

# GMT 之路

1.0

Yishuang  
shuangyi.geo@gmail.com

2013-12-31

# 目 录

前 言 .....	3
1. 图一 (pscoast) .....	4
目标.....	4
代码.....	4
结果.....	4
解释.....	4
2. 图二, 加入各种边界与填充 (pscoast) .....	5
目标.....	5
代码.....	5
结果.....	6
解释.....	6
3. 图三, 在地图上标注城市 (psxy) .....	7
目标.....	7
准备知识: postscript 文件的构成与叠加原理 .....	7
准备文件.....	8
代码.....	8
结果.....	9
解释.....	9
4. 图四, 加上地名 (pstext) .....	10
目标.....	10
准备文件.....	10
代码.....	10
结果.....	11
解释.....	11
5. 图五, 画子图 (-X, -Y) .....	12
目标.....	12
准备知识.....	12
1) 纸张大小.....	12
2) 两种坐标.....	13
代码.....	13
结果.....	14
解释.....	14
6. 图六, 二维图 (makecpt, grdimage, psscale) .....	15
目标.....	15
代码.....	16
结果.....	16
解释.....	16
附 录 .....	19
A. 安装.....	19
Windows 7.....	19
Linux.....	20

学会排错.....	22
B. 设定 (gmtset) .....	23
C. 画笔.....	24
1) 尺寸.....	24
2) 颜色.....	24
3) 形状.....	25
4) 画笔属性 (-W) .....	25
5) 填充属性 (-G) .....	26
D. 文字格式.....	26
E. ASCII 与 grid 文件 .....	27

# 前言

每样东西都有其优美的逻辑。

GMT处理的是投影与国界海岸线有关的图。如果与这两个领域无关，则不是GMT的特长，完全可以用其他软件替代。

赶在2013年结束之前写出了这本中文教程。我对计算机很感兴趣，大学、研究生在上面浪费了大量时间，换来了一些似乎没什么用的经验。现在看到很多后继者依然折腾个不停，我想他们缺少的就是这些经验。希望这本书能节约他们的时间。

我想尽量用简单的例子来讲清楚。本文面向的读者需要有一定的Linux基础，本书是基于Linux环境的bash脚本。但对bash脚本的技巧是尽量避免，只有一个例子谈到。所有的代码可以在bash命令环境中一行行敲入，等效于写成shell脚本再执行。但我强烈建议你掌握bash脚本，而不要使用CMD环境，前者强大太多。在Windows下cygwin是个不错的bash模拟器。

这个版本会介绍最基本的6个命令：pscoast, makecpt, grdimage, psscale, psxy, pstext。熟悉这6个命令你基本就能使用GMT画彩图，以及在彩图上添加记号，以及画多图。附录中有GMT的安装指南，以及一些属性设定的参考，还涉及数据格式转换的教程。

版权归作者Yishuang所有。附录中的部分图片摘自GMT\_Docs.pdf。本书在保证原始性、完整性与非盈利性条件下，可以自由传播。对本书的内容建议以及内容错误问题，欢迎电邮作者。

参考文献与网站：

- GMT网站: <http://gmt.soest.hawaii.edu>，可以下载相关程序与说明文档。安装路径doc下也可以找到说明文档。
- GMT\_Docs.pdf：包括了GMT介绍、环境与属性设定、常用命令、示例等，适合入门与常用查询。
- GMT\_manpages.pdf：GMT命令的详细操作手册。适合熟练后参考。

# 1. 图一，由单个命令生成海岸线图（pscoast）

## 目标

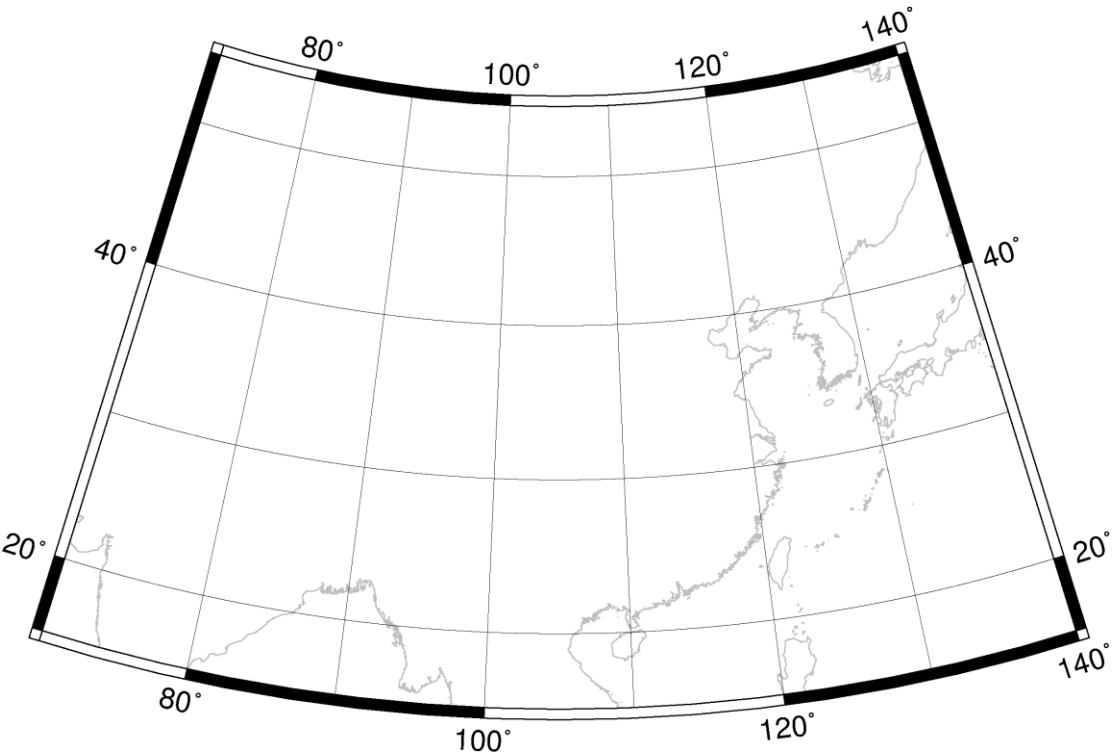
画出一张带海岸线的中国区域地图。

## 代码

```
pscoast -R70/140/15/55 -JB105/30/20/40/15c -Ba20g10 -W1/0.5p,gray >
image01.ps
```

直接在命令行中敲入便可以生成image01.ps文件。

## 结果



## 解释

蓝色是引导符，表明后面的值所定义的参数对象。

pscoast	画海岸线命令
-R70/140/15/55	作图范围，分别代表：左经度/右经度/下纬度/上纬度
-JB105/30/20/40/20c	投影方式，具体参阅另一篇教程《GMT地图投影的研究》，可以直接把需要的

