



GMT 基础教程

The Generic Mapping Tools

GMT

Version 4.2.1

Technical Reference and Cookbook

by

Pål (Paul) Wessel

School of Ocean and Earth Science and Technology
University of Hawai'i at Mānoa

and

Walter H. F. Smith

Laboratory for Satellite Altimetry
NOAA/NESDIS

October 2007



Generic Mapping Tools Graphics

目 录

序	1
1. GMT 安装教学.....	2
1.1 Windwos Xp 下安装 GMT 的方法.....	2
1.2 Linux 下安装 GMT.....	3
2. GMT 中文帮助.....	7
2.1 GMT 纵览: 历史, 体系和使用.....	7
2.2 GMT 环境 (GMT environment)	8
2.3 GMT 常用命令行选项.....	13
The -H option.....	16
2.4 练习 (Laboratory Exercises)	19
2.5 第二部分 (SESSION TWO)	22
3. Cookbook.....	41
3.1 由网格数据文件绘制等值线图.....	41
3.2 由网格数据文件绘制彩色影像图.....	42
3.3 3-D 透视网线图.....	45
3.4 3-D 照明表面图.....	47
3.5 两类直方图	48
3.6 一个典型的位置图.....	48
3.7 3 -D 柱状图.....	51
3.8 沿测线变化的时间 - 序列图.....	51
3.9 地理坐标棒图	53
3.10 数据最佳三角形线性内插 (Optimal triangulation of data)	54
3.11 数据网格化和趋势面.....	56
3.12 更多的数据网格化方法.....	58
3.13 体积计算和特定地区的圈定.....	60
3.14 彩色图案模版的使用.....	64
4. GMT 缺省设置文件 (.gmtdefaults) 注释	66
5. GMT summary.....	69
6. GMT 常用命令介绍.....	72
6.1 psbasemap GMT 画底图的指令	72
6.2 pscoast GMT 画陆块海洋和海岸线的程序	73
6.3 psxy GMT 在地图上画点线的程序	74
6.4 grdimage GMT 画二维等高图的指令	75
6.5 psscale GMT 画比例尺的指令	75
6.6 grdcontour GMT 画二维等高线图的指令	76
6.7 grd 数据的处理	76
7. 实 例	77

序

GMT 通用地学绘图软件已经在地理、地球物理等学科得到广泛应用，GMT 已经成为这个年代地球科学立最热门的主题词之一，而且将来也是。作为一个共享软件，不存在版权问题。在不断的学习探索中本人深感过程艰难，时而彷徨。有鉴于此，将收集到的有关资料与学习中的一些简单过程整理成所为的教程，并公布出来，仅供与初学者交流学习之用，作者水平有限，错误之处希望读者指正。

作者：许小强（西安 hsxxq@163.com）

版权声明及致谢

本教程版权归作者所有，任何人不得擅自更改，复制，散播，或者用于商业用途。

文中部分内容来自网络及其它途径，对相关作者表示感谢，如易长荣（正是看了他的例子，作者才真正进入了 GMT 世界）。

免责声明

由于是在学习之余义务整理出来的教程，作者没有用于商业用途，因此不对教程中错误之处引起的任何后果负责。

1. GMT 安装教学

1.1 Windwos Xp 下安装 GMT 的方法

1. 到 夏威夷大学 [SOEST](#) 的网站 点至左方 Mirrors，选取适当的 FTP 站
2. 下载 netcdf-3.6.1-win32.zip
3. 进入"4"， 即 GMT 4 版的目录，下载以下的文件

GMT_exe.zip All executables
GMT_share.zip Support run-time files (patterns , cpt)
GSHHS_coast.zip Basic GSHHS coastlines
GMT_suppl_exe.zip Supplemental executables
GMT_pdf.zip PDF documentation and Unix man pages
GMT_man.zip Unix man pages for GMT
GMT_web.zip HTML versions of all documentation
GMT_tut.zip Tutorial data sets
GMT_src.zip All source code
GMT_scripts.zip Example scripts and data
GMT_suppl.zip Supplemental program source and docs
GSHHS_high.zip High resolution GSHHS coastline files
GSHHS_full.zip Full resolution GSHHS coastline files

将 GMT 开头的一起解压缩（这样会出现一个文件夹叫做 GMT，其下有

bin (gmtprograms)
examples (gmtexamples)
lib
man
netcdf
share (rivers , borders , shorelines , patterns etc)
src
tutorial
www 等的文件夹，里面又各有许多文件。

4. 将这个 GMT 文件夹移至 C:
5. 将 netcdf-3.6.1-win32.zip 解压缩置于 C:NETCDF
6. Copy C:netcdfnetcdf.dll 到 C:windowssystem32

-
7. 到控制台->系统设定->进阶->环境变量 的“系统变量”中增加一项变量名称为 GMTROOT 而变量值为 C:GMT PATH 中 加入 C:GMTbin 储存
 8. 在 DOS 窗框中 打入 pscoast 看看是否有错误信息; 若是有一长串说明档出现, 就表示 gmt 本身安装成功
 9. 下载 [Ghostscript](#) 以及 [GSView](#) 并安装于系统中

1.2 Linux 下安装 GMT

Linux 环境下安装也很简单, 但由于大多数人对 Linux 不熟悉, 觉得安装很麻烦! 刚开始我也一样, 呵呵! 如果你不想了解 Linux 下一般安装过程的话, 推荐你使用 Ubuntu, Ubuntu 下安装软件方便的很, 很适合新手! 因为 Ubuntu 倡导是 Human Being ! 现在 Ubuntu 的源里已经把 GMT 加入了, 说明了什么? GMT 很 popular 啊! Ubuntu 下通过新里德搜索~gmt 即可, 但现在版本是 4.2.0, 有的是 4.1.0, 这无所谓了, 基本上够用了! 顺便说一句, 我使用的就是 Ubuntu! 说到这里, 不能不感谢 Jakii, 是他把我带入 Linux 世界, 准确的来说是 Ubuntu 世界, 成了 Ubuntu 的 fans! 废话少说, 开始 Linux 下 GMT 安装之旅吧。

Linux 下安装 GMT 有两大步骤:

(1) 安装 Netcdf

我一般把软件安装在/usr/local 下面, 下载的文件在放在/home/xu/download 下。安装命令如下:

```
sudo cp /home/xu/download/netcdf.tar.gz /usr/local  
sudo tar -zxvf /usr/local/netcdf.tar.gz  
cd /usr/local/netcdf-3.6.2  
sudo ./configure  
sudo make check  
sudo make  
sudo make install
```

NETCDF 安装就这几步, 其实这也是 Linux 下从源码安装软件的一般步骤。解开源码包, 里面一般都有个 Makeconfig、Readme 等文件, 里面详细介绍了安装过程。

(2) 安装 GMT

在/usr/local 建立文件夹 gmt4.3.0

```
sudo mkdir -p /usr/local/gmt4.3.0  
cd /usr/local/gmt4.3.0
```

```
sudo cp /home/xu/download/GMT* ./ ( 复制所有 GMT 相关安装文件包到欲  
安装的目录)
```

tar 开所有压缩包

余下步骤欲安装 NETCDF 一样，或者使用安装包提供的安装脚本 `install_gmt` 进行安装，但这样安装很麻烦，有时还出错。我在 Foreda 下安装成功，在 Ubuntu 下安装的不全。

GMT 官方网站推荐在线填表，然后自动安装。但国内网速实在是太慢了，在线安装谈何容易？仔细研究发现，所为的自动安装也就是创建了个自动应答文件，在线填完表后，它会让你把填表内容存成个 `GMTparam.txt`，里面记录的主要内容有：

netcdf 安装目录、gmt 安装目录、gmt 安装选项、从哪或许安装包(这里有个选项就是你已经或得安装包，我就是看到这个了才得以顺利的使用它的这个自动安装)等。

`GMTparam.txt` 内容如下：

```
# This file contains parameters needed by the install script  
# install_gmt for GMT Version 4.3.0. Give this file  
# as the argument to the install_gmt script and the whole  
# installation process can be placed in the background.  
# Default answers will be selected where none is given.  
# You can edit the values , but do not remove definitions!  
# Assembled by GMT_install_form.html , Revision: 1.59  
# Processed by install_GMT_form.pl : 1.32 on  
# Wed May 7 21:57:17 HST 2008  
# Do NOT add any spaces around the = signs. The  
# file MUST conform to Bourne shell syntax  
# -----  
# SYSTEM UTILITIES  
# -----  
GMT_make=make  
# -----  
# NETCDF SECTION  
# -----  
netcdf_ftp=n netcdf_install=n
```

```
netcdf_path=/home/xxq/netdf3.6
passive_ftp=y
# -----
#  GMT FTP SECTION
# -----
GMT_ftp=n
GMT_ftpsite=9
GMT_get_src=y
GMT_get_share=y
GMT_get_coast=y
GMT_get_high=y
GMT_get_full=y
GMT_get_suppl=y
GMT_get_scripts=y
GMT_get_pdf=y
GMT_get_tut=y
GMT_get_web=y
# -----
#  GMT SUPPLEMENTS SELECT SECTION
# -----
GMT_suppl_dbase=y
GMT_suppl_gshhs=y
GMT_suppl_imgs=y
GMT_suppl_meca=y
GMT_suppl_mex=n
GMT_mex_type=octave
GMT_suppl_mgd77=y
GMT_suppl_mgg=y
GMT_suppl_misc=y
GMT_suppl_segyprogs=y
GMT_suppl_spotter=y
GMT_suppl_x2sys=y
GMT_suppl_x_system=n
GMT_suppl_xgrid=n
```

```
# -----
# GMT ENVIRONMENT SECTION
# -----
GMT_si=y
GMT_ps=y
GMT_prefix=/home/xxq/gmt4.3.0
GMT_bin=
GMT_lib=
GMT_share=
GMT_include=
GMT_man=
GMT_web=
GMT_sharedir=
GMT_dir_full=
GMT_dir_high=
GMT_dir_cli=
# -----
# COMPILING & LINKING SECTION
# -----
GMT_sharedlib=n
GMT_cc=gcc
GMT_flock=y
GMT_triangle=n
# -----
# TEST & print FILE SECTION
# -----
GMT_run_examples=n
GMT_delete=n
# -----
# MEX SECTION
# -----
```

我用这种方法屡试不爽 在 Ubuntu 及 Feroda 下都是一次安装成功!

2. GMT 中文帮助

2.1 GMT 纵览：历史，体系和使用

2.1.1 历史回顾 (Historical highlights)

GMT 体系开始于 1987 年后期，在美国哥伦比亚大学 Lamont-Doherty 地球观测所，由研究生 Paul Wessel 和 Walter H. F. Smith 创立。1988 年 7 月，GMT 的第一版被正式介绍给 Lamont 的科学家。GMT 1 口头上（和磁带）被传播到美国、英国、日本、法国其它研究所并吸引了少数后来者。1989 年 12 月 Paul 在 SOEST 获得博士后位置并继续从事 GMT 的开发工作。GMT2.0 版于 1991 年 10 月 在 EOS 以论文发表并很快传播到全世界。我们的 GMT3.0 版在 1993 年得到美国国家科学基金会的少量赞助，并于 1995 年 8 月 15 日在 EOS 以另一篇论文发表。经过重大改进的版本 (3.1 –3.3, 3.3.1–6; 3.4, 3.4.1–3) 分别公布于 1998 年 11 月到 2003 年 5 月期间，最新版本是 2004 年 1 月的 3.4.4 版。现在 GMT 在世界范围各个学科的用户已经超过 6,000。

2.1.2 体系 (Philosophy)

GMT 采用 UNIX 体系，把复杂的任务分解为较小和易处理的部分。单独的 GMT 模块很小，容易维护，也能使用任何其他的 UNIX 工具。GMT 用 ANSI C 程序语言写成（非常简便），适应于 POSIX 和 Y2K，不受硬件（如内存）限制。GMT 有意用命令行形式写成，而不用 Windows 环境，以求最大程度的机动性。我们早就采用为标准化 PostScript 输出格式，而不用后文件（meta-file）的形式。除了支持海岸线数据外，GMT 完全 decouples 了由主 GMT 程序读取数据的操作。GMT 使用独立结构的文件格式。

2.1.3 为什么 GMT 如此普及

The price is right! 因为可以由命令行, scripts 内部和用户程序调用，GMT 具有无限的适应性。GMT 因其高质量的 PostScript 输出已经吸引了众多的用户。GMT 很容易安装在几乎所有的计算机上。

2.1.4 GMT 安装环境 (GMT installation considerations)

GMT 已经被安装在由超级计算机到笔记本 PC 机的各类计算机上。GMT 只包括大约 55,000 行代码，仅有适度的空间/内存需求。其最低需求是：

netCDF library 3.4 或更高（由 www.unidata.edu 免费下载）.

一个 C 编辑器（由 www.gnu.org 免费下载）.

大约 100 Mb 磁盘空间 (70 Mb 附加的精细分辨率和高分辨率的全球海岸线数据库) 大约 32 Mb RAM.

