绘制或添加多边形：

```
; plot created above with gsnFrame and gsnDraw set to false.

; add polygon to plot
y = (/30.,30.,0.0,0.,30./)
x = (/ -90., -45., -45., -90., -90./)

resp = True ; mods yes
resp@gsFillColor = "red" ; color

; this technique can not be used with
```

```

; paneling
gsn_polygon(wks,x,y,resp)

; this method CAN be used with paneling. Must be set to dummy variable
d = gsn_add_polyline(wks,plot,x,y,resp)

draw(plot)
frame(wks)

```

观察`gsn_polygon` 和 `gsn_add_polygon`的区别。后者是设置虚拟变量的函数，这个变量不能删除。如果用循环添加多个多边形，就要生成一组虚拟变量。

19.2 多义线

有三种接口会添加/绘制多义线到图形上: `gsn_polyline` (plot coordinates), `gsn_polyline_ndc` (page/NDC coordinates), 和 `gsn_add_polyline`. 下面的代码段说明怎样用多义线绘制盒子到图形上:

```

; plot created above with gsnFrame and gsnDraw set to false.

; add polylines to plot
y = (/30.,30.,0.0,0.,30./)
x = (/ -90.,-45.,-45.,-90.,-90./)

resp                                = True
resp@gsFillColor                    = "red" ; color
resp@gsLineThicknessF               = 2.0

; create array of dummy graphic
; variables. This is required, b/c
; each line must be associated with a
; unique dummy variable.

d = new(4,graphic)

; draw each line separately. Each line must contain two points.
do i=0,3
d(i)= gsn_add_polyline(wks,plot,x(i:i+1),y(i:i+1),resp)
end do

draw(plot)
frame(wks)

```

附录A中有许多控制多义线风格的源代码。

19.3 Polymarkers

添加polymarkers到图形上可以使用16中标记风格(图19a)。可用的三种接口是`gsn_polymarker` (图形坐标), `gsn_polymarker_ndc` (页面坐标), 和 `gsn_add_polymarker`。用户可以用 `NhlNewMarker` 函数创建自己的标记:

<http://www.ncl.ucar.edu/Document/Functions/Built-in/NhlNewMarker.shtml>

16	●	7	△
15	⊗	6	□
14	⊙	5	×
13	☆	4	○
12	☆	3	✱
11	▶	2	+
10	◀	1	•
9	◇	0	✱
8	▽		

Figure 19a: The 17 predefined polymarkers

第二十章 添加文本

有三种高级接口可以添加文本到图形上: `gsn_text` (图形坐标), `gsn_text_ndc` (页面坐标), 和 `gsn_add_text`。只有第二个函数才能在图形上生成文本, 因此它可以和其他图形镶在一起。下面的代码段说明如何在图形上添加文本:

```
; plot created above with gsnDraw and gsnFrame set to false
```

```
add_T = "text here"
x      = 0.5 ; middle of x
y      = 0.85 ; towards top of page

txres = True
txres@txFontHeightF = 0.03

gsn_text_ndc(wks,plot,add_T,x,y,txres)
draw(plot)
```

第二十一章 X-Y 图

下面的代码源说明了利用高级图形接口`gsn_csm_xy`绘制X-Y图:

```
; read in data
f = addfile("./uv300.nc","r")
u = f->U

x = u&lat
y = u
; open workstation
wks = gsn_open_wks("ps","xy")

; create plot
res = True
res@tiMainString = "Basic XY plot"
```