	投影方式复制过来。如需要画日本区域， 查阅得 “C137.5/0/4.5i” ，用这个值替换这里的 “B105/30/20/40/20c” 就可以。注意此时-R的取值要相应的改为日本所在经纬度范围。
-Ba20g10	经纬度标注的间隔，图中标出经度的 “80°，100°” 的间隔为20，即 “a20” 的意义。网格线的间隔为10，即 “g10” 的意义
-W1/0.5p,gray	W1表示海岸线，后面是画笔属性，这里表示宽度为 “0.5p” ，颜色为灰色
> image01.ps	固定用法， ‘>’ 表示 ‘新建’ ，即生成名为image01.ps的文件

## 2. 图二，加入各种边界与填充 ( pscoast )

### 目标

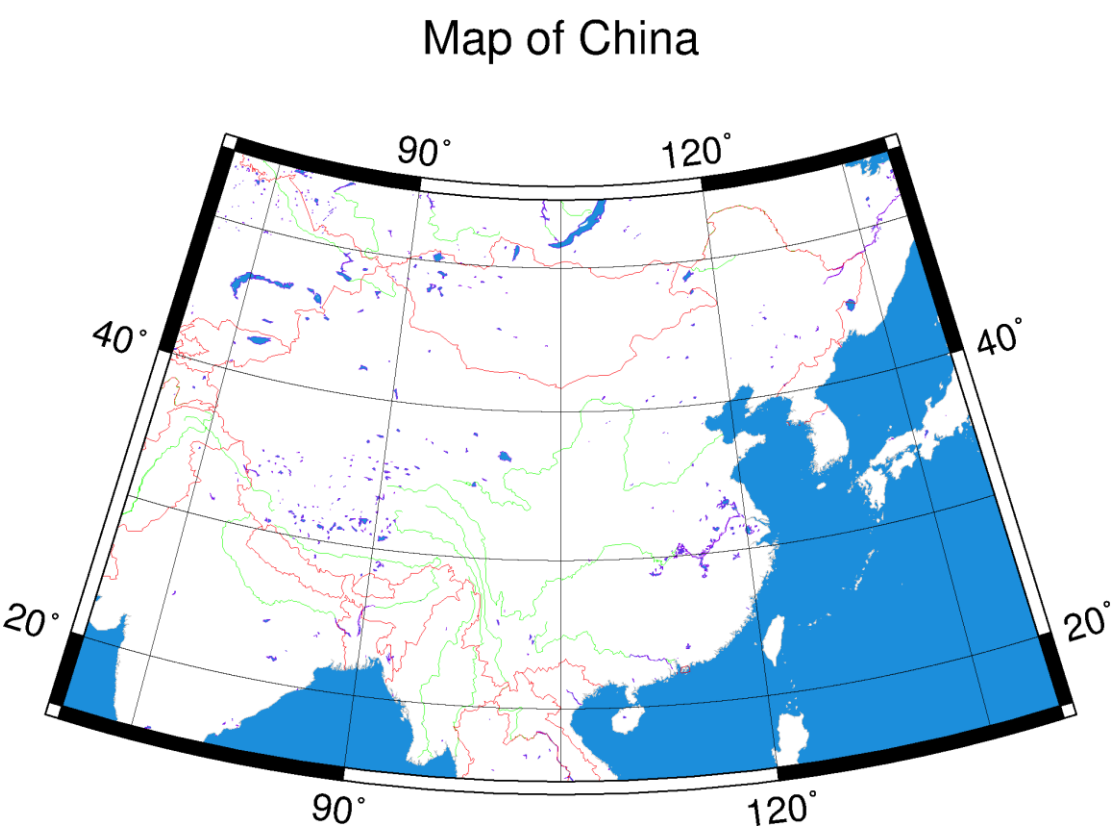
第一幅图太简单了，我想做点这样的改变：

1. 加上红色的国界
2. 加上绿色的主干河流
3. 加上紫色的内陆湖
4. 海洋区域填充为蓝色，陆地区域填充为白色
5. 经度的标注间隔改为30，网格线改为15；纬度的不变；加上标题 “Map of China”

### 代码

```
pscoast -R70/140/15/55 -JB105/30/20/40/15c -B"a30g15/a20g10:.Map of
China:" -S#1BA1E2 -Gwhite -W1/0.2p,gray -W2/0.2p,purple -I1/0.2p,green
-N1/0.2p,red > image02.ps
```

结果



解释

依次解释上面的5个功能是怎么实现的：

1	<code>-N1/0.5p,red</code>	注意-W（水陆分界）、-N（政治分界）、-I（河流）三个开关都支持这种级别设定（一个数字加斜线）。这里1政治分界表示国界，后面接的是画笔属性，即用0.5p宽的、红色的笔画线。
2	<code>-I1/0.2p,green</code>	设定1级河流
3	<code>-W2/0.2p,purple</code>	设定2级水陆分界，即湖边界
4	<code>-S#1BA1E2 -Gwhite</code>	-S接的是海洋颜色，-G为陆地颜色。怎么记呢？有次师弟问我GMT是不是中国人编的，我问咋了。他说S不就是“湿”，G不就是“干”吗.....你这么有才你妈妈知道吗？
5	<code>-Ba30g15/a20g10::</code> <code>"Map of China":</code>	斜线隔开，表示分别对经纬/纬度进行设置。 两个冒号+一个点号引导，即表示中间的内容为标题。注意标题内容存在空格的话需要用引号引起来。

### 3. 图三，在地图上标注城市（psxy）

#### 目标

1. 将北京、武汉、上海用红色的虚线连接起来
2. 将广州用半径为1cm的红色圆圈起来
3. 尝试更丰富的画笔属性变化

#### 准备知识：postscript 文件的构成与叠加原理

在画第三幅图之前，我们需要了解一下postscript文件(ps文件)的构成。postscript文件的开头有一个文件头，结尾部分有一个文件尾，中间部分可以任意叠加。原理就像你把好几张玻璃纸叠加起来一样，后生成的图像不透明部分会遮盖前面生成的，透明的部分则会透出前面生成的。而每一个GMT画图命令就等价于生成一张玻璃纸。GMT命令默认会自动生成文件头又自动生成文件尾，比如上面的例子都是一个命令（pscoast）。

我们的第三幅图需要三个命令：第一个psxy读入北京、上海、武汉的地理坐标，并把这三个城市连起来；第二个psxy把广州标注一个圆圈；最后用pscoast加入海岸线与国界。这三个命令怎样共同合作生成一张图？单纯的三个命令会有三个文件头与文件尾，这样不符合上面的标准。所以这三个命令依次生成时，我们需要这样的规则：第一个psxy画图时生成文件头，但是不要生成文件尾；第二个psxy画图时文件头与文件尾都不要生成；pscoast画图时不要生成文件头，但是生成文件尾。这样，便满足了ps文件的规则。用命令的方式解释上面的规则，就是：

```
psxy ... -K > image03.ps
psxy ... -K -O >> image03.ps
pscoast ... -O >> image03.ps
```