此外，我们推荐 使用一个 PostScript 打印机或相当 的打印机（例如 ghostscript），PostScript 预览器（例如 ghostview），任何类型的 UNIX 操作系统，比较多的磁盘空间和 RAM。

2.2 GMT 环境 (GMT environment)

2.2.1 Tutorial setup (见附件“GMT 安装”)

2.2.2 GMT 环境 (The GMT environment)

图 1.1 为运行 GMT 时你需要了解的关系

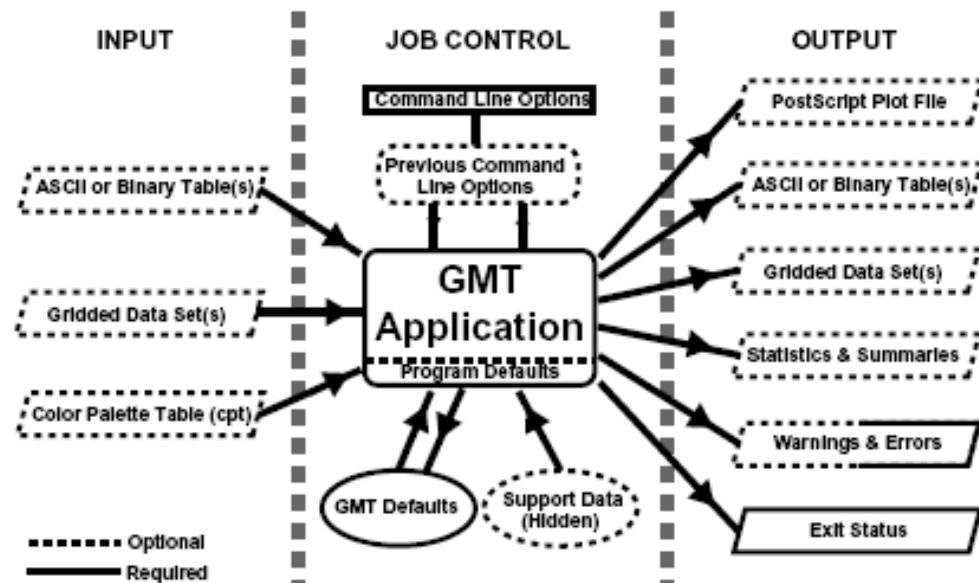


图 1.1: GMT 运行环境

数据输入 (Input data)

一个GMT 程序可以输入也可以不输入数据。要认识3类不同类型的输入文件 更多的细节可在技术参考的 Appendix B 中看到) :

1. 数据表。电子表格的列数有一定限制，行数不限。可以分为两类：

ASCII 码文件（除非数据文件巨大，为首选格式）

单段文件 [缺省]

具有内部标头记录的多段文件 (-M)

二进位文件（加快输入/输出速度）

单段文件 [缺省]

多段文件（段的头记录 NaN 域） (-M)

2. 网格数据组。为数据矩阵（数据点在两个坐标方向等距分布）。有两种格式：

网格线配准 (Grid-line registration)

像素配准 (Pixel registration)

你可以从几个文件格式中选择（也可以定义你自己的文件格式），但 GMT 缺省格式是 netCDF。

3. 调色板表（用于影像图、彩色图和等值线图）。

任务控制 (Job Control)



GMT 程序可以由以下几方面获得运行参数：

1. 由命令行选项/转换开关或程序缺省值所提供
2. 用省略符号，选项后不加参数说明参数同前（保存在文件 .gmtcommands 内）
3. 使用 GMT 各种参数的缺省值（保存在文件 .gmtdefaults 内）
4. 可以使用隐含的像海岸线或 PostScript 图案这样的数据

数据输出 (Output data)

GMT 得到的 6 种输出文件列表如下：

1. PostScript 绘图文件
2. 数据表
3. 网格数据组
4. 统计和概要
5. 警告和出错信息，写入 stderr.
6. 退出状态 (0 为正常，其他为失败)

注意：GMT 自动创建和更新其公共转换开关命令选项的历史纪录。这些选项的历史文件是 .gmtcommands ，在每一个运行 GMT 程序的目录下都被创建。许多 GMT 使用中出现的基础问题都是由于没有充分了解图 1.1 所示的关系。

2.2.3 UNIX 环境，入门级知识

重定向 Redirection

大多数 GMT 程序由终端（称为标准输入 stdin）或文件输入数据，向终端输出数据（称为标准输出 stdout）。可用重定向来代替标准输入/输出。

GMTprogram input-file >output-file

GMTprogram < input-file > output-file

GMTprogram input-file >> output-file “>>” 加入到已有文件的后面

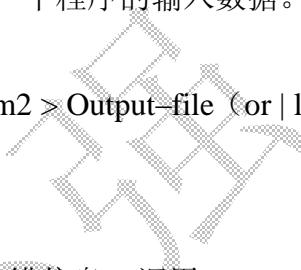
编者注： UNIX 的感叹号 (!) 表示允许覆盖已有文件， DOS 批处理文件不用感叹号。

管道命令 Piping (|)

有时我们想把一个程序的输出结果作为另一个程序的输入数据。可用 UNIX 的管道命令。 DOS 批处理文件相同。

Someprogram | GMTprogram1 | GMTprogram2 > Output-file (or | lp)

标准错误 Standard error (stderr)



大多数 UNIX 和 GMT 命令有时会输出出错信息。调用 stderr 输出典型的不同数据流，可由标准输出设备（stdout）重定向。为重定向出错信息我们使用：

UNIX 命令 >& errors.log

如我们想把程序输出结果和出错信息分别存储在不同的文件，我们使用下面语法：

(GMTprogram > output.d) >& errors.log

文件名扩展或替代符 (File name expansion or ``wild cards'')

如“*”代表任意个字符，“？”代表 1 个任意字符，“[list]”匹配字符列表等。

范例

GMT 命令 data_*.d 指对所有以 “data_” 开始并以 “.d” 字符结束的文件进行操作。

GMT 命令 line_?.d 指对所有以 “line_” 开始，后面跟任意一个字符，并以 “.d” 字符结束的文件进行操作。

GMT 命令 section_1[0-9]0.part_[12] 只对 sections 100 到 190 范围的数据进行处理， 只用 10 的整数 sections，同时得到 part 1 和 2 。

2.2.4 GMT 缺省参数设置 GMT Defaults

大量的次要功能（多于 50 个）只需改变 GMT 缺省参数设置即可实现。这些参数控制了诸如字体类型和大小、绘制底图的线宽、线性内插等许多参数。GMT 的缺省参数保留在文件 `.gmtdefaults` 中。用户可在主目录中保存一个 `.gmtdefaults` 文件，而在某些子目录中保存几个特定的 `.gmtdefaults` 文件。如果在当前目录下没有 `.gmtdefaults` 文件，GMT 就打开用户的 `.gmtdefaults` 文件。如果再没有的话，就采用系统缺省的 `.gmtdefaults` 文件。后者为 GMT 开发者设定的，但在 GMT 安装时可能被改变。在这个水平上的典型改变就是选择 SI 单位而不是英美制单位。并不推荐全面改变系统的缺省设置，因为某些应用程序可能有赖于缺省值的标准设置。用户可以创建一个新的 `.gmtdefaults` 文件，事先调整好当前 GMT 所使用的 `.gmtdefault` 值。

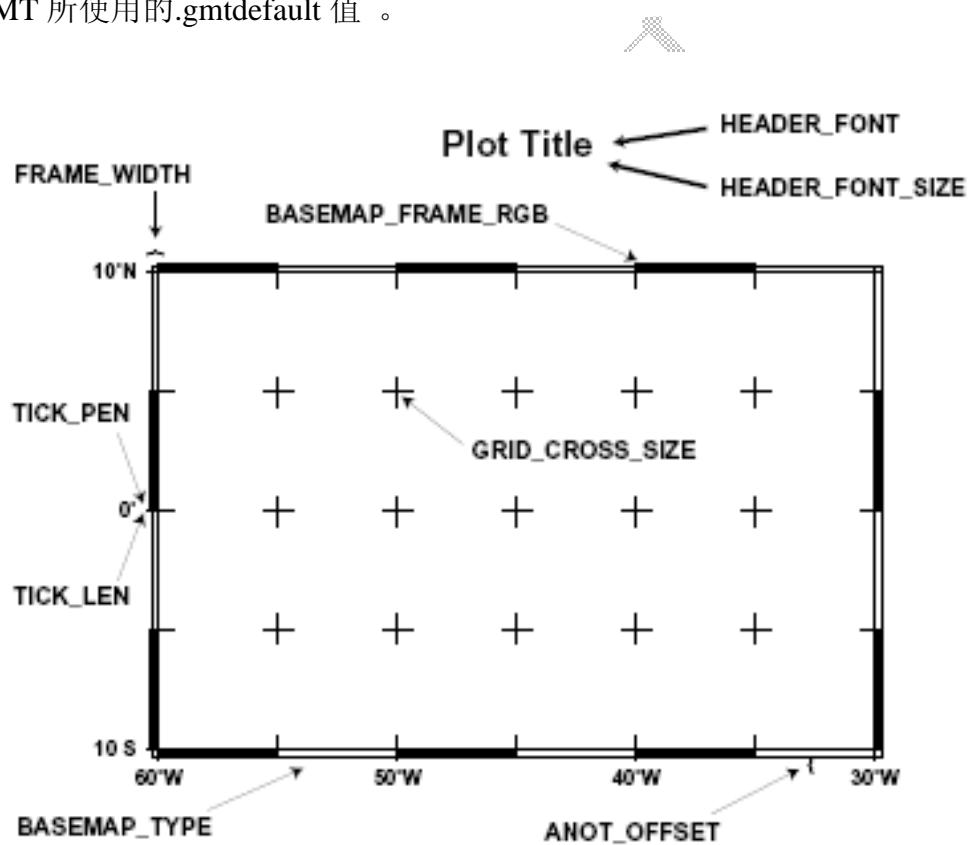


图 1.2: 影响绘图外观的一些 GMT 参数

至少有两个很好的原因来解释为什么要把 GMT 缺省选项放在一个单独的参数文件中：

1. 使用命令行语法来设置太多的选项并不实际，因为许多参数极少或从来不被改变（例如地图投影所使用的椭球模型）。

2. 保持独立的.gmtdefaults 文件对于特定的任务很方便，简单运行子目录里的 GMT 文件就可以达到特殊效果。例如，刊物论文的最终成图常要求对字体和字号标准化等，就可以把所有这类设置保存在一个单独的.gmtdefaults 文件中以简化该过程。同样，用于制作幻灯片的 GMT script 和在激光打印机输出的图经常使用不同的配色方案和大小。把不同的设置组织成不同的.gmtdefaults 文件可以把与图宏编辑（micro-editing）相关的麻烦减至最小。

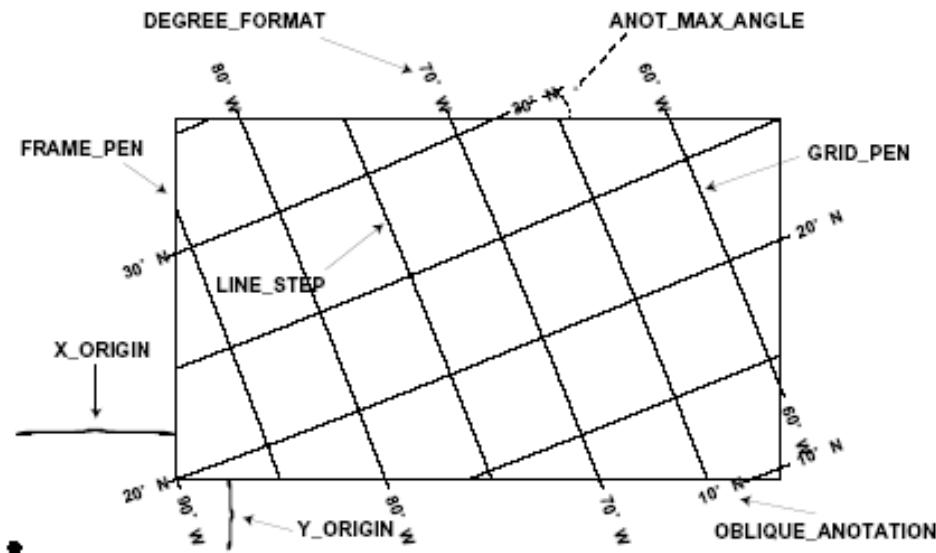


图 1.3: 影响绘图外观的更多的 GMT 参数

GMT 总是企图打开名为.gmtdefaults 的文件。有时，覆盖缺省值是值得的。我们提供另一种方法作为选择，即在 GMT 同一个命令行增加用加号 (+) 做前缀的备用的.gmtdefaults 文件。一种可能比较好的方法是每个程序开始时创建一个当前.gmtdefaults 文件的拷贝，然后把用户需要的.gmtdefaults 文件复制到当前目录，最后在程序结束取消改变。为了在程序内部改变一些 GMT 参数，可以使用 gmtset 命令。例如，为把注释字体改变为 12 点的 Times-Bold，执行命令

```
gmtset ANOT_FONT Times-Bold ANOT_FONT_SIZE 12
```

除修改单位、比例外还有大量参数，有 29 个参数直接影响到绘图效果。完全的列表见.gmtdefaults 主页。在程序的结尾，可以把特殊的参数重新设回 GMT 的原始设置。我们建议你至少把所有常用参数练习一次，以便了解如何通过设计一种方案来改变参数。