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

为了放置两条以上直线，需要创建一个足够大的数组:

```
; read in data
f = addfile("./uv300.nc","r")
u = f->U

x = u&lat
y = u

; create array to hold multiple lines
data = new(/2,dimsizes(x)/,float)

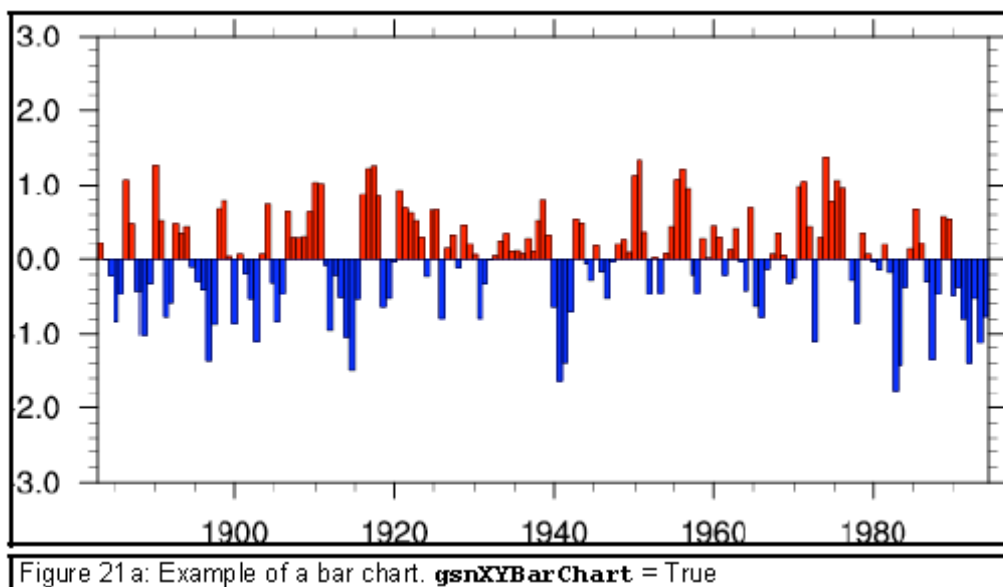
; use coordinate subscripting to select
; two lines from u
data(0,:) = u(0:,{82})
data(1,:) = u(0:,{69})

; open workstation
wks = gsn_open_wks("ps","xy")

; create plot
res = True
res@xyLineThicknesses = (/1.0,2.0/)
res@xyLineColors = (/blue/)
```

```
plot = gsn_csm_xy(wks,y,data,res)
```

附录A中列出了许多修改X-Y图形风格的源代码，包括**gsnXYBarChart**，它可将X-Y图形转变成长条图（图21a）。



其它x-y图形例子见：

<http://www.ncl.ucar.edu/Applications/xy.shtml>

设置**xyMarkLineMode** to "Marker"可以将直线图转变为散点图。

附录A中给出了散点图的源代码。

第二十二章 指明标签

下面的代码段说明了用户用自定义标签代替默认标签的方法：

```
custom_labs = (/"Jan","Feb","Mar"/)
x_values = x&time
```

```
res = True
res@tmXBMode = "Explicit"
res@tmXBValues = x_values
res@tmXBLabels = custom_labs
res@tmLabelAutoStride = True
```

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

分配给**tmXBValues**属性的值必须部分和图形接口决定的值相等。例如，如果X轴是时间，增加量为秒，那么用户不能分配给**tmXBValues**不相关的整数。

附录 A：命令源代码

详尽的代码源见：

<http://www.ncl.ucar.edu/Document/Graphics/Resources/>

Axis - <http://www.ncl.ucar.edu/Document/Graphics/Resources/tr.shtml>

Name	Function	Default	Example
trYReverse trXReverse	reverse x or y axis	False	True
trYMinF trXMinF	set minimum of x or y axis	0.0	3
trYMaxF trXMaxF	set maximum of x or y axis	1.0	900
trYLog trXLog	turn on/off log axis	False	True