冒号表示省略的命令，可以随意改变。这里写出的模板却是始终不变的。这里，‘-K’就表示“我后面还有人呢别写文件尾”，‘-O’表示“我跟着别人来的别写文件头”；‘>’与



'>>' 是管道命令，前者表示“我要一个干净的文件，没有就新建，有了就删掉再新建”，后者表示“把我的东西追加到这个文件的后面，当然如果实在没有那就新建吧”。

如果有 4 个命令，假设为 COMMAND1，COMMAND2，COMMAND3，COMMAND4，就是：

```
COMMAND1 ... -K > file.ps  
COMMAND2 ... -K -O >> file.ps  
COMMAND3 ... -K -O >> file.ps  
COMMAND4 ... -O >> file.ps
```

所以除了开头与结尾，中间的都是一样的，可以像“玻璃纸”一样任意叠加。

## 准备文件

文件1，北京武汉上海三地的坐标，BjWhSh.txt：

```
116 40  
114 30  
121 31
```

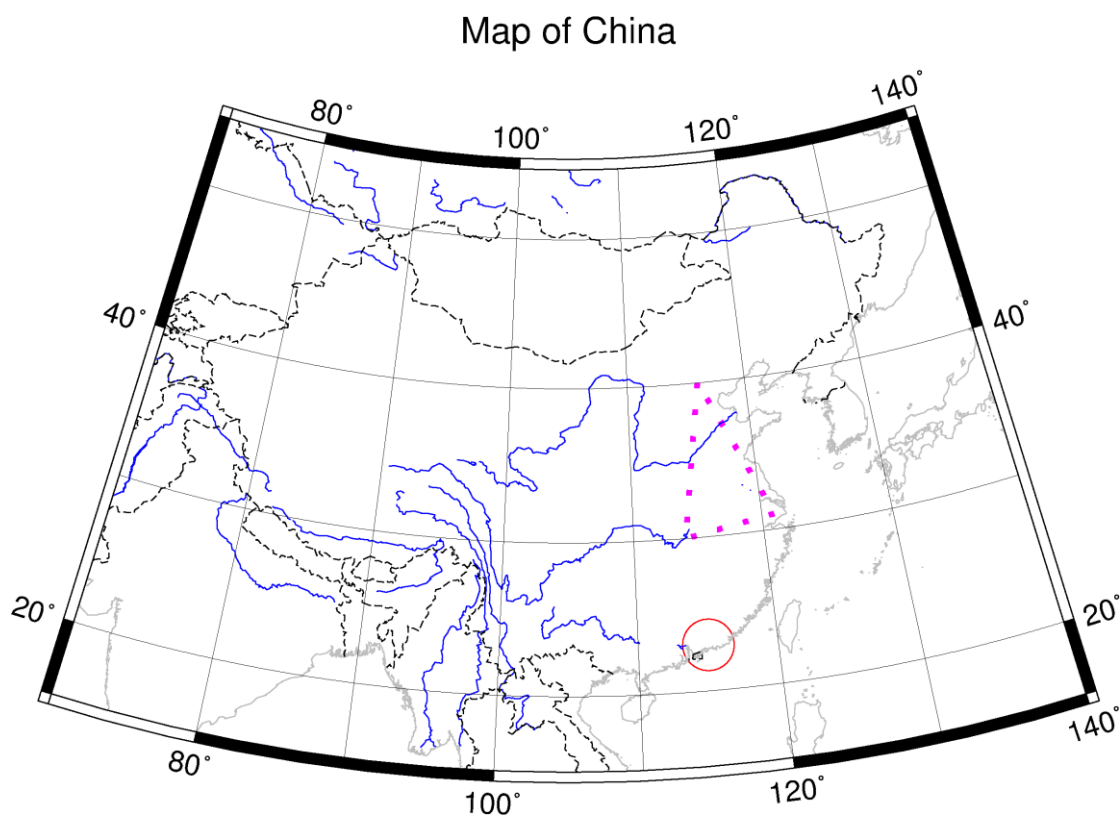
文件2，广州的坐标：Gz.txt：

```
115 23
```

## 代码

```
psxy BjWhSh.txt -R70/140/15/55 -JB105/30/20/40/20c -B"a20g10:.Map of  
China:" -L -Wfat,255/0/255,. -K > image03.ps  
psxy Gz.txt -R -J -B -Sc1c -Wthinner,red -K -O >> image03.ps  
pscoast -R -J -B -W1/0.5p,gray -N1/0.5p,black,-- -I1/0.5p,blue -O >>  
image03.ps
```

结果



解释

■ 第一个 psxy

画连接北京、武汉、上海的虚线

BjWhSh.txt	输入文件
-R70/140/15/55	范围
-JB105/30/20/40/20c	投影
-B"a20g10:.Map of China:"	坐标标注与标题，注意标题中带空格时必须要用引号
-L	封闭的曲线，可删掉此开关后对比变化
-Wfat,255/0/255,.	画曲线的类型，'fat' 对应宽度，'255/0/255' 是rgb颜色，'.' 表示虚线。具体参照附录 '画笔' 章节

■ 第二个 psxy

把广州用一个红圈圈起来

Gz.txt	输入文件
-R -J -B	这些开关只有引导符没有值，表示沿用上面已经定义的，或者缺省值。注意尽管这样依然不能省略。
-Sc1c	第一个c表示圆圈，1c表示圆圈半径为1厘米。
-Wthinner,red	画笔属性

## ■ pscoast

-W1，画海岸线；-N1，画国界；-I1，画主干河流。注意长江珠江的入海处没有画出来，

因为在GMT这些部分不归类于主干河流（-I1），而是属于内陆湖岸（-W2）。

## 4. 图四，加上地名（pstext）

### 目标

把上面四座城市的名称加上去，并尝试不同字体、字号与文字旋转。

### 准备文件

四个城市坐标与名称，4names.txt：

```
116 40 22 0 0 TR Beijing
114 30 24 90 4 MC Wuhan
121 31 26 45 10 BL Shanghai
115 23 22 180 15 TL Guangzhou
```

一共有7列，1、2列是经纬度，第7列是字符内容；第3~6列依次是，字体大小（单位p），

旋转角度（从水平方向逆时针），字体编号（‘pstext -L’ 查看所有字体编号），对齐方式（参照本章节的‘解释’部分）。

### 代码

```
psxy BjWhSh.txt -R70/140/15/55 -JB105/30/20/40/20c -B"a20g10:.Map of
China:" -L -Wfat,255/0/255,. -K > image04.ps
psxy Gz.txt -R -J -B -Sc1c -Wthinner,red -K -O >> image04.ps
```

```
pstext 4names.txt -R -J -B -Gblack -Wgray -S0.6p,red -K -O >> image04.ps
psxy 4names.txt -R -J -B -Sc0.3c -W0.3p,black -Gred -K -O >> image04.ps
pscoast -R -J -B -W1/0.5p,gray -N1/0.5p,black,- -I1/0.5p,blue -O >>
image04.ps
```