2.2.5 GMT 单位 (GMT Units)

GMT 可以接受的空间单位有厘米、英寸、米或点。有两种方法保证 GMT

了解用户想要使用哪种单位。

1. 给所应用的维度直接添加用户打算使用的单位。这种方法直接而清楚，比如，`-X4c` 表明是 4 cm。

2. 用`.gmtdefaults` 中的 `MEASURE_UNIT` 设置用户打算使用的单位。如无明确标明，所有的维度都被解释为缺省设置单位。

后一种方法不太安全，因为其他用户可能使用不同的单位，你的程序可能不像预期地那样执行。因此，我们推荐你始终采用直接添加用户打算使用的单位的方法。

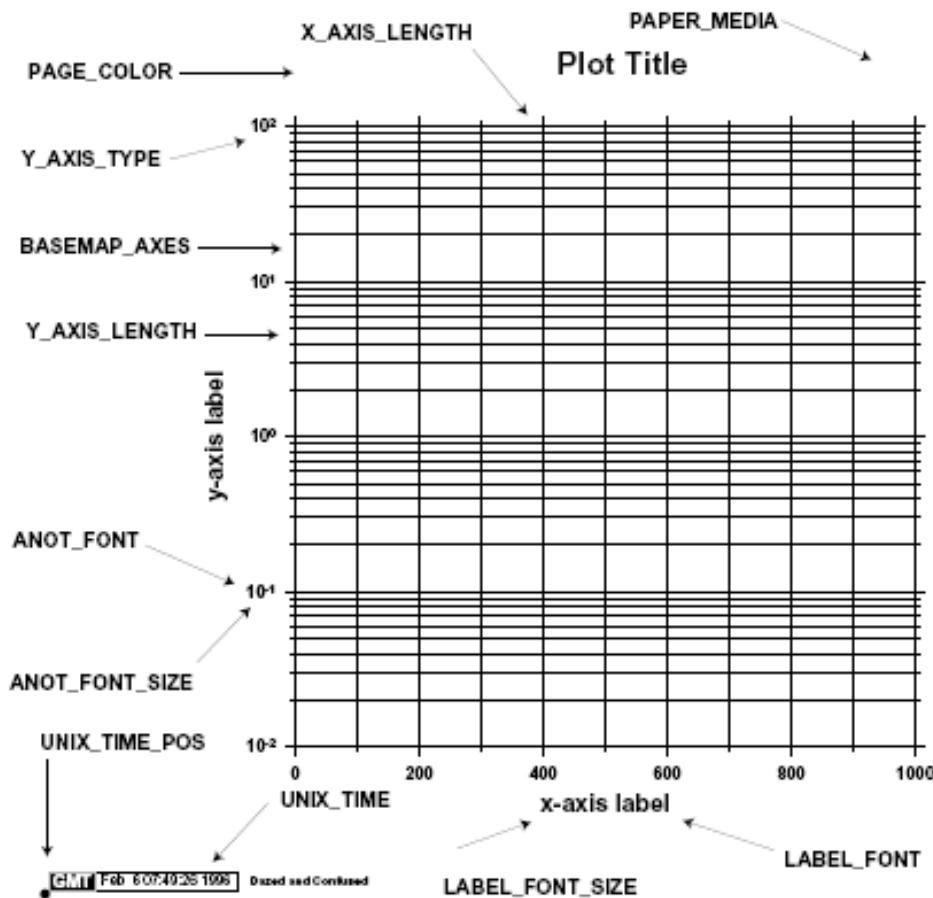


图 1.4: 影响绘图外观的更多的 GMT 参数

2.3 GMT 常用命令行选项

GMT 有 13 种选项在所有的程序中都相同。了解如何使用这些选项至关重要，按照字母顺序简述如下。

2.3.1 The `-B` option

这是 GMT 中最复杂的选项，但在大多数使用的例子中实际上又十分简单。

写为：

-Bxinfo[/yinfo][:"title string":][W|w][E|e][S|s][N|n]

该选项用选定的标志间隔指定了待绘制地图的边界。

xinfo 和 *yinfo* 的格式为

[a]tick[m|c][ftick[m|c]][gtick[m|c]][l/p][:axis label":][:,unit label":]

这里 **a**, **f**, 和 **g** 分别为刻度标记、刻度和网格的间隔。 **m|c** 选择分 (**m**) 或秒 (**c**)。 缺省设置是 4 个边界全部绘制 (表示为 **W, E, S, N**)。通过重写代码来改变缺省设置 (如, **WSn**)。 大写字母 (如, **W**) 说明同时画轴和轴的刻度标记。轴标记的格式受你的.gmtdefaults 文件中的 DEGREE_FORMAT 控制。

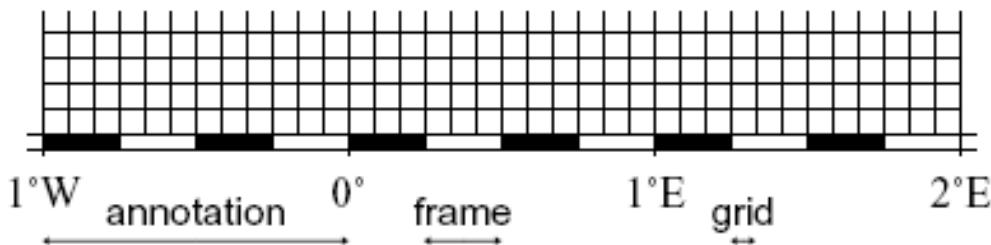


图 1.5 使用刻度标记、刻度和网格间隔的地理图边界

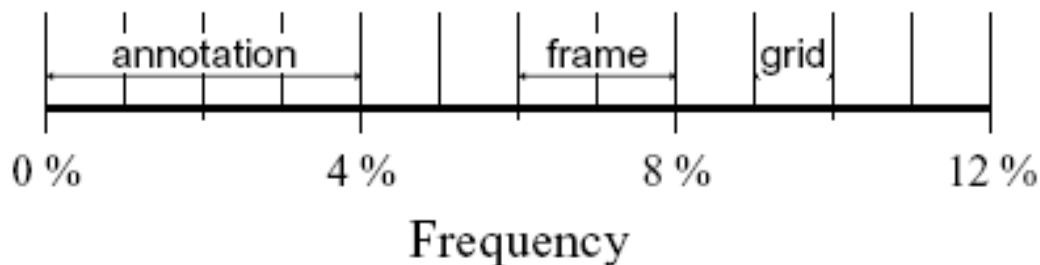


图 1.6: 线性笛卡尔投影轴

图 1.6: 线性笛卡尔投影轴。长刻度线为标记刻度, 短刻度线为刻度间隔, 轴名用 “:” 括起, 刻度单位“%”用“:,% :”表示。我们使用 **-R0/12/0/1 -JX3/0.4 -Ba4 f2 g1:Frequency::,%:**

log10 轴选项 (Options for log10 axes)

1. *tick* 参数必须选 1, 2, 或 3 。相应的对数标记间隔分别为 1, 1 –2–5, 或 1, –2, –3, –4, ..., –9。
2. *tick* 后加 **l** , 刻度标注为 \log_{10} 的每个整数值, 即 0, 1, 2 等。
3. *tick* 后加 **p**, 刻度标注为 \log_{10} 的指数形式 (如 10^{-5})。

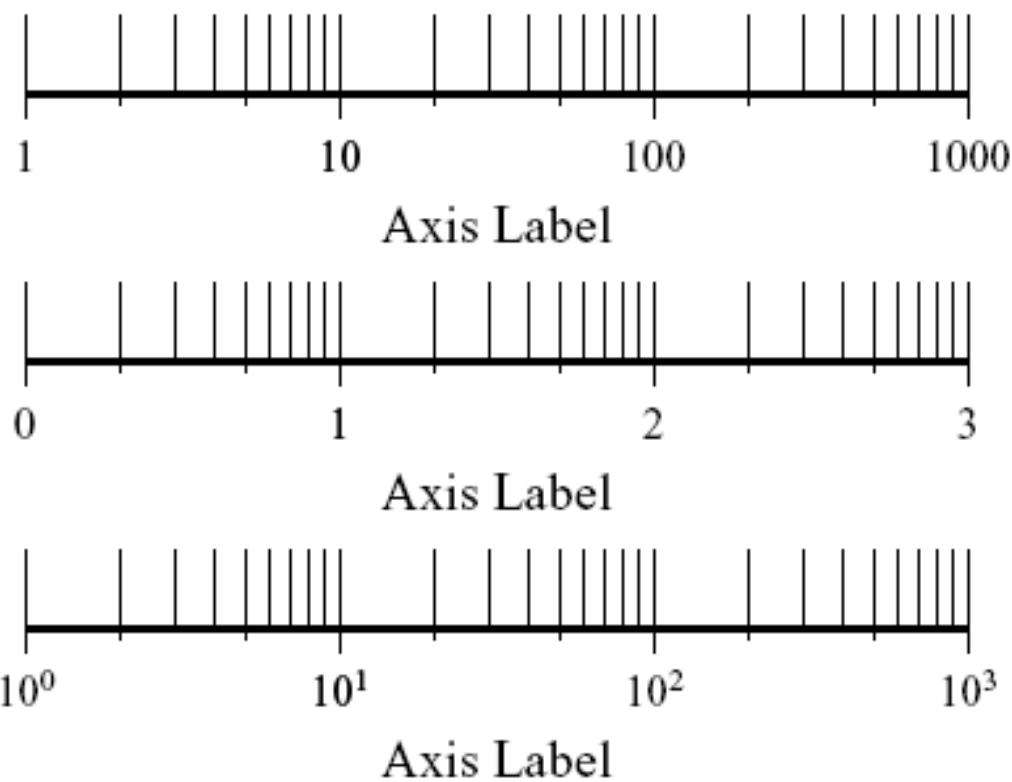


图 1.7: 使用不同刻度标记、刻度和网格间隔的对数坐标

图 1.7: 使用不同刻度标记、刻度和网格间隔的对数坐标。我们可以选择实际数值为刻度标记。参数值选 1，每个 10 的整数幂都标记；选 2，标记 10 的 0.1, 0.2, 0.5 次幂；选 3，标记 10 的每个 0.1 次幂。上图为-R1/1000/0/1 -JX3l/0.4 -Ba1 f2 g3；中图为 -Ba1 f2 g3 l；下图为 -Ba1 f2 g3 p.

指数轴选项 (Options for exponential axes)

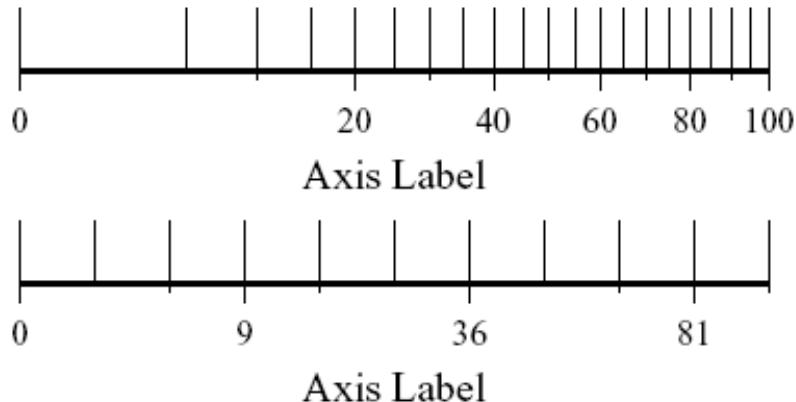


图 1.8: 指数或幂投影轴

刻度后加字母 p，刻度标记间隔是被转变后的单位，但标记仍将按照未改变

的单位来标。如果 `tick = 1`, 幂=0.5 (即开平方), 等刻度的标记将为 $1, -4, -9$, 等。

图 1.8: 指数或幂投影轴。上图用指数 0.5 做坐标轴。这里间隔标记为实际数值。写为 `-R0/100/0/1 -JX3 p0.5/0.4 -Ba20 f10 g5`. 下图, 虽然刻度标记使用相应的非投影值, 但间隔为投影值。写为 `-Ba3 f2 g1 p`

2.3.2 The `-c` option

`-c` 选项指定绘图拷贝数。[缺省为 1]

The `-H` option

`-H` 选项让 GMT 知道输入数据文件具有一个 (缺省) 或多个头纪录。如果多于 1 个头纪录, 一定要在 `-H` 选项后指定数字。如 `-H4`。见图 1.9。

2.3.3 The `-J?` options

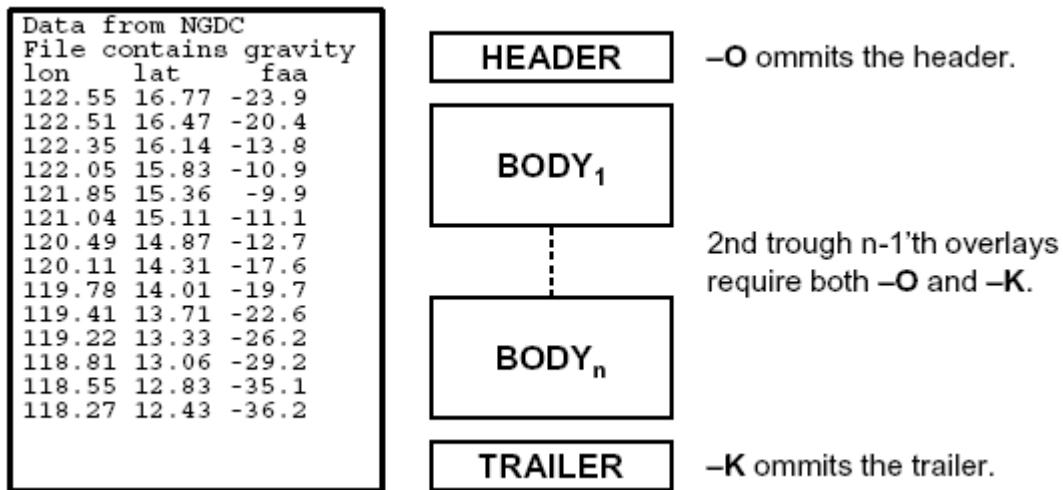


图 1.9

选择地图投影。`-J` 后面的代码决定投影类型。以用户单位指定地图宽度 (或轴长)。GMT 所允许的投影见图 1.10。该指南我们将选择下面投影之一 (全部 GMT 投影的详细资料见 `psbasemap` 主页) :

墨卡托投影 (Mercator:)

`-JMwidth.`

图 1.9: 输入文件可以有由 `-H` 指定的任意多个头纪录, 一个最髄的 *PostScript* 文件由多个单独的堆栈组成。

正形投影 (Orthographic:)

-JG $lon_0/lat_0/width.$ lon_0/lat_0 指定投影中心

Albers 圆锥投影 (Albers conic:)

-JB $lon_0/lat_0/lat_1/lat_2/width.$ 给出投影中心和 2 条标准平行线

Eckert 投影 (Eckert IV and VI:)