Contour - <http://www.ncl.ucar.edu/Document/Graphics/Resources/cn.shtml>

Name	Function	Default	Example
cnFillOn	turn on/off color filled contours	False	True
cnLinesOn	turn on/off contour lines	True	False
cnFillMode	set type of contour fill	"AreaFill"	"RasterFill"
cnLevelSelectionMode	control contour levels	"AutomaticLevels"	"ExplicitLevels" "ManualLevels"
cnMinLevelValF cnMaxLevelValF	set minimum or maximum contour level	dynamic	5 35
cnLevelSpacingF	set contour spacing	dynamic	2
cnLevels	set contour elvels when cnLevelSelectionMode is "ExplicitLevels"	dynamic	(/3,5,7,9,10,45/)
cnLineThicknessF	set thickness of contour lines	1.0	2.0
cnFillPatterns	set pattern fills	"SolidFill"	(/1,3,-1/) (-1 is transparent)
cnInfoLabelOn	turn on/off the contour info label	True	False

Labelbars - <http://www.ncl.ucar.edu/Document/Graphics/Resources/lb.shtml>

Name	Function	Default	Example
cnFillOn	turn contour fill on/off	False	True
cnFillMode	set contour fill mode	"AreaFill"	"RasterFill"
cnLabelBarEndStyle	set style for end labels	"IncludeOuterBoxes"	"ExcludeOuterBoxes"
gsnSpreadColors	span full range of colormap	False	True
gsnSpreadColorStart	begin colormap at particular index	2	46
gsnSpreadColorEnd	end colormap at particular index	<i>ncolors</i> -1	89
lbLabelBarOn	turn on/off the labelbar	True for gsn_csm interfaces	False
lbOrientation	set orientation of labelbar	horizontal for gsn_csm interfaces	"vertical"
lbLabelAutoStride	automatically pick nice labelbar label stride	False	True
lbTitleOn	turn on/off a labelbar title	False	True
lbTitleString	set labelbar title	Null	"m/s"
lbLabelAlignment	st where the labelbar label is oriented wrt to the color boxes	"ExternalEdges"	"BoxCenters"
pmLabelBarOrthogonalPosF	moves the labelbar orthogonally to its position. For a horizontal labelbar, this is up and down.	N/A	-0.03
pmLabelBarParallalPosF	moves the labelbar perpendicularly to its position. For a vertical labelbar, this is left and right.	N/A	-0.01
pmLabelBarWidthF	set the width of the labelbar	set for the user in the gsn_csm interfaces	
pmLabelBarHeightF	set the height of the labelbar	set for the user in the gsn_csm interfaces	

GSN - <http://www.ncl.ucar.edu/Document/Graphics/Resources/gsn.shtml>

Name	Function	Default	Example
gsnAddCyclic	turn on/off the addition of a cyclic point to the longitude coordinate values	True for data that has 1D coordinate variables	False
gsnCenterString	see figure 1a	N/A	"string here"
gsnDraw	draw the plot	True	False
gsnFrame	advanced the frame (page)	True	False
gsnLeftString	see figure 1a	long_name (if exists) in gsn_csm interfaces	"Salinity"
gsnMaximize	maximizes plot and rotates to landscape if necessary	False	True
gsnPanelFigureStrings	add a series of strings to the upper left corner of each plot in a panel	N/A	("/a","b","c/")
gsnPanelLabelBar	turn on/off a common labelbar in a panel plot	False	True
gsnRightString	see figure 1a	units (if exists) in gsn_csm interfaces	"ppm"
gsnScalarContour	force vector/scalar gsn_csm interfaces to draw vectors over the scalar field	False	True
gsnSpreadColors	span full range of colormap	False	True
gsnSpreadColorStart	begin colormap at particular index	2	46
gsnSpreadColorEnd	end colormap at particular index	ncolors-1	89
gsnXYBarChart	changes an x-y line into a bar chart	False	True
gsnXRefLine	add a vertical reference line to a plot	None	1.0
gsnXRefLineColor	change color of X reference line	foreground color	"green"
gsnYRefLine	add a horizontal reference line to a plot	None	0.0
gsnYRefLineColor	change color of Y reference line	foreground color	"blue"