结果



解释

一共有5行命令 ,其中1、2、5是直接来自上一幅图 ,这里只是插入了3( pstext )、4( psxy )行。如果你掌握了上面讲的叠加原理 ,这里的应该很容易理解。

■ pstext

4names.txt	输入文件
-R -J -B	继承上面的设定
-Gblack	字体颜色 , 黑色
-Wgray	字体框填充 , 灰色
-S0.6p,red	文字描边 , 0.6p宽度 , 红色

## ■ psxy

4names.txt	这个文件有7列，而前面的psxy例子给的文件只有2列。当你指定了符号类型 ( -Sc0.3c ) 时，会自动只读取前两列。
-Gred	圆圈填充为红色

添加这个命令是为了清晰的显示4names.txt第6列的‘对齐方式’作用。红点为原点，也是旋转中轴点。分垂直方向T(op)、M(idle)、B(ottom)与水平方向 L(ef t)、C(enter)、R(ight)共6种对齐。‘TR’即为右上角对其（即小红点所在位置）。

## 5. 图五，画子图(-X,-Y)

### 目标

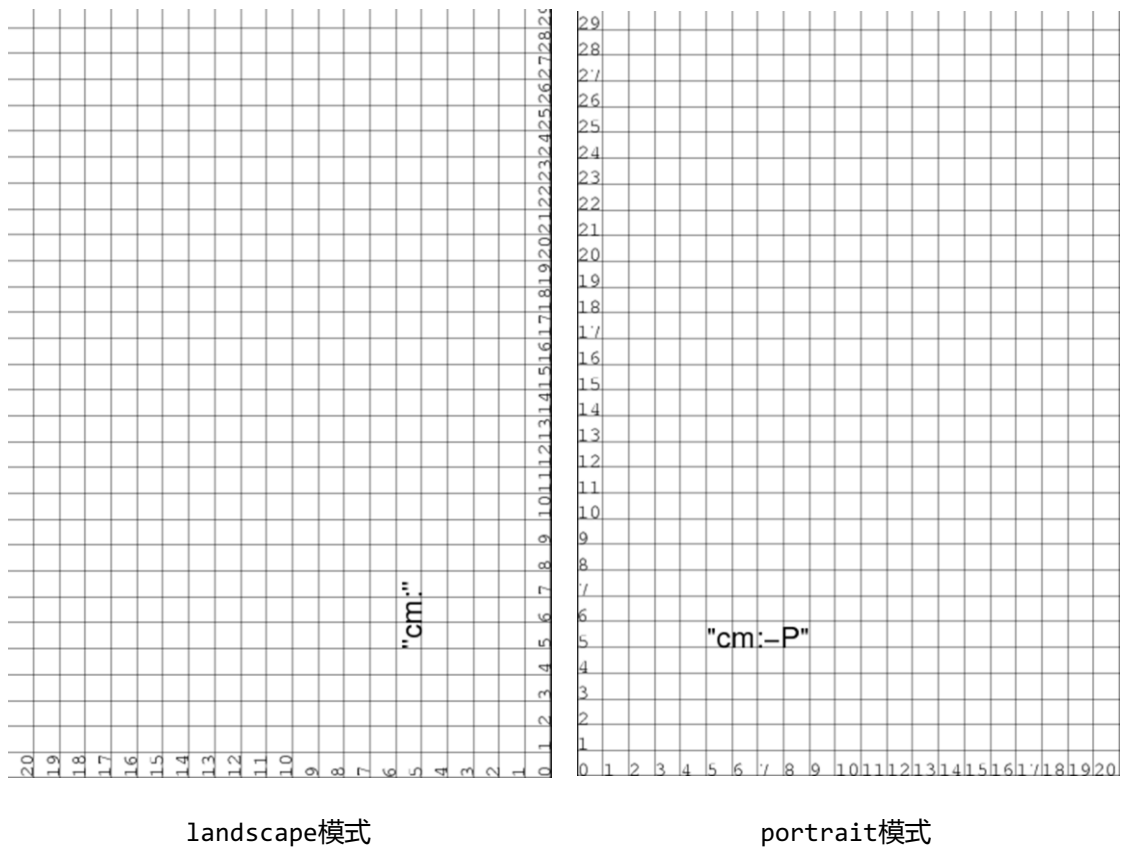
将4幅地图按照2行2列的排列组成一幅新图。

### 准备知识

#### 1) 纸张大小

如果需要在一张纸上画两幅并列的图，怎么运用上面的叠加原理？直接画两幅图的话会完全重叠以至于覆盖前面的内容，我们需要使用平移开关使两幅图错开一定距离。

首先需要了解画图纸张。默认是A4纸（21cm\*29.7cm），默认画图的起点是右下角，向上x轴，向左y轴（landscape模式），即风景照一般采用的构图，适合宽的图；如果加上‘-P’开关，左下为起点，向右x轴，向上y轴。变成了‘portrait模式’，即肖像画一般采用的构图，适合高的图。注意图中字符的方向，以及坐标的增加方向。