-JK[f|s] $lon_0/width.$ 给出中央子午线

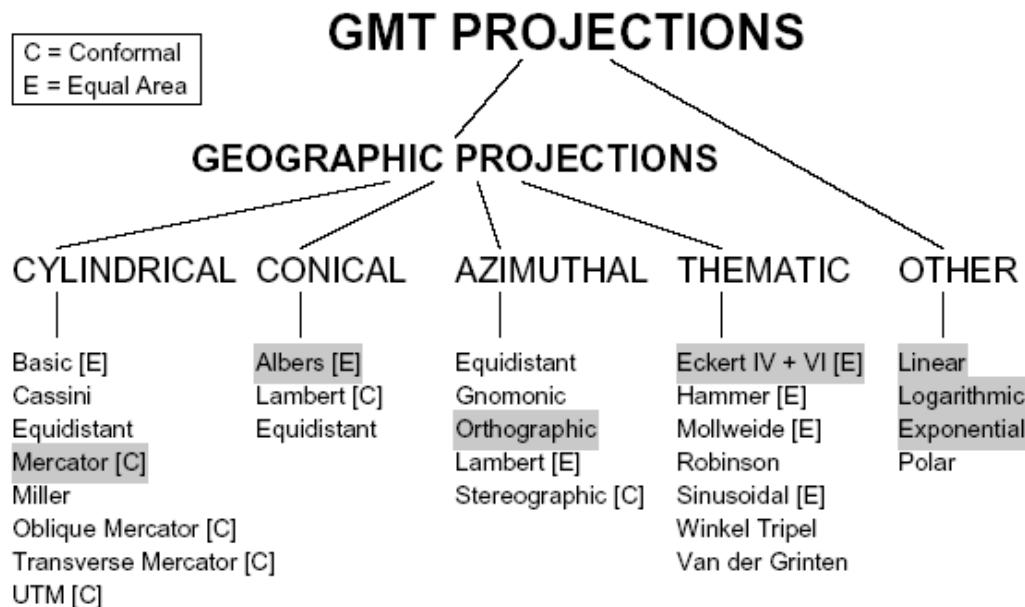


图 1.10: GMT 允许的 25 种地图投影

线性投影 (Linear:)

-JX $width/height.$ 给出绘图宽度[和高度]。宽度[和高度]可用下列方式之一给出:

1. **-JXwidth[d]**—标准线性比例。如果 x、y 用地理坐标的度来表示, 加 “d”。允许在 360° 范围内选择, 并自动加注度符号 “°”。
2. **-JXwidthl**—缩放前数值的 \log_{10}
3. **-JXwidthppower**—缩放前数值的指数 Raise values to power 可使用负的宽度[和高度]来改变轴的方向 (如 y 轴正值向下增加)。

2.3.4 The -K -O options

-K 和 **-O** 选项控制了多次覆盖绘图 PostScript 代码的形成。所有的 PostScript 文件都必须有一个头 (用来初始化), 一个文件体 (绘图), 和一个文件尾 (打印输出) 见图 1.10)。这样, 在覆盖几个 GMT 图时, 我们必须保证

第一个图省略了文件尾，所有中间的图同时省略了文件头和文件尾，而最后的图则省略了文件头。**-K** 省略了文件尾，说明更多的 *PostScript* 代码将会加到后面[缺省是结束绘图过程]。**-O** 选择覆盖绘图模式并省略了文件头[缺省是初始化一个新的绘图过程]。大多数多次覆盖绘图的未预料到的结果都可以追索到该选项的错误使用。

2.3.5 The **-P** option

-P 选择肖像模式。缺省为风景画模式，沿 PAPER_MEDIA 所选图纸宽度方向改变 x 轴原点，并逆时针旋转坐标系 90° 转变为肖像模式。缺省的 PAPER_MEDIA 设图纸类型为 Letter (或在 SI 单位为 A4)；当使用不同的图纸时，该值要改变，比如 11" x 17" 或更大的界面。(图 1.11)。

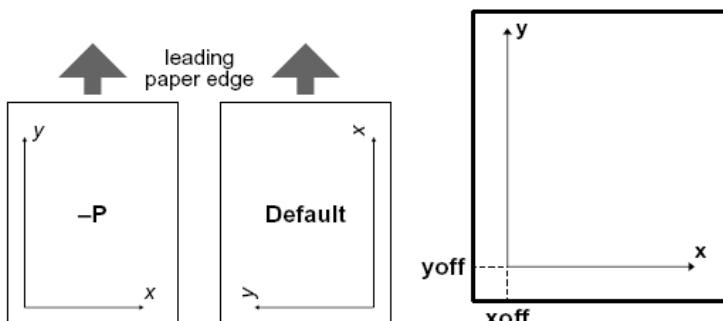


图 1.11 用户可以指定风景画模式

以通过**-X -Y** 自由改变。

2.3.6 The **-R** option

-Rxmin/xmax/ymin/ymax[r] 指定绘图范围。支持十进位或指数标注。用度、分 [和秒] 表示绘图范围，用 dd:mm[:ss] 格式。加字符 **r** 用左下角和右上角坐标来表示矩形绘图范围。(图 1.12)。

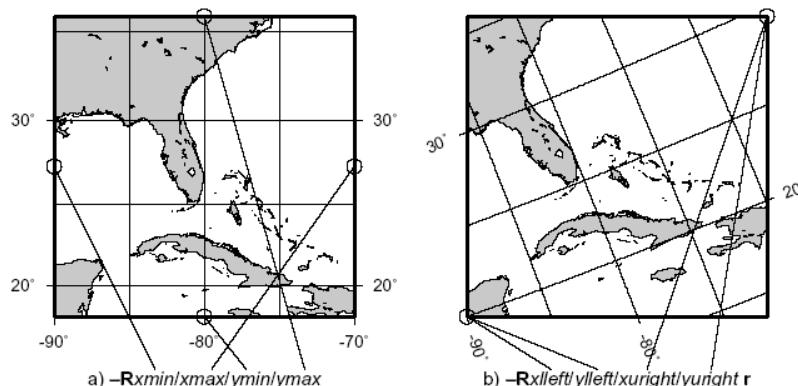


图 1.12：可用两种方式指定绘图范围

2.3.7 The -U option

-U 绘制 UNIX 系统时间标志。可在当前命令行加入一个任意的字符串(用双引号限定), 或 c 代码。(图 1.13)



图 1.13: -U 选项使标注绘图日期很方便

2.3.8 The -V option

-V 选择**verbose**模式, 将建立程序运行报告文件`stderr` [Default runs ``silently"].

2.3.9 The -X -Y options

-X 和 **-Y** 为以英寸为单位表示的投影坐标原点偏移量 ($xoff,yoff$) 对于新图缺省偏移量为 (1,1); 对于覆盖模式缺省偏移量为 (0,0)。默认所有的偏移都是对上一个原点而言 (见图 1.11)。绝对偏移 (即相对于图纸坐下角的固定点 (0,0)) 可以通过在偏移量后面加 “a” 来实现。除非使用这些选项改变了图纸原点, 以后的覆盖操作偏移方式同前面设定。偏移量用当前坐标系为单位 (可在开始时用**-P** 选项旋转图纸; 以后的覆盖操作将省略**-P** 选项) .

2.3.10 The -: option

对于地理数据, 通常认为第一列为经度, 第二列为纬度值。如果要调换经纬度顺序, 必须使用`-:` 选项。

2.4 练习 (Laboratory Exercises)

现在开始一些简单的绘制坐标轴和海岸线地图练习。按照所有重要的**-B**, **-J**, 和 **-R** 选项的顺序进行, 并熟悉几种 GMT 投影。要练习使用 GMT 命令 **psbasemap** 和 **pscoast**。

2.4.1 线性投影 (Linear projection)

我们开始绘制一个线性 x-y 投影的底图图框。x 轴由 10 到 70, 每 10 个单位标记; y 轴由 -3 到 8, 每 1 个单位标记, 最后的图框大小为 4x3 英寸。批处理文件如下:

```
psbasemap -R10/70/-3/8 -JX4i/3i-B10/1:"My first plot": -P >plot.ps
```

可用 `ghostview` 显示结果 `plot.ps`。

练习

1. 试改变-JX 值
2. 试改变-B 值
3. 省略-P

2.4.2 对数投影 (Logarithmic projection)

然后绘制一个对数 log-log 投影的底图图框。假定原始原始 x 的数据范围由 3 到 9613; y 的数据范围由 3.2×10^{20} 到 6.8×10^{24} 。批处理文件如下：

```
psbasemap -R1/10000/1e20/1e25 -JX9il/6il -B2:"Wavelength(m)  
":/a1pf3:"Power (W)":WS > plot.ps
```

练习

1. 试在轴长后不加 l
2. 由 -B 字串中去掉 p 字符
3. 在-B 后 “/” 的每一侧加上 g3

2.4.3 墨卡托投影 (Mercator projection)

尽管随纬度的增加存在明显的水平变形问题，但由于保证了区域地图位置的可靠性，等角墨卡托投影 (-JM) 还是被科学家所使用。这是 GMT 提供的几种圆柱投影之一。完整的语法很简单

-JMwidth

为了绘制海岸线图，我们使用命令 `pscoast` 时将自动访问基于 GSHHS 数据库 1.3 的海岸线数据。除常用的转换外，我们可能需要使用几个 `pscoast` 专用的选项（见表 1.2）。

表 1.2：绘制或覆盖海岸线图时的主要选项

选项	作用
-A	去除高分辨率图中的细部特点，即给出不绘制面积的上限
-D	选择数据分辨率（全 full, 高 high, 中 intermediate, 低 low, 或粗 crude）
-G	设定陆地颜色（缺省为无色）
-I	绘制河流（可选择不同等级的精细程度）
-L	绘制地图比例尺（长度比例尺可以是公里、英里或海里）
-N	绘制国家的行政区划（包括美国的州界限）
-S	设定水体颜色（缺省为无色）
-W	绘制海岸线并设定线宽

-W, -G, -S 必选其一。我们首先练习的例子是拉丁美洲
`pscoast -R-90/-70/0/20 -JM6i -P -B5g5 -G180/120/60 > map.ps`
 练习

1. 增加**-V** 选项
2. 试改用**-R270/290/0/20**, 标注发生什么改变?
3. 编辑你的.gmtdefaults 文件, 分别把 DEGREE_FORMAT 的参数由 0 变到 4。
4. 选择另一个区域并改变陆地颜色
5. 选择另一个包括北极或南极的区域
6. 试用 **-W0.25p** 代替 (或增加) **-G**.

2.4.4 艾伯特投影 (Albers projection)

Albers 投影 (**-JB**) 是一种等积圆锥投影, 它的等角同类投影是兰伯特 (Lambert) 圆锥投影 (**-JL**)。它们的用法几乎相同, 这里我们只以 Albers 投影为例。一般的语法是

-JB lon₀/lat₀/lat₁/lat₂/width

这里 (lon_0, lat_0) 是地图 (投影) 中心, 而 lat_1, lat_2 是两条标准平行线, 即圆锥面和地球表面的交线。使用以下命令:

```
pscoast -R-130/-70/24/52 -JB-100/35/33/45/6i -B10g5:"Conic Projection": -N1/2p  
-N2/0.25p -A500 -G200 -W0.25p -P > map.ps
```

练习

1. 改变 GRID_CROSS_SIZE, 用十字线代替网格线
2. 改变**-R** 为矩形图框说明, 代替原来的最小和最大经纬度值。

2.4.5 正交投影 (Orthographic projection)

方位角正交投影 (**-JG**) 是几种具有类似语法和特点的投影之一, 可以想象为从空间无穷远处来看地球。该投影的语法为:

-JG lon₀/lat₀/width

这里 (lon_0, lat_0) 是地图 (投影) 中心, 试用下命令:

```
pscoast -R0/360/-90/90 -JG280/30/6i -Bg30/g15-Dc-A5000 -G255/255/255  
-S150/50/150 -P > map.ps
```

练习

1. 试用矩形图框选项**-R** 绘制矩形图框美国地图。

2.4.6 Eckert IV 和 VI 投影

我们用 Eckert IV 和 VI 投影 (**-JK**) 来结束地图投影的简介。这是全球专题地图所使用投影类型中的两种，他们都是等积投影，语法是：

-JK[f|s] lon₀/width

这里 **f** 是 Eckert IV (4) 而 **s** (Default) 是 Eckert VI (6)。lon₀ 为中央子午线 (其优先级高于根据**-R** 范围中间值所得到的经度值)。一个简单的 Eckert VI 世界地图可由下面命令创建

```
pscoast -R0/360/-90/90 -JKs180/9i -B60g30/30g15 -Dc -A5000 -G180/120/60
-S100/180/255 -W0.25p > map.ps
```

练习

1. 以格林威治为中央子午线
2. 用**-L** 加入地图比例尺

2.5 第二部分 (SESSION TWO)

2.5.1 一般信息 (General Information)

有 17 条直接用于创建或修改图形的 GMT 命令 (表 2.1)，还有 43 条命令主要与数据处理有关。这一节将重点讨论在图上绘制线、符号和文本。我们将扩大我们已经获得的关于各种 GMT 地图投影、选择数据域和边界标注等方面的技巧。

表 2.1: GMT 所有 1-D 和 2-D 绘图命令列表

命令	作用
底图	
psbasemap	创建一个具有任意尺度的空的底图框架
pscoast	绘制海岸线、填充大陆、绘制河流和行政边界
点和线	
pswiggle	沿 (x,y) - 轨迹绘制空间 - 时间序列曲线
psxy	在 2-D 图形中绘制符号、多边形和线
psxyz	在 3-D 图形中绘制符号、多边形和线
柱状图	
pshistogram	绘制矩形柱状图
psrose	绘制极坐标图 (扇形图/玫瑰花图)
等值线图	
grdcontour	由 2-D 栅格数据绘制等值线
pscontour	使用最佳三角形法直接由 xyz 数据绘制等值线或影像图
表面图	

grdimage 由 2-D 网格数据构筑彩色影像图
grdvector 由 2-D 网格数据绘制矢量场
grdview 由 2-D 网格数据显示 3-D 透影视像

几个有用的命令

psclip 使用多边形文件初始化用户剪裁路径
psimage 绘制 Sun 光栅图像文件
psmask 创建剪裁路径或覆盖模板? *Create clipping paths or generate overlay to mask*
psscale 绘制灰度或彩色图例条
pstext 在图上写字串

绘制线和符号, **psxy** 在 GMT 中是最常用的命令。除了常用的命令行转换开关外, 它还有大量专用的选项, 以及根据所选用的不同符号而使用不同的文件格式。这些情况使 **psxy** 命令比 GMT 的大多数工具更难精通。表 2.2 为 **psxy** 完整的选项列表

表 2.2: **psxy** 命令可选转换开关项

选项	作用
-A	限制沿大圆弧进行线性内插
-C cpt	由 z-值和 <i>cpt</i> 文件确定符号颜色
-E[x][y][cap]/[pen]	绘制具有指定属性的误差棒
-G fill	设定符号或多边形的充填颜色
-L	构成封闭多边形
-M[flag]	多段输入数据, 数据头由 <i>flag</i> 开始
-N	在地图边界仍保留完整符号
-S[symbol][size]	选择 16 种符号之一 (见表 2.3)
-W pen	设置线或符号边框的画笔属性