Legends - <http://www.ncl.ucar.edu/Document/Graphics/Resources/lg.shtml>

Name	Function	Default	Example
pmLegendWidthF	set width of a legend	dynamic	0.6
pmLegendHeightF	set height of a legend	dynamic	0.3
lgTitleOn	turn on legend title	False	True
lgTitleString	set title string	N/A	"Profiles"
lgOrientation	set orientation of the legend	"horizontal"	"vertical"
lgPerimOn	turn the legend perimeter on/off	True	False
xyExplicitLegendLabels	change default legend labels	N/A	(/"a","b"/)
pmLegendOrthogonalPosF	adjust the legend orthogonally	N/A	-0.03
pmLegendParallelPosF	adjust the legend perpendicularly	N/A	0.2

XY curves - <http://www.ncl.ucar.edu/Document/Graphics/Resources/xy.shtml>

Name	Function	Default	Example
xyDashPatterns	set line pattern	solid	(/0,2/) (/"solid","dash"/)
xyLineThicknesses	set line thicknesses	1.0	(/2.0,3.0,4.0/)
xyLineColors	set line colors	foreground color	(/"red","blue"/)
xyMarkLineModes	set whether lines contain markers, lines, or both markers and lines	"Lines"	"Lines" "Markers" "MarkLines"
xyMarkers	set marker styles	asterisk	5
xyMarkerColor	set marker colors	foreground color	"green"
xyMarkerSizeF	set marker size	0.01	0.03

Maps - <http://www.ncl.ucar.edu/Document/Graphics/Resources/mp.shtml>

Name	Function	Default	Example
mpLimitMode	determine how a map is zoomed in	depends on projection	"LatLon" "Corners"
mpMinLatF	set minimum latitude for map zoom	dynamic	30.
mpMaxLatF	set maximum latitude for map zoom	dynamic	60.
mpMinLonF	set minimum longitude for map zoom	dynamic	-70.
mpMaxLonF	set maximum longitude for map zoom	dynamic	89.
mpFillOn	turn on/off map fill	True for gsn_csm interfaces	False
mpCenterLonF	set center longitude of projection	0	180.
mpDataBaseVersion	set map database resolution	"LowRes"	"MediumRes" "HighRes" (must be downloaded)
mpLandFillColor	set color of land areas	"gray" for gsn_csm interfaces	"brown"
mpOceanFillColor	set color of ocean areas	"transparent"	"SkyBlue"
mpInlandWaterFillColor	set color of inland water areas	"transparent"	"blue"
mpOutlineOn	turn on/off the map outlines	True	False
mpOutlineBoundarySets	set various continental outlines on and off	"Geophysical"	"Geophysical" "AndUSStates" "National"
mpGeophysicalLineThicknessF	set line thickness of map outlines	1.0	2.0
mpGeophysicalLineColor	set color of map outlines	foreground	"red"
mpUSStateLineColor	set color of US state boundaries	foreground	"blue"

Polygons, polylines, polymarkers -

<http://www.ncl.ucar.edu/Document/Graphics/Resources/gs.shtml>

Name	Function	Default	Example
gsFillColor	set fill color for inside of polygon	transparent	"red"
gsEdgeColor	set color of the outline of a polygon	none	"black"
gsEdgesOn	turn on/off polygon edge	False	True
gsLineColor	set polyline color	foreground color	"orange"
gsLineThicknessF	set polyline thickness	1.0	2.5
gsMarkerIndex	set marker style	asterisk (0)	5
gsMarkerColor	set marker color	foreground color	"purple"
gsMarkerSizeF	set marker size	0.007	0.014

附录 B：高级图形接口

gsn generic interfaces

`gsn_xy gsn_vector_map`
`gsn_y gsn_vector_scalar_map`
`gsn_contour gsn_streamline`
`gsn_contour_map gsn_streamline_map`
`gsn_vector gsn_map`
`gsn_vector_scalar`