## 2) 两种坐标

第一种是与投影相关的坐标，如上面的4个城市经纬度，投影方式不同，在图中的位置也不同。

第二种是与纸张相关的绝对坐标，即上面的网格线显示的坐标。如果要平移图形，就要在这种坐标下指定。

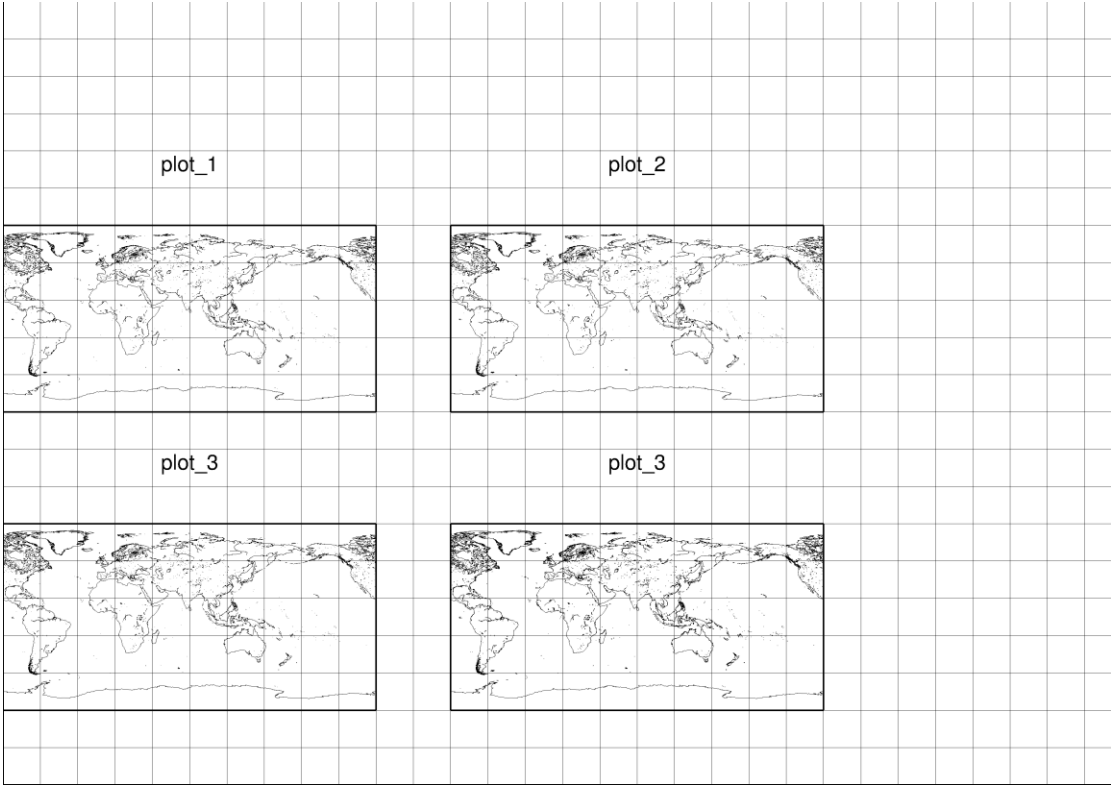
## 代码

```
#!/bin/bash
gmtset HEADER_FONT_SIZE=16p
gmtset BASEMAP_TYPE=plain
psbasemap -R0/100/0/100 -Jx1c -Bg1 -X0c -Y0c -K > image05.ps
pscoast -R -JQ80/10c -B0:."plot_1": -W1 -Y10c -K -O >> image05.ps
pscoast -R -J -B0:."plot_2": -W1 -X12c -K -O >> image05.ps
pscoast -R -J -B0:."plot_3": -W1 -X-12c -Y-8c -K -O >> image05.ps
```

```
pscoast -R -J -B0:."plot_3": -W1 -X12c -O >> image05.ps
```

结果

注意这张纸横过来了，由于没有加-P，所以是图形本身是竖着的，需要旋转纸



解释

这是一个完整的bash脚本文件，第一行 ‘#!/bin/bash’ 表示用bash解释下面的语言，固定用法。

■ gmtset

改变gmt的默认设定 打开路径下的隐藏文件 ‘./gmtdefaults4’ 可以查看更多默认设置。

HEADER_FONT_SIZE	标题字体大小
BASEMAP_TYPE	=plain时去掉地图周围的 ‘铁轨花纹’ ；默认=fancy

■ basemap

不必理解这个命令。在这里的作用就是加上背景里的网格线，网格间隔为1cm；并设定起点为 ( 0 , 0 )

## ■ 4 个 pscoast

注意画图顺序与-X-Y的关系。

psbasemap	-X0c -Y0c	将画图的起点设为 ( 0 , 0 ) , 默认起点为 '-X2.5c -Y2.5c'
Plot_1	-Y10c	( 在原先的基础上 ) 将第一幅图的起点上移10cm , 成为 ( 0c , 10c )
Plot_2	-X12c	( 在第一幅图的起点上 , ) 将第二幅图的起点右移12c , 成为 ( 12c , 10c )
Plot_3	-X-12c -Y-8c	( 在第二幅图的起点上 , ) 将第三幅图的起点左移12c , 下移8c , 成为 ( 0c , 2c )
Plot_4	-X12c	( 在第三幅图的起点上 , ) 将第四幅图的起点右移12c , 成为 ( 12c , 2c )

可以发现 , '-X -Y' 改变的是相对上一次的值 , 且是永久改变。前缀 'a' 使得这次改变没有永久性 , 具体参照《GMT\_Manpages.pdf》中的psbasemap对这个开关的描述。如果命令不带 '-X -Y' , 则起点与上个命令的重合。

## 6. 图六 , 二维图 ( makecpt , grdimage, psscale )

### 目标

1. 使用 'GMT4/doc/examples/ex01/osu91a1f\_16.nc' 下的二进制文件画二维彩图
2. 加上海岸线
3. 加上色标尺

这里 , 依次需要这样三个命令 : grdimage负责生成二维彩图 , pscoast负责在这个图上加上海岸线 , psscale负责加入色标尺( 即颜色与值的对应关系 )。用命令的方式解释上面的规则 , 就是 :

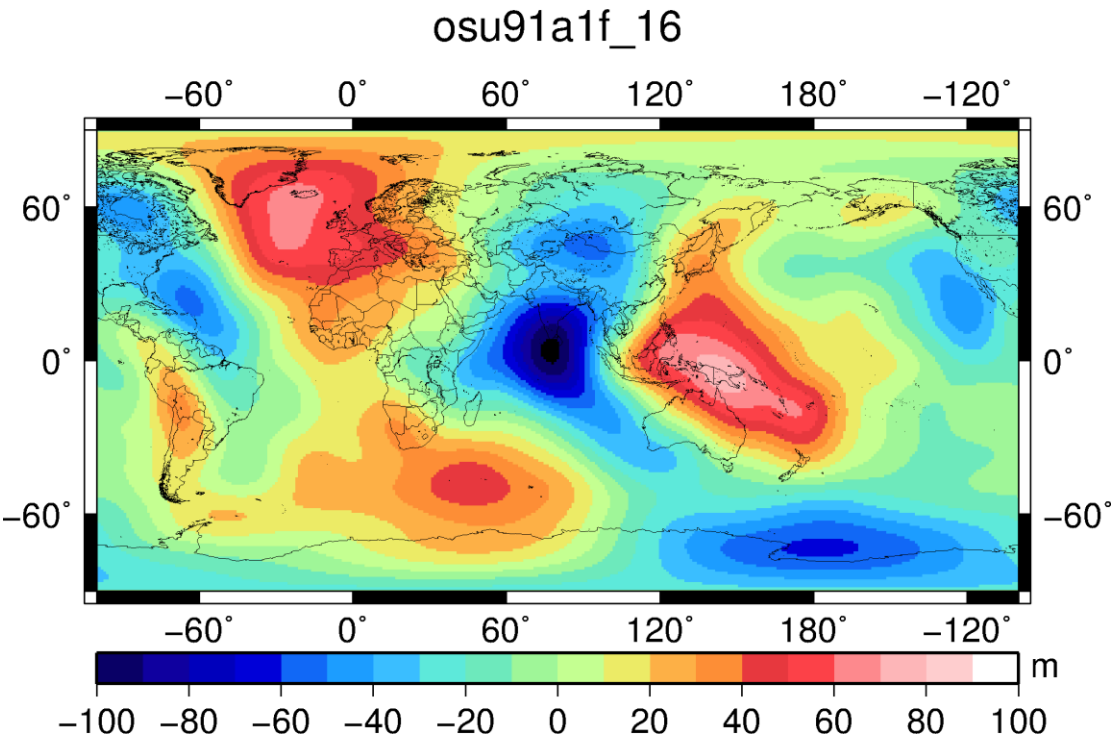
```
grdimage ... -K > image06.ps
pscoast ... -K -O >> image06.ps
psscale ... -O >> image06.ps
```