符号可以是空心的 (只用 **-W**, 不用 **-G**) 或实心的 (**G**, 符号边框选项用 **-W**)。
-S 选择所需符号代码和给出符号大小信息。如果没有用 **-S** 选择符号, 则认为在输入数据文件的最后一列给出。每个符号的大小可以任选, 数值由输入文件提供。可用的 15 个符号列表如下:

表 2.3: **psxy** 可选符号。小写字符 (**a, c, d, h, i, s, t, x**) 表明符号在给定直径的圆之内。大写字符 (**A, C, D, H, I, S, T, X**) 表明符号的面积等于给定直径的

圆的面积。

选项	符号
-Sa <i>size</i>	星号, <i>size</i> 为外接圆的半径
-Sb <i>size[/base][u]</i>	条棒, <i>size</i> 棒宽, 加 u 表示 <i>size</i> 用 x-轴单位; 棒由 <i>base[0]</i> 延长到 y-值
-Sc <i>size</i>	圆, <i>size</i> 为直径
-Sd <i>size</i>	菱形, <i>size</i> 为边长
-Se	椭圆, 方向 (<i>direction</i> , 由水平方向逆时针计)、以英寸做单位的长轴 <i>major</i> 、和短轴 <i>minor</i> 由输入数据文件读取。
-SE	椭圆, 方位角 <i>azimuth</i> (由垂直方向顺时针计), 以公里做单位的长轴 <i>major</i> 、和短轴 <i>minor</i> 由输入数据文件读取。
-Sf <i>gap/tick[l r L R]</i>	断层, <i>gap</i> 和 <i>tick</i> 设置短线的长度和间隔, 如 <i>gap < 0</i> 即指短线编号; l 或 r 表示短线画在断层线的左侧或右侧 [缺省画在线上], 大写 L 或 R 表示用三角形代替短线。
-Sh <i>size</i>	六边形, <i>size</i> 为边长
-Si <i>size</i>	倒三角形, <i>size</i> 为边长
-Sl <i>size/string[% font]</i>	字符, <i>size</i> 为字体大小; 增加一个字符或一个字串, 字体可任选。
-Sp	点, 无须设大小 (当前分辨率下的 1 个像素)
-Ss <i>size</i>	正方形, <i>size</i> 为边长
-St <i>size</i>	三角形, <i>size</i> 为边长
-Sv [<i>thick/length/width</i>][n <i>norm</i>]	箭头, 方向 <i>direction</i> (由水平方向逆时针计) 和长度 <i>length</i> 由输入数据文件读取。可选项, 设定箭柄的宽度 <i>thick</i> , 箭头的宽度 <i>width</i> 和长度。如果增加 n <i>norm</i> , 则所有长度小于 <i>norm</i> 的箭头的属性都依 <i>length/norm</i> 成比例。
-SV [<i>thick/length/width</i>][n <i>norm</i>]	箭头, 用方位角 <i>azimuth</i> (由北向东旋转的角度) 代替方向 <i>direction</i> 。地图的角度根据所选的地图投影来计算。
-Sw [<i>size</i>]	扇形, 起点和终点的方向 (由水平方向逆时针计) 由输入数据文件读取。
-Sx <i>size</i>	十字, <i>size</i> 为十字线的长度。

因为有些符号需要更多的输入数据, 符号的大小和颜色都可以由输入数据文

件确定，数据的格式可能会引起混淆。通用的输入数据的格式为（可选项用方括号表示）：

x,y[z][size][σ_x][σ_y][symbol]

只有前两列是输入文件所必需的，它们一定是经度和纬度（或 *x* 和 *y*）。其他可选项在遇到以下情况之一时使用：

1. 如果你打算分别设定每个符号的颜色，用**-C** 提供一个 cpt 文件，增加数据文件第 3 列，让 *z* 值在 cpt 文件所取值范围内。
2. 如果你打算分别设定每个符号的大小，在第 4 列给出符号大小。
3. 为绘制误差棒，使用**E** 选项，并增加 1 列或 2 列数据，分别为 $\pm dx$ 和 $\pm dy$ 值。**-E** 的形式决定需要 1 列（**-Ex** 或 **-Ey**）还是 2 列（**-Exy**）数据。
4. 如果你使用**-Sv**（或**-SV**）绘制箭头，则 **size** 实际上为 2 列，包括了每个箭头的方向 *direction*（或方位 *azimuth*）和长度。
5. 如果你使用**-Se** 绘制椭圆，则 **size** 实际上为 3 列，包括了每个椭圆的方向 *direction*，以英寸做单位的长轴 *major* 和短轴 *minor*（用 **-SE** 时，改为方位角 *azimuth*，轴长以公里做单位）。

在做练习之前，我们需要讨论两个关键的转换开关选项，它们是指定笔属性和符号及多边性的填充。

2.5.2 指定笔属性 (Specifying pen attributes)

GMT 中的画笔有 3 个属性，宽度 *width*、颜色 *color* 和结构 *texture*。多数命令接受下列笔属性选项：

-Wwidth[/color][texture][p]

Width 通常用当前设备分辨率作为测量单位（即 *.gmtdefaults* 文件里的 **DOTS_PR_INCH**）。因此，如果 *dpi* 设为 300，每个单位就是 1/300 英寸。加 **p** 指定笔宽用点（1/72 英寸）做单位。注意，根据你的设备的 *dpi* 的不同，笔宽为 5 可有不同的物理宽度，而 **5p** 则总是 5/72 英寸。最细笔宽可以通过设定 0 宽度来实现，但实际物理宽度取决于设备分辨率。

color 可以指定为灰度，范围 0–255（由黑色到白色线性变化），或采用 RGB 系统，你可以设定 *r/g/b*，每个颜色的范围 0–255。这里 0/0/0 为黑色，而 255/255/255 为白色。

texture 属性控制了线的结构。绘制点线，在 *width* 和 *color* 后简单加“**to**”；绘制虚线加“**ta**”。为精确描述你所需要的线型，可以使用“*tstring:offset*”，这里 *string* 是被下划线所分割的一些整数。这些数字分别代表了线段的长度和线段之间间隔的长度。*offset* 指定线起点的偏移量。例如，如果你想要画一条 2 个单位

宽的黄色虚线，其构成依次是 20 个单位的长线段，10 个单位间隔，再接 5 个单位的短线段，10 个单位间隔；由距离原点 10 个单位起开始画。写为：

-W2/255/255/0t20_10_5_10:10

这里，**texture** 单位可以指定为 dpi 或 points。

2.5.3 指定填充属性 (Specifying fill attributes)

许多绘图命令允许用户填充多边形或符号。有两种形式的填充：

-Gfill

-Gdpi/pattern[:Br/g/b[Fr/g/b]]

第一种形式，我们可以指定为灰度（0–255）或颜色（r/g/b 范围 0 –255），类似于线的颜色设置。第二种形式，允许我们使用预定的位图模式。**pattern** 可以是 1 – 90 范围的数字，或 1–、8–或 24–位的 Sun 光栅文件的名字。前者将产生 GMT 提供的 90 种预定义的 64 x 64 位图案之一，图案在 GMT 的技术参考附录 E 中。后者允许用户使用标准 Sun 光栅文件创建一个定制的重复图像。**dpi** 参数设置页面中该图像的分辨率；由一系列这种“瓷片”镶嵌填满面积。如指定 **dpi** 为 0 将产生 **.gmtdefaults** 文件设定的当前可以得到的最高分辨率。如用大写字母**-GP** 代替**-Gp**，图像将反相，即黑白像素交换（只能用于 1-bit 图像或预设的位图图案）。对于这些图案和其它 1-bit 图像可以分别指定背景色和前景色，增加 **:Br/g/b[Fr/g/b]** 它们将分别代替缺省的白色和黑色像素。设置一种前景或背景色在只显示前景或背景像素的地方可以产生一种透明图像效果。由于 *PostScript* 执行的限制，使用**-G** 的光栅图像大小一定要小于 146 x 146 像素。大的图像见 **psimage**。Sun 光栅文件的格式在 GMT 的技术参考附录 B 中有描述。注意在 *PostScript Level 1* 下，图案的填充采用多边形回型路径的方式。复杂的回型路径要求比 *PostScript* 解释程序所分配的内存更多。因此有可能导致某些 *PostScript* 解释程序（特别是支持老式激光打印机的程序）出现内存溢出和中断。所以我们推荐你使用标准的灰度填充来代替图案。

范例

使用 GMT 的 **minmax** 命令将得到数据范围的极值

返回

data: N = 7 <1/5> <1/5>

告诉我们数据文件有 7 条纪录，前两列分别是最小值和最大值。使我们了解使用 **-R** 和 **-JX** 选项时如何设计线性投影的范围。试做以下练习：

1. 绘制 0.3 英寸大的空心圆。

2. 改为用白色填充的圆。
 3. 绘制 0.5 英寸的星形，圆周为宽 1.5p 的红色虚线。
- 为简化线的绘制，我们不选择任何符号并指定线宽：

```
psxy data -R -JX -P -B -W0.5p > plot.ps
```

练习

1. 绘制一个绿蓝色的多边形。
2. 试用预制的图案。

一个常见的问题是：“怎样才能用 **psxy** 命令绘制一个与线型相关的符号？”回答是，我们必须调用 **psxy** 命令两次。主要因为被剪贴的多边形要保存在内存中，而我们能绘制多大的多边形要受计算机 RAM 的限制。换句话说，符号一次可绘制一个，但画多少次并没有限制。因此，为了把符号与线属性相联系，我们必须使用覆盖方式：

```
psxy data -R -JX -B -P -K -W0.5p > plot.ps
psxy data -R -JX -O -W -Si0.2i >> plot.ps
```



最终的 **psxy** 命令包括了比较复杂的内容，我们打算在一个以海岸线为背景的底图上绘制几次地震的震中位置。我们想用符号的大小来反映地震的震级，用符号的颜色表示震源的深度。练习的目录下可以发现两个数据文件 `quakes.ngdc` 和 `quakes.cpt`。文件 `quakes.ngdc` 的前几行格式如下：

Historical Tsunami Earthquakes from the NGDC Database

Year	Mo	Da	Lat+N	Long+E	Dep	Mag
1987	01	04	49.77	142.29	489	4.1
1987	01	09	39.90	141.68	067	6.8

该文件有 3 行头记录（包括空行）但我们只对每行的第 5, 4, 6, 7 列感兴趣。除提取数据外，我们还要在震级和以英寸做单位的符号的大小之间建立比例关系。了解震级的范围后，选 0.02 作为系数乘以震级得到的符号大小比较合适。有多种方法可以重新规定输入文件的格式以使其符合 **psxy** 的输入格式，包括手工编辑，使用 MATLAB 或 UNIX 工具。我们可以简单地使用 UNIX 工具 awk 来做这件事，但考虑到我们只介绍 DOS 批处理文件，重点介绍相应的命令 gawk：

```
gawk “{if (NR > 3) print $5, $4, $6, 0.02*$7}” quakes.ngdc > quakes.d
```

执行这条命令的过程是，程序打开数据文件 `quakes.ngdc`，根据条件语句，跳过前 3 行，由第 4 行开始读取，依次读取需要的第 5, 4, 6 和 7 列，同时对第 7 列的数据乘以系数 0.02。输出文件 `quakes.d` 格式如下：

```
142.29 49.77 489 0.082
```

141.68 39.90 067 0.136

...等等

4列数据顺序为震中的经度(x坐标)、纬度(y坐标)、震源深度和表示震级符号的直径(单位英寸)。下面我们将按照地震学传统的颜色方案,把浅源地震表示为红色(深度0–100 km),中源地震表示为绿色(深度100–300 km),深源地震表示为蓝色(深度>300 km)。

文件quakes.cpt在深度和颜色之间建立联系:

```
#z0 red green blue z1 red green blue
0 255 0 0 100 255 0 0
100 0 255 0 300 0 255 0
300 0 0 255 1000 0 0 255
```

除了注释行外(由#开始) cpt文件的每一条记录都决定了z值落在 z_0 和 z_1 之间的符号的颜色。如果一条记录 z_0 和 z_1 两组的R/G/B不一致,那么将根据z值的大小进行线性内插得到中间色。我们在这里选择的是颜色固定的间隔。

我们现在使用Mercator投影完成我们的例子,例中还绘制了地图比例尺:

```
pscoast -R130/150/35/50-JM6i-B5-P-G200-Lf134/49/42.5/500-K > map.ps
psxy-R-JM-O-Cquakes.cpt quakes.d -Sci-W0.25p >> map.ps
```

在-Sc选项后增加i是为了保证符号的大小以英寸为单位。

练习

1. 选择其它符号
2. 把深源地震的颜色由蓝色改为灰色

2.5.4 Plotting text strings

许多情况下我们需要用文本来注释图,在GMT中使用**pstext**命令来实现。除了一般的命令开关外,还有7个选项要用(表2.4)。

表2.4:一些最常用的**pstext**选项

选项	作用
-C <i>dx/dy</i>	设定文本和文本框之间的间隔(见 -W)
-D <i>dx/dy</i>	设定字串投影位置偏移量
-G <i>fill</i>	设定文本颜色
-L	列出字体标识符并退出
-N	保持图边界符号完整
-S <i>pen</i>	选择空心字体并设置画笔属性
-W [<i>fill</i>][o [<i>pen</i>]]	打印文本框,Paint the text box; draw the outline if o is appended

(also see -C)

用 **pstext** 输入数据可望含有以下信息:

x y size angle fontno justify text

7个参数的意义为:

x y X 坐标, Y 坐标;

size 以点做单位的字体大小;

angle 文本基线和水平线之间的角度
(逆时针测算);

fontno 选用字符集代码;

justify 为字串对齐方式选择, 设定文本

串相对于 x、y 坐标的位置。相应的代码为 1~LT, 2~CT, 3~RT, 4~LM, 5~CM,
6~RM, 7~LB, 8~CB, 9~RB;

text 为字符串或待标注文本。图 2.2 说明这些概念并显示 *justify* 选项所用的双字符代码。

文本串可以是一个或几个单词, 也可以包括特殊字符的 8 进位代码, 以及用于选择或中断下标或符号字体的切换符。以下是经过验证的切换符:

表 2.5: GMT 文本选择或中断切换符

@+	上标使用转换开关
@-	下标使用转换开关
@#	小型大写字符使用转换开关
@!	由后面两个字符创建一个合成符号
@@	打印符号@本身
@E @e	Æ æ
@O @o	Ø ø
@A @a	Å å

注意这些选择或中断切换符(以及十进位代码)可以被用在 GMT 程序的任何地方, 包括-B 选项中。在 GMT 技术参考手册的 Appendix F 可以找到一个十进位编码列表。为了得到带重音的欧洲字符, 必须要在你的.gmtdefaults 文件中把 WANT_EURO_FONT 设为 TRUE 。

我们将用以下的程序验证 **pstext** 命令:

```
cat << EOF | pstext -R0/7/0/7 -JX7i -P -B1g1 -G255/128/0 | ghostview -
```

```

1 1 30 0 4 BL It           's P@al, not Pal!
1 2 30 0 4 BL Try @%33%ZapfChancery@%% today
1 3 30 0 4 BL @~D@~g@      -b@- = 2@~pr@~G@~D@~h.
1 4 30 0 4 BL University of Hawaii at M@!a\305noa
EOF

```

这里我们已经使用了 *UNIX* 的即时文本命令: << EOF 将顺序处理把下面各行作为输入文件, 直到发现 EOF 为止。用管道命令符号 “|” 使 *PostScript* 直接到 *ghostview* (符号- 通知 *ghostview* 发现管道)。

练习

1. At $y = 5$, add the sentence `` $z^2 = x^2 + y^2$ ''.
2. At $y = 6$, add the sentence ``It is 80° today''.

编者注: 也可以借助其它可以读取.ps 文件的图形处理软件, 比如 CorelDRAW, 来给 GMT 绘制的图形增加包括汉字在内的文字注释。这要比用 GMT 的命令 **pstext** 方便得多。具体使用方法是, 用 CorelDRAW 导入 GMT 完成的.ps 文件, 用加入文本块的方式添加所需要的汉字及各种字符串, 存盘和打印。

2.5.5 由网格数据组构筑等值线 (Contouring gridded data sets)

GMT 提供了几种方法创建网格数据组, 我们将讨论其中的两种。首先, 假设我们已经有了网格数据组。在 GMT 补充文件中有一个程序, 从几个公共的数据中已经提取了几种全球的网格数据组, 其中包括 ETOPO5, 地壳年龄, 重力和大地水准面, 以及美国大陆的 DEM 数据。这里, 我们将使用 **grdraster** 命令来提取 1 个 GMT 已有的网格数据组, 以后用于构筑等值线图:

`grdraster 1 -R-66/-60/30/35 -Gbermuda.grd -V`

我们先用 GMT 的 **grdinfo** 命令来看看文件 bermuda.grd 的信息:

`grdinfo bermuda.grd`

这个文件包含了百慕大地区的测深数据, 深度由-5475 米到-89 米。我们打算用这些数据来绘制等值线图, 由 **grdcontour** 命令来实现。和前面的绘图命令一样, 我们要用选项-J 来设定地图投影类型。这里我们没有指定绘图地区, 因为已经由网格文件缺省给出了范围。为了完成绘制任务, 我们还必须补充有关所绘等值线的信息。**grdcontour** 是一个复杂的命令, 含有很多的选项, 最有用的列表如下:

表 3.1: **grdcontour** 最有用选项

选项	作用
-A anot_int	标记等值线间距
-C cont_int	非标记等值线间距
-G gap	设定标记等值线两个标记之间的间隔
-L low/ high	只绘制 <i>low</i> 到 <i>high</i> 范围内的等值线
-N unit	给标记等值线的标记后加单位 <i>unit</i>
-Q cut	不画少于 <i>cut</i> 个节点的等值线
-S smooth	重新采样平滑等值线，用网格大小/平滑系数作为新网格的增量
-T[+ -][gap/ length][: LH]	给最里面的封闭等值线绘制短线，短线起点在等值线，朝向低值方向。 设定短线的间距和短线长度，在封闭等值线的中心书写字符。
-W[a c] pen	设置非标记等值线和标记等值线间距的线属性
-Zfactor[/ offset]	进行处理之前，先减去偏移量并用参数乘以数据值
我们首先画一个简单的等值线图，设标记等值线间距为 1 km，非标记等值线间距为 250 m。投影类型为墨卡托投影，边框宽 7 英寸，每 2 度标记：	
<code>grdcontour bermuda.grd -JM7i -C250 -A1000 -P -B2 ghostview -</code>	
练习	
1. 用 -S4 对等值线进行平滑	
2. 用 -T 对所有的高地和低地做 tick 标记	
3. 用 -Q10 略去微小的地形特点	
4. 用 -R-70/-60/25/35 代替该地区	
5. 试用另一地区剪裁我们的数据域	
6. 绘制数据比例尺（单位 km），并把加到标记等值线数值后面	

2.5.6 由任意的空间数据构筑等值线

除了上述的已有网格文件的情况外，在我们绘制等值线图和彩色代码图之前，必须先把我们的数据转换为 GMT 能够读出的正确格式。我们先要区别两种情况：

- 可得到的 (x, y, z) 数据是规则的网格数据
- 在空间不均匀分布的 (x, y, z) 数据

前一种情况只需要对数据进行简单的重定格式（用命令 **xyz2grd**），而后一种情

况就必须要内插为规则的网格，这个过程就是所谓网格化。GMT 提供了 3 种不同的网格化方法。这里，我们讨论最常用的两种。

所有的 GMT 网格化程序都需要用户指定网格化的范围和输出文件名字。

- R *xmin/xmax/ymin/ymax*** 需要网格化的范围
- I *xinc[m|c]/yinc[m|c]*** 网格间隔（加 m 或 c 为以弧度表示的分或秒）
- G *gridfile*** 输出文件名字

2.5.7 最近邻点网格化法 (Nearest neighbor gridding)

GMT 程序 **nearneighbor** 执行一种简单的“最近邻点”平均操作。当数据密度比较大时，这是优选的网格化数据的方法。

nearneighbor 是一种局部操作，它只考虑临近待计算节点的数据控制点，只使用在搜寻半径之内的数据点。为了给节点赋值，需要增加的条件是节点周围 n 个扇区中每个的每一个至少要有一个数据点。计算节点值的方法是，在搜寻半径之内对每个扇区内的最近数据点进行加权平均，每个点的权重根据节点到数据点的距离按照下式计算：

$$\bar{z} = \frac{\sum_{i=1}^n z_i w_i}{\sum_{i=1}^n w_i} \quad w_i = \left(1 + \frac{9r_i^2}{R^2}\right)^{-1}$$

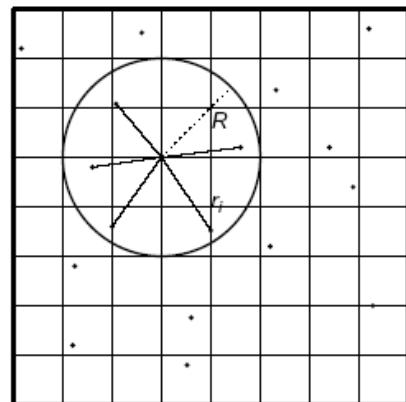


图 3.1：邻点搜寻的几何表示

最重要的转换开关如表 3.2 所列。

表 3.2：最近邻点搜寻法所使用的转换开关

选项	作用
-S <i>radius[k]</i>	设置搜寻半径。加 k 表示半径的单位为千米[缺省为 x-单位]
-E <i>empty</i>	给空白节点赋值[缺省为 NaN]
-N <i>sectors</i>	设置搜寻扇区数[缺省为 4]
-W	由输入数据的第 4 列读相对权重

我们用文件 ship.xyz 来练习数据网格化，该文件包含了加州 Baja 外船载测深仪的测量数据。我们希望计算 5' x 5' 的网格。运行 **minmax** 得到
ship.xyz: N = 82970 <245/254.705> <20/29.99131> < -7708/-9>
据此我们可以选择适当的网格化范围：

nearneighbor -R245/255/20/30 -I5m -S40k -Gship.grd -V ship.xyz

用以下命令来生成等值线图

grdcontour ship.grd -JM6i -P -B2 -C250 -A1000 | ghostview -

练习

1. 试用 100 km 的搜寻半径和 10' x 10' 的网格

2.5.8 张性样条网格化法 (Gridding with Splines in Tension)

作为另一种选择，我们可以使用一种全局的操作来网格化我们的数据。这个过程通过程序 **surface** 来实现，它是一种经过改良的标准最小曲率算法，允许用户把张量引入表面。物理意义是迫使一个连续弹性薄板通过所有的数据点；节点位置上的表面的值即为网格数据。数学上，我们想找到一个函数 $z(x, y)$ 要满足以下约束条件：

$$\begin{aligned} z(x_k, y_k) &= z_k, \quad \text{for all data} & (x_k, y_k, z_k), \quad k=1, n \\ (1-t)\nabla_z^4 - t\nabla_z^2 &= 0 \quad \text{其它} \end{aligned}$$

这里 t 是“张力”， $0 \leq t \leq 1$ 。基本上，随着 $t \rightarrow 0$ ，我们得到最小曲率解；而随着 $t \rightarrow \infty$ ，接近一个和谐解 (**which is linear in is linear in cross-section**)。该程序的一些最重要的转换开关如表 3.3 所列。

表 3.3: **surface** 的一些选项

选项	作用
-A aspect	对向各项异性的网格设置方向比值
-C limit	设置收敛极值。缺省为数据范围的 1/1000
-T tension	设值张性系数 [缺省为 0]

2.5.9 预处理 (Preprocessing)

surface 程序假定数据已经被预处理过，消除了混淆现象。为使我们必须保证这一步在网格化之前完成，GMT 提供了 3 种预处理的方法，分别为命令 **blockmean**、**blockmedian** 和 **blockmode**。第一个命令对网格间隔框内的值取平均值，第二个命令返回中值，后一个命令返回众值。作为一般规律，我们对于大多数平滑数据（比如位场 potential fields）采用平均数，而对于比较粗略的，非高斯数据（比如地形数据）采用中值或众值。除了要求的-R 和 -I 转换开关外，这些预处理采用同样的选项：

表 3.4: 一些预处理选项

选项	作用
-N	选择像素配准[缺省为网格线]
-W[i o]	加 i 或 o 表示由第 4 列读出或写入权重 如果我们用中值法对上例的船载测量数据进行预处理，写为： blockmedian -R245/255/20/30 -I5m -V ship.xyz > ship_5m.xyz 输出数据现在可以用到 surface 中 surface ship_5m.xyz -R245/255/20/30 -I5m -Gship.grd -V 如果我们用新的网格文件 再次运行 grdcontour ，将会发现与用 nearneighbor 生成的网格绘制的等值线图明显不同。因为 surface 是一种全局方法，它评价的是所有节点的解，甚至没有任何数据的约束。有大量可以使用的选项：
	<ol style="list-style-type: none"> 1. 我们可以把距数据点过远的节点规定为 NaN 值 2. 我们可以把等值线不可靠的区域填充为白色。 3. 我们可以绘制覆盖了绝大多数非限定区（并非所有地区）的大陆界线。 4. 我们可以设定一个剪裁路径，只显示被限定区域内的等值线。
	现在来探讨后面这种方法。 psmask 命令可以读同样的预处理数据并根据数据的分布设置等值线的 mask 。一旦剪裁路径被激活，我们可以绘出最后等值线区的轮廓，然后第二次调用 psmask 来最终解除剪裁。表示如下： psmask -R245/255/20/30 -I5m ship_5m.xyz -JM6i -B2 -P -K -V > map.ps grdcontour ship.grd -JM -O -K -C250 -A1000 >> map.ps psmask -C -O >> map.ps

练习

1. 用你选的任何颜色为陆地着色

2. 把剪裁路径染为浅灰色 (t 在第一个 **psmask** 调用中使用-G) .

2.5.10 cpt 文件格式 (The cpt file format)

在前面于地震相关的范例中，我们已经简单地介绍过 cpt 文件。这里我们将进行比较深入地讨论。cpt 文件的一般格式为：

$z_0 \ R_{\min} G_{\min} B_{\min} z_1 R_{\max} G_{\max} B_{\max} [A]$

...

$z_{n-2} R_{\min} G_{\min} B_{\min} z_{n-1} R_{\max} G_{\max} B_{\max} [A]$

cpt 文件可以只表示灰度（在表中只使用红色一列），绿色和蓝色列选项只用于彩色 cpt 表。最后一列选项可用来标注彩色图例棒（由 **psscale** 创建）, **U**, **L**, 和 **B** (在位置 A) 表示我们打算把标记写在彩色图例棒的上面，下面还是分别写在两边。你也可以按同样的方式使用 **psscale -B** 选项。比如在 **psbasemap** 中。

Cpt 文件可以用任何方法创建。GMT 提供两种模式：

1. 创建一个简单的、线性的颜色表作为主色表（有几种是内置的），确定拟表示的颜色边界的 z 值 (**makecpt**)
2. 由内置主 cpt 颜色表创建颜色表，在一个网格数据文件中 z 值的直方图均衡分布。（**grd2cpt**）

也可以手工，或用 **gawk** 或其它工具来创建这些文件。但我们这里的讨论只限于 **makecpt**。主要的变量是主色表的名字（如果你运行一个没有任何 cpt 文件的程序，将显示一个 cpt 文件列表）和与其相应的等距的 z 值。主要选项如下：

表 4.1: **makecpt** 允许的主要选项

选项	作用
-C	给出所调用的主 cpt 文件名
-I	倒转颜色级数指向
-V	在 verbose 模式下运行
-Z	构筑一个连续的而不是离散的表

下面练习是分别创建离散的和连续的彩色 cpt 文件，数据范围由 -20 到 60，每 10 个单位改变一次颜色。试分析两种表示的区别：

makecpt -Crainbow -T-20/60/10 -> disc.cpt

makecpt -Crainbow -T-20/60/10 -Z > cont.cpt

我们可以用 **psscale** 命令绘制这些色谱表。表 4.2 所列的选项值得注意：

表 4.2: **psscale** 的主要转换开关和选项

Option	Purpose
-C cptfile	必需的 cpt 文件
-D xpos/ ypos/ length/ width[h]	设定色谱图例的左侧中心坐标和图例的长 / 宽。 加 h 改为水平图例并给出顶部中心坐标
-I max_intensity	增加照明效应
此外, -B	选项可用来绘制标题和单位符号 (以及随意设定色谱图例的标注, 记号和网格线宽度)
psbasemap -R0/8.5/0/11 -Jx1i -P -B0 -K > bar.ps	
psscale -D3i/3i/4i/0.5ih -Cdisc.cpt -B:discrete: -O -K >> bar.ps	
psscale -D3i/5i/4i/0.5ih -Ccont.cpt -B:continuous: -O -K >> bar.ps	
psscale -D3i/7i/4i/0.5ih -Cdisc.cpt -B:discrete: -I0.5 -O -K >> bar.ps	
psscale -D3i/9i/4i/0.5ih -Ccont.cpt -B:continuous: -I0.5 -O >> bar.ps	

练习

1. 用主表 hot 重做 **makecpt** 练习并重做 bar plot.
2. 试设定 **-B10g5**.