下面的代码画出二维数组的等值线图:

```
plot = gsn_contour(wks,data,res)
```

gsn_csm interfaces

下面列表中， `_ce` 代表等距投影。 `_hov` 代表 `hovmuller` 图标。其它的接口名称不需要说明。对于 `gsn` 一般， `gsn_csm` 是返回图形对象的函数。注意：下面许多接口可以分成好几类。

Contour:

`gsn_contour_shade` (客户化等值线填充)
`gsn_csm_contour`
`gsn_csm_contour_map` (选择投影)
`gsn_csm_contour_map_ce`
`gsn_csm_contour_map_polar`
`gsn_csm_contour_map_overlay` (覆盖另外的等值线)

```
plot = gsn_csm_contour(wks,data,res)
```

Streamline:

`gsn_csm_streamline`
`gsn_csm_streamline_map` (选择投影)
`gsn_csm_streamline_map_ce`
`gsn_csm_streamline_map_polar`
`gsn_csm_streamline_contour_map`
`gsn_csm_streamline_contour_map_ce`
`gsn_csm_streamline_contour_map_polar`

```
plot = gsn_csm_streamline(wks,u,v,res)
```

Vector:

`gsn_csm_vector`
`gsn_csm_vector_map`
`gsn_csm_vector_map_ce`
`gsn_csm_vector_scalar_map`
`gsn_csm_vector_scalar_map_ce`
`gsn_csm_vector_scalar_map_polar`

```
plot = gsn_csm_vector(wks,u,v,res)
```

Pressure/Height:

`gsn_csm_pres_hgt`

`gsn_csm_pres_hgt_streamline`
`gsn_csm_pres_hgt_vector`

Misc:

`gsn_csm_lat_time`
`gsn_csm_time_lat`
`gsn_csm_hov`
`gsn_csm_xy`
`gsn_csm_y`

gsn special interfaces

Polylines: 这些接口给图形添加多段线

`gsn_polyline` (图形坐标)
`gsn_polyline_ndc` (页面坐标)
`gsn_add_polyline` (面板,图形坐标)

Polymarkers: 这些接口给图形添加多标记

`gsn_polymarker` (图形坐标)
`gsn_polymarker_ndc` (页面坐标)
`gsn_add_polymarker` (面板, 图形坐标)

Polygons: 这些接口给图形添加多边形

`gsn_polygon` (图形坐标)
`gsn_polygon_ndc` (页面坐标)
`gsn_add_polygon` (面板, 图形坐标)

Text:这些接口给图形添加文本

`gsn_text` (图形坐标)
`gsn_text_ndc` (页面坐标)
`gsn_add_text` (面板, 图形坐标)
`gsn_create_text` (没有坐标, 和`gsn_add_annotation`一起使用)

Colormaps: 这些接口用来手动操作颜色地图。 可以通过下面链接查看内部颜色地图:

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

`gsn_define_colormap` `gsn_merge_colormaps` `gsn_draw_colormap`
`gsn_retrieve_colormap` `gsn_draw_named_colors` `gsn_reverse_colormap`
`hlsrgb` `hsvrgb` `rgbhls`
`rgbhsv` `rgbbyiq` `yiqrghb`

Miscellaneous: 这些接口执行各种函数

`gsn_add_annotation` `gsn_panel`
`gsn_attach_plots` `gsn_table`
`gsn_create_labelbar`
`gsn_create_legend`
`gsn_histogram`
`gsn_labelbar_ndc`
`gsn_legend_ndc`
`gsn_open_wks`