代码

```
makecpt -Chaxby -T-100/100/10 > 'image03.cpt'
grdimage osu91a1f_16.nc -C'image03.cpt' -Rd -JQ80/15c
-B60:."osu91a1f_16": -K > 'image06.ps'
pscoast -R -J -W -N1 -O -K >>'image06.ps'
psscale -C'image03.cpt' -D"7.5/-1/15/0.5h" -B"20/:m:"
-O >>'image06.ps'
```

结果



解释

■ makecpt

创建色标尺。可以查出osu91a1f\_16.nc的取值范围为-103~76。二维彩图就是基于一套值与色彩的对应关系，即色标尺。色标尺有多种配色方案，这里取的 'haxby' 为：用深蓝色表示-100，天蓝色表示-50，绿色表示0，红色表示50，白色表示100。这个变化关系可以用makecpt直接生成。

-Chaxby	采用 'haxby' 配色方案，常见的还有 'rainbow'，'polar'。替换名字即可
---------	--

	比较差异。在命令行中打入空命令 'makecpt' 可以查看所有的配色名称。
-T-100/100/10	从-100 到 100，每隔 10 定一个配色。

## ■ grdimage

读入二进制文件画彩图

osu91a1f_16.nc	二进制输入文件
-C'image03.cpt'	采用的色标尺；这里来自上面 makecpt 的生成
-Rd	取值范围，等效为-R-180/180/-90/90，即全球取值范围
-JQ80/15c	投影方式，参考“GMT 地图投影研究”
-B60:."osu91a1f_16":	60 的标注间隔，不加网格线；引号内为标题

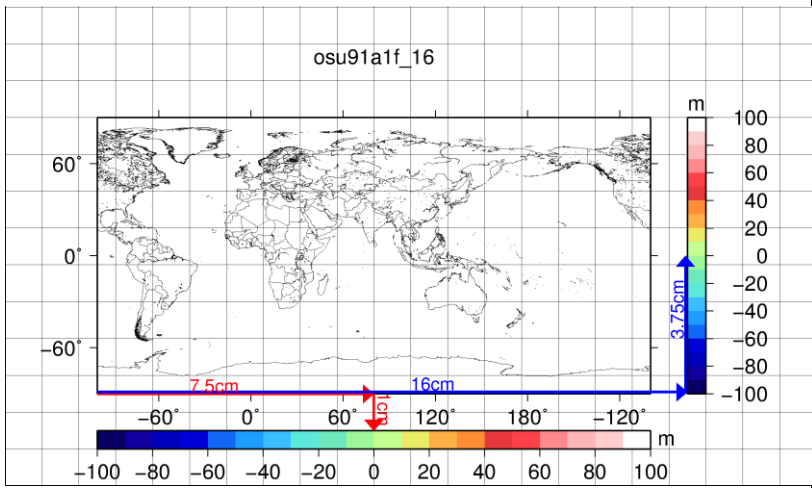
## ■ pscoast

画海岸线。

## ■ psscale

画图像最下方的色标尺。

-C'image03.cpt'	输入标尺文件
-D"7.5/-1/15/0.5h"	<p>标尺的位置与大小。四个值分别表示“平移经度/平移纬度/标尺长度/标尺宽度”。“平移”这个概念见下面的示意图。红色表示色标尺中心点先左移 7.5cm，后下移 1cm。最后的 h 表示水平 (horizontal) 标尺，缺省的话是垂直标尺。</p> <p>如果把这个参数改成“-D16/3.75/7.5/0.5”，会生成右边的色标尺，蓝色参数表示它的平移过程。</p> <p>-D 参数不会移动后面命令的作图起点。</p>

	
<code>-B"20/:m:"</code>	<p>色标尺的标注间隔为 20。两个冒号中的内容为坐标标签。斜线表示后面的为 y 坐标设定，即把单位 m 放在 y 轴方向。两个冒号用不同的引导表示的定义不同，前面已经说到用点号引导的是标题。没有引导的是坐标轴标签，等号引导的是前缀，逗号引导的是单位。即：<code>[: "Axis label":] [:="prefix":] [:,"unit label":]</code></p>

## 附 录

### A. 安装

介绍微软系统与Linux系统上的安装方法.前者环境为Windows 7, 后者为Ubuntu

10.04. XP的安装与检查方法同win7完全一样.

其他Linux可能会遇到缺少一些必须文件, 要自行到google搜索相关错误原因.

#### Windows 7

##### ■ 准备文件

从 [ftp://ftp.soest.hawaii.edu/gmt/windows/gmt-4.5.9\\_install32.exe](ftp://ftp.soest.hawaii.edu/gmt/windows/gmt-4.5.9_install32.exe) 下载

gmt安装包 (,如果需要64位的将32改成64即可)

##### ■ 安装文件

双击安装完毕.可选择修改安装路径.

##### ■ 系统路径设置

这一步并不需要修改什么, 只是检查确保无误. 先进入环境变量:

我的电脑(右键)→属性→(左侧面板)系统高级设置→(标签页)高级→(右下角)环境变

量, 应该有两个变量:

```
GMT_SHAREDIR=C:\programs\GMT4\share  
path=C:\programs\GMT4\bin;C:\programs\GMT4\bin
```

第一个变量要与安装路径一致(我这里是默认路径);

这里的path有两个路径(分号隔开). 你的可能含有更多的路径, 但是至少包含这两个.

有缺失的话要添补上. 如果以前安装过gmt, 请删掉第一个环境变量, 以及删掉path中对应的两个路径后再安装. 安装完后, 或者改动环境变量后, 需要重启(使环境变量生效).

##### ■ 测试是否安装成功

进入CMD( 按下开始, 搜索框输入cmd, 回车 ), 输入gmtinfo. 如果出现下面的内容, 则安装成功:

```
usage: gmtinfo leg(s)
```

## Linux

### ■ 准备文件

ftp://ftp.scc.u-tokai.ac.jp/pub/gmt/下载:

- netcdf.tar.bz2
- gmt-4.5.9.tar.bz2 (GMT 安装包)
- GSHHS2.1.1\_full.tar.bz2 (地图文件)

### ■ 安装 netCDF

将netcdf.tar.bz2拷贝到 "~/Downloads/" 路径下, 并切换到该路径下.

```
tar xf netcdf.tar.bz2    #解压缩

./configure              #配置

make check

sudo make install
```

将会把库文件安装在 /usr/local/lib, 头文件安装在

/usr/local/include, 执行文件安装在 /usr/local/bin.