2.5.11 照明和亮度 (Illumination and intensities)

GMT 允许虚拟的照明和光线强度。我们可以想像太阳位于无穷远处, 以一定的方位角和俯角照射地面。地表面向太阳的部分被照亮, 而背向太阳的部分变暗。没有任何阴影只是由于地形波动的结果。很显然, 当地表的真实斜率和太阳方向都参加了计算, 如果地表没有地形的起伏, 只有质材的不同, 这里存在着不确定因素。例如, 在我们绘制热流异常的网格时, 朝太阳方向的倾斜意味什么呢? 有多种方法可以解决这个问题, GMT 提供了一种相对比较简单的方法。我们可以计算在太阳照射方向的表面梯度, 并把计算值标准化为 ± 1 的范围, $+1$ 表示最强的太阳照射, -1 意味完全黑暗。虽然我们通常不显示亮度, 它也应当被加入。GMT 把亮度作为一个独立的数据组, 这些值通常源于我们想反映的地球表面的起伏, 它们可以被单独观测为像背散射 (back-scatter) 这样的信息量。

GMT 中的颜色被设定为计算机屏幕的 RGB 系统, 它用红、绿、蓝混合组成其它颜色。RGB 系统是一种高斯坐标系, 可表示为一个彩色立方体, 其原因在参考书的 Appendix I 有较好地的解释。基于 RGB 值, 很难把一种颜色变暗或加

亮，此时可用另一种称为 HSV 的坐标系统代替。如果你沿颜色立方体的黑、白角顶的垂直轴线握住它，其余的 6 个角顶投影在水平面上形成一个 6 边形。6 边形的 6 个角顶分别是原始色红 Red, 黄 Yellow, 绿 Green, 青 Cyan, 兰 Blue, 和洋红 Magenta。CMY 色是补色，在打印时被混合产生新的颜色（现在打印机增加了纯黑色（K）以避免由 CMY 混合成灰色）。在这个坐标系中， $0\text{--}360^\circ$ 角是色调 Hue (H)；饱和度 Saturation 和 色值 Value 比较难解释。这样解释就足够了，如果我们打算变暗任何纯色（在立方体表面上），就保持 H 固定同时增加黑色；如打算变亮，就增加白色；对于在立方体内部的色点，我们将增加或减少灰色。这种操作在 HSV 坐标系中是有效的；因此所有的 GMT 阴影操作都包括了由 RGB 体系向 HSV 体系的转化，产生照明效果，再返回修改后的 RGB 值。

2.5.12 彩色影像 (Color images)

一旦 cpt 文件被建立，由一个网格文件创建一个彩色影像图相对就比较简单了。这里，我们根据 USGS 的全球 30" DEM (数据 id 9) 数据，提取一个 DEM 子集：

grdraster 9 -R-108/-103/35/40 -Gus.grd

使用 **grdinfo** 命令我们发现该文件数据范围由 1000m 到 4300m，所以按照下列命令建立 cpt 文件：

makecpt -Crainbow -T1000/5000/500 -Z > topo.cpt

彩色影像图用 **grdimage** 命令创建，使用常用的公共命令选项 (缺省的 **-R** 来自数据设置) 和一个 **cpt** 文件；其它的主要选项有：

表 4.3: **grdimage** 的主要选项
选项 作用

-E dpi 设置所需要的影像分辨率[缺省为数据分辨率]

-I intenfile 使用 *intensfile* 文件的亮度数据建立虚拟照明

-M 使用 (电视) YIQ 转换改为灰度

绘制一无图案彩色图，并在图上覆盖一彩色图例棒，批处理文件如下：

grdimage us.grd -JM6i -P -B2 -Ctopo.cpt -V -K > topo.ps,

psscale -D3i/8.5i/5i/0.25ih -Ctopo.cpt -I0.4 -B:/m: -O >> topo.ps

该无图案彩色图缺少细节并且没有反映 Rocky Mountain 地区地形的复杂性。这就需要模拟照明效果。我们假设太阳光源在东面，因此我需要用 **grdgradient** 命令确定 $N90^\circ E$ 方向光线地形的亮度梯度。与要求输入输出文件的

名字不同，要用到的选项为：

表 4.4: **grdgradient** 选项

选项	作用
-A azimuth	计算梯度的方位角
-M	指出地理网格
-N[t e][norm[/ offset]]	用 <i>norm/offset</i> [= 1/0 by default] 标准化梯度。 插入 t 使用 \tan^{-1} 变化来标准化 插入 e 使用累积 Laplace 变化来标准化

图 4.1 显示远离正态分布的测深数据的原始分布（左面）。使用反正切转换我们可以保证有更多较均衡的分布（右面）。反正切转换简单地对原始斜率进行评估（箭头为 x 值）并返回相应的反正切值（标准化落在 ± 1 范围内：水平箭头指出 y-值）。

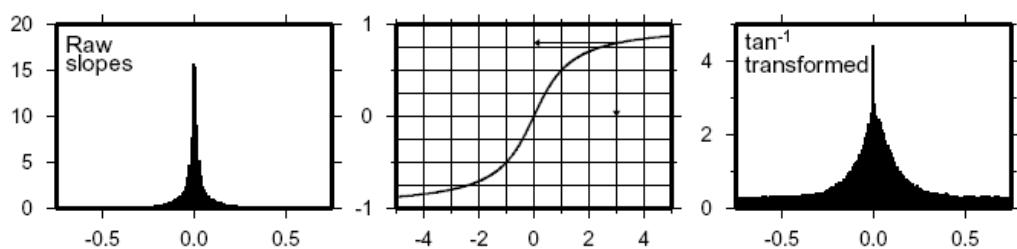


图 4.1: 反正切操作如何进行

-Ne 和 **-Nt** 都能产生较好的梯度。我们更愿意使用 **-Ne** 选项；*norm* 值是主观选择的，你可以在 0.5–5 范围内试验。在我们的例子中，选择

grdgradient us.grd -Ne0.8 -A100 -M -Gus_i.grd

给出 cpt 文件和 2 个网格化数据组，我们就可以生成阴影立体影像图：

grdimage us.grd -Ius_i.grd -JM6i -P -B2 -Ctopo.cpt -K > topo.ps

psscale -D3i/8.5i/5i/0.25ih -Ctopo.cpt -I0.4 -B:/m: -O >> topo.ps

练习

1. 构筑一个灰度影像图
2. 使用 **-N1** 重新运行 **grdgradient**

2.5.13 Perspective views

GMT 通常把 vantage 点限于无穷远处，这样我们可以模拟通过峡谷进行低空飞越。GMT 模块由网格化的数据文件产生透视图的命令是 **grdview**。可产生

两类图：

1. 鱼网或线框图（可以添加也可以没有等值线）
2. 彩色编码表面图（可选阴影、等值线或覆盖）

不考虑投影类型，有一些说明在这里必须要指出：

1. relief_ 文件是一个表面网格化数据组
2. -J 选择地图投影类型
3. -JZheight 用于垂直比例尺
4. -Eazimuth/elevation 用于 vantage 点

此外，还有一些需要的选项：

表 4.5: **grdview** 最常用的选项

选项	作用
-C cptfile	彩色编码表面图和等值线网线图所要求的 cpt 文件
-G drape_file	用 <i>drape_file</i> 代替 <i>relief_file</i> 给颜色赋值
-I intens_file	照明强度文件
-Qm	选择网线图
-Qs[m]	使用多边形的表面图，加 m 显示网线。 该选项为 -W 所允许
-Qi dpi[g]	扫描线转换成图。指定 <i>dpi</i> , 加 g 规定为灰度图, -B 不起作用。
-W pen	在表面图上面画等值线（除 -Qi 外）

对于较小的数据组，网线图效果比较好。我们再次以百慕大 ETOPO5 数据文件的一个小的子集为例生成一个 quick-and-dirty cpt 文件：

grd2cpt bermuda.grd -Cocean > bermuda.cpt

可由下面命令得到一个简单的网线图

grdview bermuda.grd -JM5i -P -JZ2i -E135/30 -B2 -Cbermuda.cpt > map.ps

练习

1. 选择另一个 vantage 点和垂直高度

彩色编码视图

我们来建立一个美国西南 Rockies 的透视，彩色编码视图，命令如下：

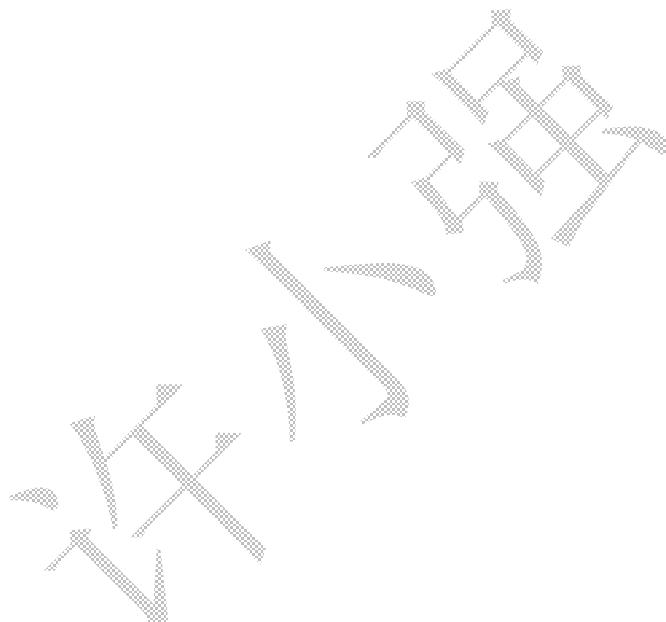
grdview us.grd -JM6i -E135/35 -Qi50 -Ius_i.grd -Ctopo.cpt -V -B2 -JZ0.5i > view.ps

因为我们选择的分辨率为 50 dpi，该视图显得较粗，但以成图快作为补偿，而

且允许我们试用各种不同的分辨率和比例尺。 在决定了最后的值后，我们为最后的输出选用适当的 dpi 。

练习

1. 选择另一种分辨率和比例尺
2. 用另一个照明方向重新运行 **grdgradient** 和绘图
3. 选择较高的 dpi, 比如 200



3. Cookbook

3.1 由网格数据文件绘制等值线图

批处理文件如下

```

gmtset GRID_CROSS_SIZE 0 ANOT_FONT_SIZE 10
psbasemap -R0/6.5/0/9 -Jx1i -B0 -P -K -U"Example 1 in Cookbook" >
example_01.ps
pscoast1 -R-180/180/-90/90 -JH0/6i -X0.25i -Y0.5i -O -K -Bg30 -Dc -G200 >>
example_01.ps
grdcontour1 -R osu91a1f_16.grd -JH -C10 -A50f7 -G4i -L-1000/-1 -Wc0.25pta
-Wa0.75pt2_2:0 -O -K -T0.1i/0.02i >> example_01.ps
grdcontour2 -R osu91a1f_16.grd -JH -C10 -A50f7 -G4i -L-1/1000 -O -K
-T0.1i/0.02i >> example_01.ps
pscoast2 -R0/360/-90/90 -JH180/6i -Y4i -O -K -Bg30; "Low Order Geoid": -Dc
-G200 >> example_01.ps
grdcontour3 osu91a1f_16.grd -JH -C10 -A50f7 -G4i -L-1000/-1 -Wc0.25pta
-Wa0.75pt2_2:0 -O -K
-T0.1i/0.02i:+ >> example_01.ps
grdcontour4 osu91a1f_16.grd -JH -C10 -A50f7 -G4i -L-1/1000 -O -T0.1i/0.02i:+
>> example_01.ps
del .gmt*