附录 C：命名的颜色列表

255 250 250	snow	30 144 255	dodger blue
248 248 255	ghost white	0 191 255	deep sky blue
245 245 245	white smoke	135 206 235	sky blue
220 220 220	gainsboro	135 206 250	light sky blue
255 250 240	floral white	70 130 180	steel blue
253 245 230	old lace	176 196 222	light steel blue
250 240 230	linen	173 216 230	light blue
250 235 215	antique white	176 224 230	powder blue
255 239 213	papaya whip	175 238 238	pale turquoise
255 235 205	blanched almond	0 206 209	dark turquoise
255 228 196	bisque	72 209 204	medium turquoise
255 218 185	peach puff	64 224 208	turquoise
255 222 173	navajo white	0 255 255	cyan
255 228 181	moccasin	224 255 255	light cyan
255 248 220	cornsilk	95 158 160	cadet blue
255 255 240	ivory	102 205 170	medium aquamarine
255 250 205	lemon chiffon	127 255 212	aquamarine
255 245 238	seashell	0 100 0	dark green
240 255 240	honeydew	85 107 47	dark olive green
245 255 250	mint cream	143 188 143	dark sea green
240 255 255	azure	46 139 87	sea green
240 248 255	alice blue	60 179 113	medium sea green
230 230 250	lavender	32 178 170	light sea green
255 240 245	lavender blush	152 251 152	pale green
255 228 225	misty rose	0 255 127	spring green
255 255 255	white	124 252 0	lawn green
0 0 0	black	0 255 0	green
47 79 79	dark slate gray	127 255 0	chartreuse
105 105 105	dim gray	0 250 154	medium spring green
112 128 144	slate gray	173 255 47	green yellow
119 136 153	light slate gray	50 205 50	lime green
190 190 190	gray	154 205 50	yellow green
211 211 211	light gray	34 139 34	forest green
25 25 112	midnight blue	107 142 35	olive drab
0 0 128	navy blue	189 183 107	dark khaki
100 149 237	cornflower blue	240 230 140	khaki
72 61 139	dark slate blue	238 232 170	pale goldenrod
106 90 205	slate blue	250 250 210	light goldenrod yellow
123 104 238	medium slate blue	255 255 224	light yellow
132 112 255	light slate blue	255 255 0	yellow
0 0 205	medium blue	255 215 0	gold
65 105 225	royal blue	238 221 130	light goldenrod
0 0 255	blue	218 165 32	goldenrod 1

84 134 11	dark goldenrod	255 69 0	orange red
188 143 143	rosy brown	255 0 0	red
205 92 92	indian red	255 105 180	hot pink
139 69 19	saddle brown	255 20 147	deep pink
160 82 45	sienna	255 192 203	pink
205 133 63	peru	255 182 193	light pink
222 184 135	burlywood	219 112 147	pale violet red
245 245 220	beige	176 48 96	maroon
245 222 179	wheat	199 21 133	medium violet red
244 164 96	sandy brown	208 32 144	violet red
210 180 140	tan	255 0 255	magenta
210 105 30	chocolate	238 130 238	violet
178 34 34	firebrick	221 160 221	plum
165 42 42	brown	218 112 214	orchid
233 150 122	dark salmon	186 85 211	medium orchid
250 128 114	salmon	153 50 204	dark orchid
255 160 122	light salmon	148 0 211	dark violet
255 165 0	orange	138 43 226	blue violet
255 140 0	dark orange	160 32 240	purple
255 127 80	coral	147 112 219	medium purple
240 128 128	light coral	216 191 216	thistle
255 99 71	tomato		

用户可以通过下面的代码画出命名的颜色的测试:

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

begin
  wks = gsn_open_wks("x11","gsn_draw_named_colors")
  colors = (/ "white", "black", "PeachPuff", "MintCream", "SlateBlue", \
    "Khaki", "OliveDrab", "BurlyWood", "LightSalmon", "Coral", \
    "HotPink", "LemonChiffon", "AliceBlue", "LightGrey", \
    "MediumTurquoise", "DarkSeaGreen", "Peru", "Tomato", \
    "Orchid", "PapayaWhip" / )