如果需要改变安装路径, 使用参数: --prefix=\$yourPath. 并且安装完后要声明:

setenv NETCDFHOME=\$yourPath (如果不改路径则不用声明)GMT安装过程通过变量

\$NETCDFHOME找到库文件

如果看到下面的信息表示check成功:

```
=====
All 7 tests passed
=====
```

## ■ 安装 GMT

复制文件 `gmt-4.5.9.tar.bz2` 与 `GSHHS2.1.1_full.tar.bz2` 到你要安装的路径, 如

`~/install/`, 然后执行:

```
tar xf gmt-4.5.9.tar.bz2          #解压缩

./configure                        #配置

make && make install              #编译与安装

#若 make 成功, 则 install. 若遇到错误, 参考下文的 ``学会排错". 这里解释一下
这三个命令的区别: configure, 检测系统的编译环境, 并按照用户指定的(或者默认
的)参数生成 Makefile 供后面的 make 调用; make, 把源文件编译成可执行文件和库
文件; make install, 把上面得到的文件复制到相应路径, 可能需要 root 权限.

make clean                        #清除过程文件

cp GSHHS2.1.1_full.tar.bz2 GMT4.5.9 #拷贝边界数据到安装路径

tar xf GSHHS2.1.1_full.tar.bz2    #解压缩, 会在 share 路径下加入边界数
据

echo PATH=~/install/GMT4.5.9/bin:$PATH >> ~/.bashrc

#添加 GMT 执行路径到 bash 配置文件.bashrc(如果不是 bash 要做相应修改, 如 csh
对应.cshrc, 注意配置语法略不同)

source ~/.bashrc

#使.bashrc 修改生效
```

## ■ 检验是否安装成功

```
# 切换到第一个例子目录，同样可以对其他例子做测试

cd ~/install/doc/examples/ex01

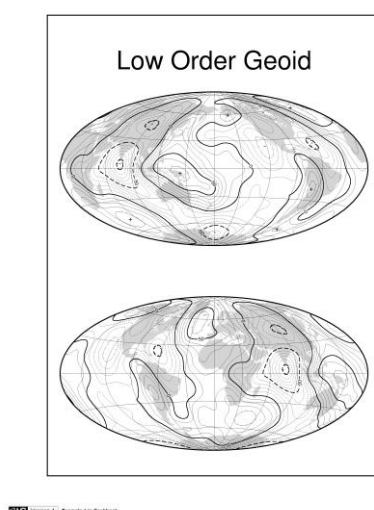
# 执行 shell 脚本

sh job01.sh

#切换到上层路径。

cd ..
```

如果可以看见下面的example\\_01.ps(需要ps查看软件)，表明安装成功



## 学会排错

安装过程中可能会遇到各种问题，要查看输出信息中第一个ERROR所在行的内容。比如安装netCDF时，告诉我strlen错误，定位到对应的cpp文件，发现缺少头文件声明，加上#include "string.h"便可以解决问题。

又如编译GMT时，提示"error: X11/Intrinsic.h: No such file or directory"，上google搜索这句内容，发现是缺少libxt-devel，安装后继续报错：找不到文件X11/Xaw/Scrollbar.h，google搜索结果为缺少包libxaw7-dev。安装完后顺利编译。

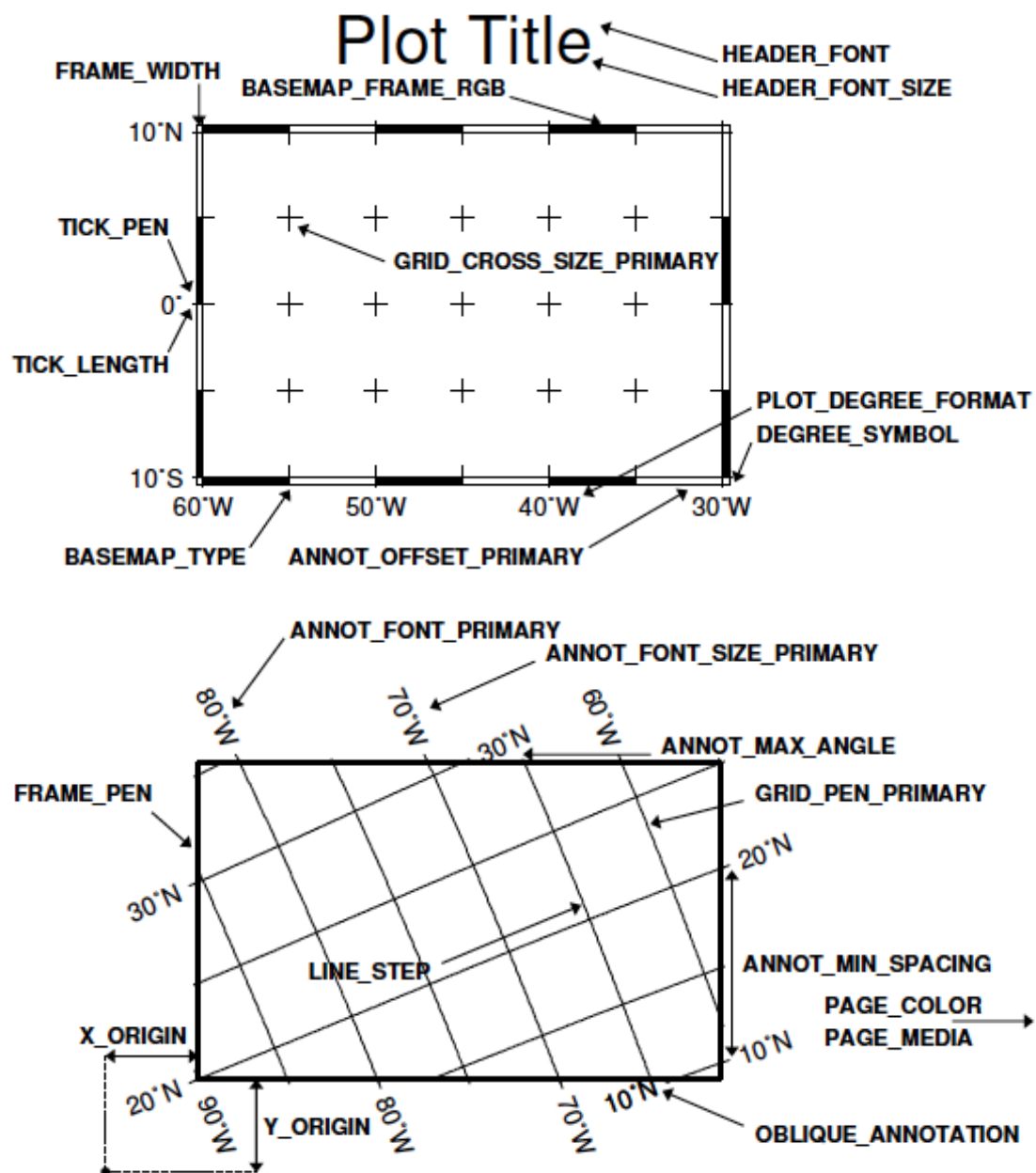
## B. 设定 (gmtset)

例如设置标题字体大小：

```
gmtset HEADER_FONT_SIZE=20p
```

参数意义的图示可以从GMT\_docs.pdf文件中查看，这里从该文档直接复制过来。

所有的参数可以在.gmtdefaults4中直接修改。





## C. 画笔

### 1) 尺寸

GMT 支持 3 四种长度单位：

- c ( 厘米 )
- i ( 英寸 , inch )
- p ( 点 , =1/72 inch )
- m ( 米 )

如：5c 表示 5 厘米

GMT 还支持名称定义的尺寸

<i>Pen name</i>	<i>Width</i>	<i>Pen name</i>	<i>Width</i>
faint	0	thicker	1.5p
default	0.25p	thickest	2p
thinnest	0.25p	fat	3p
thinner	0.50p	fatter	6p
thin	0.75p	fattest	12p
thick	1.0p	obese	18p

### 2) 颜色

#### ■ 灰度

( 范围 0-255 的数字 : n )

0: black ; 255: white

#### ■ RGB 颜色

( 两个斜线/隔开的三个数字 : r/g/b , 三个数字范围 0-255 )

0/0/0: black; 255/255/255: white;

## ■ HSV 颜色

( 两个 '-' 隔开的三个数字 : h-s-v ; h 范围 0-360 , s 与 v 范围 0-1 )

h-s-v; h: 0-360; s, v: 0-1;

## ■ CMYK 颜色

Cyan/magenta/yellow/black; each: 0-100%

## ■ 颜色名

633 个颜色名 ; man gmtcolors 显示所有颜色名 ; white, black, red, orange, yellow, green, cyan, blue, magenta, gray/grey, brown; 支持 DrakGreen, LightBlue

## ■ 十六进制颜色

#rrggbb , 网页颜色也采用这种记法

## 3) 形状

- a) . 虚线 ( 长宽相等 , 间隔为 4 倍宽 width )
- b) - 短划线 ( 长为 8 倍宽 , 间隔为 4 倍宽 )
- c) ..- 点-点-短划线 ( 任意组合 )
- d) 4\_8\_5\_8:2p [自定义长度和间隔]短划线长度分别为 4p、5p , 间隔都为 8p , 起始位移为 2p

## 4) 画笔属性 ( -W )

定义海岸线、自定义图形、插入文字的轮廓样式。

```
-W[width[c|i|p|m]], [color], [texture[cjijpjm]]
```

示例:

<i>Pen example</i>	<i>Comment</i>
<b>-W0.5p</b>	Solid black line, 0.5 point thick
<b>-Wgreen</b>	Solid green line with default width
<b>-Wthin,red,-</b>	Dashed, thin red line
<b>-Wfat,.</b>	Fat dotted line [black]
<b>-W0.1c,120-1-1</b>	Green (in h-s-v) pen, 1 mm thick
<b>-Wfaint,100/0/0/0,...-</b>	Very thin, cyan (in c/m/y/k), dot-dot-dashed line

## 5) 填充属性 ( -G )

定义一块区域的填充样式，可以是颜色，也可以是 GMT 内置的花纹

<i>Fill example</i>	<i>Comment</i>
<b>-G128</b>	Solid gray
<b>-G127/255/0</b>	Chartreuse, R/G/B-style
<b>-G#00ff00</b>	Green, hexadecimal RGB code
<b>-G25-0.86-0.82</b>	Chocolate, h-s-v – style
<b>-GDarkOliveGreen1</b>	One of the named colors
<b>-Gp300/7</b>	Simple diagonal hachure pattern in b/w at 300 dpi
<b>-Gp300/7:Bred</b>	Same, but with red lines on white
<b>-Gp300/7:BredF-</b>	Now the gaps between red lines are transparent
<b>-Gp100/marble.ras</b>	Using user image of marble as the fill at 100 dpi

## D. 文字格式

图形中文字（如标题、图例、坐标轴标注等）的字体、颜色、字号、上下标、希腊字母等。

用 '@' 作为转义引导符。

<i>Code</i>	<i>Effect</i>
<b>@~</b>	Turns symbol font on or off
<b>@+</b>	Turns superscript on or off
<b>@-</b>	Turns subscript on or off
<b>@#</b>	Turns small caps on or off
<b>@_</b>	Turns underline on or off
<b>@%fontno%</b>	Switches to another font; @%% resets to previous font
<b>@:size:</b>	Switches to another font size; @:: resets to previous size
<b>@;color;</b>	Switches to another font color; @;; resets to previous color
<b>@!</b>	Creates one composite character of the next two characters
<b>@@</b>	Prints the @ sign itself

## E. ASCII 与 grid 文件

ASCII 文件是我们通常自己建立的数据文件，如一个包含三列的数据文件，每一列分别为：经度，纬度，值。这种文件可以直接用文本编辑器打开，非常直观。如：

```
-179.50 -89.50 -2.8210
-178.50 -89.50 -2.8246
-177.50 -89.50 -2.8282
-176.50 -89.50 -2.8319
-175.50 -89.50 -2.8355
-174.50 -89.50 -2.8392
-173.50 -89.50 -2.8429
-172.50 -89.50 -2.8466
-171.50 -89.50 -2.8503
-170.50 -89.50 -2.8541
```

然而，这样的数据格式很多情况下并不能直接被 GMT 使用，需要转成 grid 二进制文件。GMT 中的 grd 打头的命令，如 `grdimage`（画二维图），`grdcontour`（等值线图）。还有几个命令是可以直接读取 ASCII 文件的，如 `psxy`（画点或线）。

在画第 6 幅图的时候，如果自己提供 `ascii` 文件，需要通过下面的命令转成 grid 文件：

```
blockmean filename.txt -Rd -I0.25 > interp.xyz
surface -Rd -I0.25 interp.xyz -Gfilename.grd
```

`blockmean` 是插值，`-I` 是插值分辨率，即按照 0.25 度的分辨率插值。如果原始数据分布不均匀，这个命令就很有用。第二个命令是通过插值的 ASCII 文件生成 `grb` 文件，`-I` 也是分辨率，决定后面 `grdimage` 生成图形的精细程度。