  rows = 4
  cols = 5
  gsn_draw_named_colors(wks,colors,(/rows,cols/)) ; Draw these
                                                    ; named colors.
end
```

附录 D：常见错误信息

- 1、(0) check_for_y_lat_coord: Warning: Data either does not contain a valid latitude coordinate array or doesn't contain one at all
(0) check_for_lon_coord: Warning: Data either does not contain a valid longitude coordinate array or doesn't contain one at all

解决方法：根据2.6所列内容重新命名x、y的维。

- 2、(0) is_valid_lat_ycoord: Warning: 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.
(0) is_valid_lat_xcoord: Warning: The units attribute of the X coordinate array is not set to one of the allowable units values (i.e. 'degrees_east'). Your longitude labels may not be correct.

解决方法：添加或改变纬度/经度坐标数组的单位属性（2.6）。

- 3、(0) gsn_add_cyclic: Warning: The range of your longitude data is not 360. You may want to set gsnAddCyclic to False to avoid a warning message from the Spline function.

解决方法：在所有gsn_csm高级地图接口上，循环指针被添加到数据里。如果不相称（如局部图形），需要设置gsnAddCyclic = False。

- 4、(0) warning:_NhlCreateSplineCoordApprox: Attempt to create spline approximation for Y axis failed: consider adjusting trYTensionF value
warning:IrTransInitialize: error creating spline approximation for trYCoordPoints; defaulting to linear

解决方法：这可能是由于经度坐标变量有错误而产生的。有两种可能性：1) 不正确；2) 数据有裂口。

- 5、fatal:ContourPlotDraw: Workspace reallocation would exceed maximum size 16777216 fatal:ContourPlotDraw: draw error fatal:PlotManagerDraw: error in plot draw fatal:_NhlPlotManagerDraw: Draw error

解决方法：你的数据太大，NCL默认大小是16MB。你必须扩大其大小：

```
setvalues NhlGetWorkspaceObjectId()  
          "wsMaximumSize": 33554432  
end setvalues
```

附录 E：术语

attribute: 用@将任何类型的数据分配给NCL变量。变量的属性可以包含描述性信息。属性用来为gsn和gsn_csm图形函数设置图形选项。

color index: 一个代表当前色表中的指数的整数值。0是背景颜色，1是前景颜色。颜色指数值可以用定义图形颜色的任何图形源代码一起使用，看“named color”。

coordinate variable: 和命名的维变量或文件变量联系在一起的变量，它包含许多坐标信息。坐标变量必须是单独的值。它们通过一套gsn_csm图形脚本来定义X、Y轴的值。

named color: 代表预先定义的颜色。命名的颜色可以被预先定义图形颜色属性的图形源代码使用。要使用命名的颜色，颜色必须是当前色表的一部分。看“color index”。

named dimension: 用!分配的维变量或文件变量。

NDC coordinates: (标准化的页面坐标) 页面左下角是(0,0)，右下角是(0,1)，左上角是(1, 0)，右上角是(1,1)。NDC中有几个特殊的接口函数。

panel/paneling: 在一页上放置多个图形。注意，gsn_panel中图形尺寸相同。不同尺寸的图形可以利用视口源代码vpXF 和vpYF来手动放置图形（详见13章）。

plot coordinates: 和NDC坐标表示页面且不改变不同，图形坐标是图形产物。一个地图图形有经度和纬度坐标。一个时间序列坐标有一个时间坐标和一个其它任何坐标。图形坐标有许多接口（如：

gsn_polyline, gsn_add_text）。

resources: 改变图形默认行为的值。资源的前两个字母小写，告诉用户是什么类型。剩下的部分描述做什么（如：cnFillOn打开等值线图形的颜色，txFontHeightF设置文本字体高度）。

viewport: 视口是NDC的矩形子区域，它指明在哪里画图形对象。视口准确含义取决于图形对象。例如，xy图形对象，视口指明包含曲线的网格放在哪里，标签写在视口外部。另外，对于文本对象，视口是一个围绕在文本周围的矩形。

workstation: 一个输出设备，如XWindow 系统显示器，ps文件，PDF文件或NCGM。