```

部分命令注释

psbasemap 画宽 6.5i 高 9i 的图框，非投影直角坐标，单位 1i。无标注及刻度，纸竖置，纸左下角标 GMT、日期时间、字串

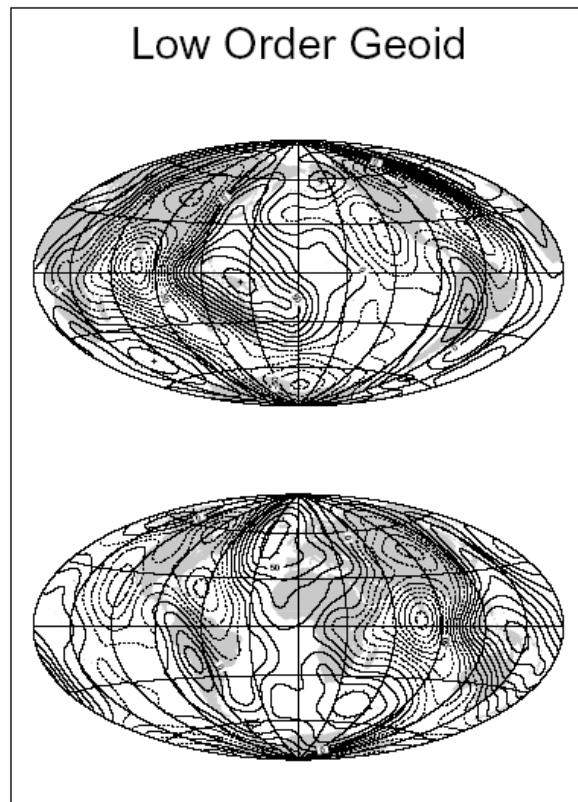
grdcontour¹ 范围同 pscoast，调用网格文件 osu91a1f_16.grd（例中为全球 1 度×1 度的 geoid 网格数据），等值线间隔为 10 个单位，每 50 个单位标注一条，字号 7，标注等值线每两个标注之间间隔 4i，绘制等值线范围 -1000~ -1，非标记线属性：线宽 0.25 点（每点为 1/72i），虚线；标记线属性：线宽 0.75 点，虚线长 2 点，间隔 2 点，起点无偏移，Overlay 写入，封闭的无标记线表示方法为每隔 0.1i 画 1 个 0.02i 的线段，指向低值方向。

grdcontour² 绘制等值线范围 -1~1000，等值线属性采用缺省值（实线）。

pscoast² 中央经度线为 180 度, Y 轴偏移 4i, 图名为“Low Order Geoid”

grdcontour³ 封闭的无标记线视其值的正负, 在中心分别标注“+”和“-”。

del .gmt* 删除当前目录下所有扩展名以 gmt 开头的文件



3.2 由网格数据文件绘制彩色影像图

批处理文件如下

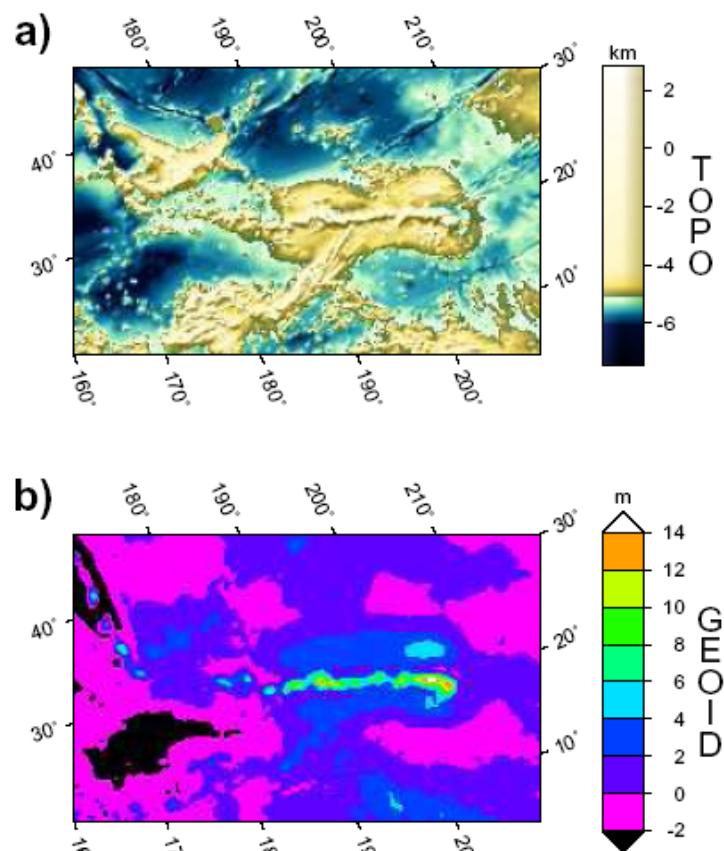
```
gmtset HEADER_FONT_SIZE  
30 OBLIQUE_ANOTATION 0  
DEGREE_FORMAT 0  
  
makecpt -Crainbow -T-2/14/2 >  
g.cpt  
  
grdimage1 geosat5.grd -R160/20/220/30r -JOc190/25.5/292/69/4.5i -E100 -K -P  
-B10 -Cg.cpt -U/-1.25i/-1i/"Example 2 in Cookbook" -X1.5i -Y1.25i >  
example_02.ps  
  
psscale1 -Cg.cpt -D5.1i/1.35i/2.88i/0.4i -O -K -L -B2:GEOID:/:m: -E >>  
example_02.ps  
  
grd2cpt etopo5.grd -Crelief -Z > t.cpt  
  
grdgradient etopo5.grd -A0 -Nt -Getopo5_int.grd  
  
grdimage2 etopo5.grd -Ietopo5_int.grd -R -JO -E100 -B10:"H@#awaiian@#  
T@#opo and @#G@#eoid:" -O -K -Ct.cpt -Y4.5i >> example_02.ps  
  
psscale2 -Ct.cpt -D5.1i/1.35i/2.88i/0.4i -O -K -I0.3 -B2:TOPO:/:km: >>  
example_02.ps  
  
echo -0.4 7.5 30 0.0 1 2 a) > tmp  
echo -0.4 3.0 30 0.0 1 2 b) >> tmp  
pstext tmp -R0/8.5/0/11 -Jx1i -O -N -Y-4.5i >> example_02.ps
```

部分命令注释

makecpt 本例以内置色谱文件库的 rainbow 为标准建立色谱文件，文件名 g cpt，由 -2 到 14，间隔 2

grdimage¹ 用 geosat5.grd 和 g.cpt 绘制等值区图。分区间隔和各区的充填颜色由文件 g.cpt 确定，设置 grd 的分辨率为 100dpi。

Psscale¹ 以图左下角为原点，图例框的左边界 X 坐标（该图宽 4.5i，图例框与图右下角的距离为 $5.1 - 4.5 = 0.6i$ ）；图框中心的 Y 坐标（2 倍 2.7i 为图高），图的长、宽（2.88i、0.4i）。按 g.cpt 文件的分级绘制等大的矩形框，间隔 2 单位，图例的右边竖标“GEOID”，上面标“m”。图例的顶和底分别绘制一个高度为图例宽 1/2 的三角形，分别填充前景色和背景色。



grd2cpt 以色谱文件库的 rainbow 为标准由 etopo5.grd 建立色谱文件 t.cpt

grdgradient 由 etopo5.grd 计算向北面的方向梯度文件。-Nt 用一种累积柯西分布标准化，服从 $gn = (2 * amp / PI) * atan((g - offset) / sigma)$ ，如果没有给出，这里的 sigma 用 $(g - offset)$ 的 L2 标准来评估。

grdimage² 用 etopo5.grd、etopo5_int.grd 和 t.cpt 绘制影象图，设置 grd 的分辨率为 100dpi，写标题。

psscale² 以图左下角为原点，图例的左边界 X 坐标，中心的 Y 坐标，图例的长、宽，按 t.cpt 文件的分级绘制等大的矩形框，图例的右边竖标“TOPO”，上面标“km”。-I0.3：在图例上表示光照效应，表示出光照强度 0.3 以内的效应（最大 1.0）。

```
echo -0.4 7.5 30 0.0 1 2 a) > tmp
echo -0.4 3.0 30 0.0 1 2 b) >> tmp
```

用 echo 写两行 ASCII 码文本，分别为“a)” 和 “b)” 及其参数，存入文件 tmp
 pstext 写 tmp 内容，即在给定位置写 “a)” 和 “b)”。

附：数据文件

GMT 软件包里已经有一批供用户选用的内置.cpt 文件。例如 GMT_rainbow cpt
 (彩虹型) 用 HSV 格式给出标准彩虹每 1 度颜色渐变过程。文件内容如下：

```
# @ (#) GMT_rainbow.cpt 1.2 03/19/99
#
# Standard Rainbow colortable sampled every 1 degree in hue
# COLOR_MODEL = HSV
0 300 1 1 1 299 1 1
1 299 1 1 2 298 1 1
2 298 1 1 3 297 1 1
.....
```



例中设 z 值由 -2 到 14，分别把 -2、14 与 rainbow.cpt 文件的 0、300 的颜色对应，并将 HSV 变换为 RGB 格式，得 g.cpt 内容如下：

```
-2 255 0 255 0 255 0 255      (由 -2 到 0 间隔, RGB=255/0/255)
0 73 0 255 2 73 0 255
.....
```



12 255 0 0 14 255 0 0

B 0 0 0 (背景, 即<-2, RGB=0/0/0)

F 255 255 255 (前景, 即>14, RGB=255/255/255)

N 128 128 128 (空白, RGB=128/128/128)

用 grdxyz 读出的 geosat5.grd 的内容如下，为 xyz 格式的网格文件

```
164.333 47.5833      -1.433
164.417 47.5833      -1.44
164.5 47.5833      -1.451
.....
```

grdimage 命令用 geosat5.grd 和 g.cpt 来绘制彩色充填的等值区图。

t.cpt 内容如下

```
- 7.438 255 0 255      - 6.08677 127 0 255      (z 由 - 7.438 到 -6.08677, RGB 由
255/0/255
                                              演变到 127/0/255)
```

```
- 6.08677 127 0 255      - 5.75589 0 0 255
```

.....

```
- 4.15904 255 127 0 2.804 255 0 0
```

.....

用 grdxyz 读出的 gtopo5.grd 的内容如下，为 xyz 格式的网格文件

```
159.833 47.5833      -5.244
```

```
159.917 47.5833      -5.249
```

```
160  47.5833        -5.25
```

.....

用 grdxyz 读出的 etopo5_int.rgd 的内容如下，为 xyz 格式的网格文件，z 值为由 etopo5.grd 计算的向北面的方向梯度。物理意义为太阳在正北方向时，不同的地形的亮度。

```
159.833 47.5833 0.0519122
```

```
159.917 47.5833 0.0118648
```

```
160  47.5833       -0.000182332
```

.....



3.3 3-D 透视网线图

批处理文件如下

```
echo -10 255 0 255 > zero.cpt
echo 0 100 10 100 >> zero.cpt
grdcontour geoid.grd -Jm0.45i -E60/30 -R195/210/18/25 -C1 -A5 -G4i -K -P
-X1.5i -Y1.5i -U/-1.25i/-1.25i/"Example 4 in Cookbook" > example_04.ps
pscoast -Jm -E60/30 -R -B2/2NEsw -G0 -O -K >> example_04.ps
echo 205 26 0 0 1.1 | psxyz -Jm -E60/30 -R -SV0.2i/0.5i/0.4ii -W1p -O -K -N >>
example_04.ps
echo 205 22.2 36 -90 1 5 N | pstext -Jm -E60/30 -R -O -K -N >> example_04.ps
grdview topo.grd -Jm -Jz0.34i -Czero.cpt -E60/30 -R195/210/18/25/-6/4
-N-6/200/200/200 -Qsm -O -K -B2/2:"Topo ( km )":neswZ -Y2.2i >>
example_04.ps
echo 3.25 5.75 60 0.0 33 2 H@#awaiian@# R@#idge | pstext -R0/10/0/10 -Jx1i -O
>> example_04.ps
del zero.cpt
```

部分命令注释

echo 用 echo 建立一个灰度.cpt 文件 zero.cpt

grdcontour - E60/30: 视点转角/水平倾角

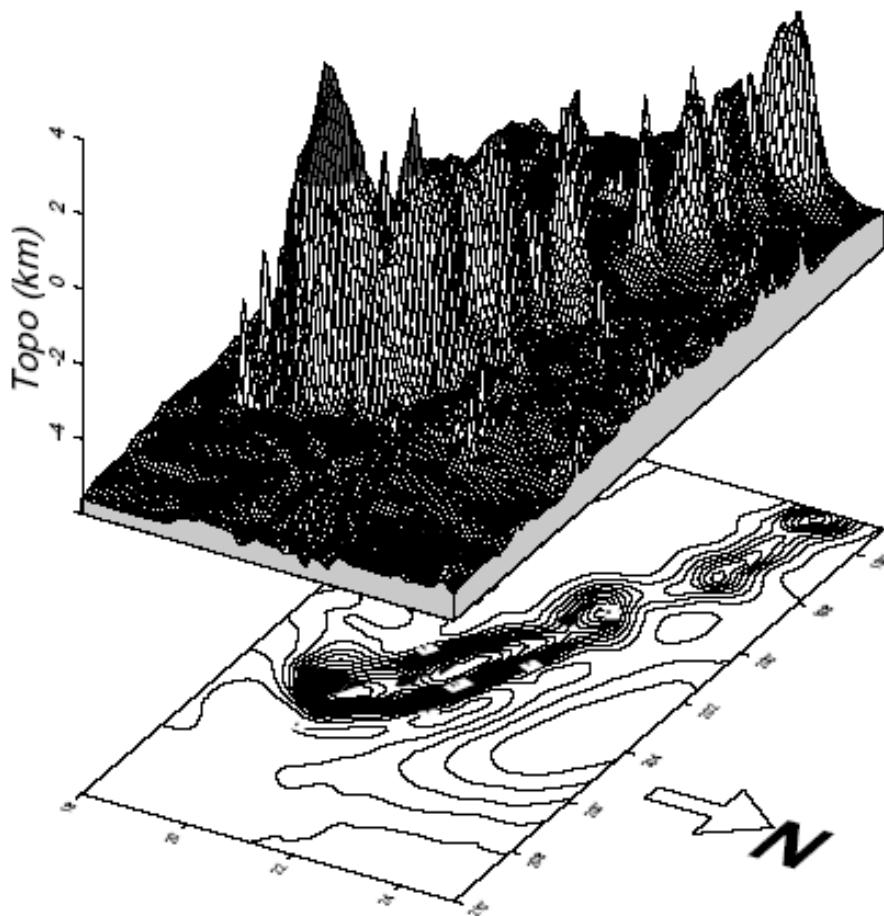
echo 205 26.....

205 26 0 0 1.1 作为 **psxyz** 的输入值, 即由标准输入给出画符号 (SV) 的起点坐标 (205,26,0), 方向 (0), 长度 (1.1i)。-Jm -E-R 都同上, -SV 箭体宽箭/头长箭/头宽, -N 符号可画在图框外; -E60/30: 视点转角/倾角。

echo 205 22.2

205 22.2 36 -90 1 5 N 作为 **pstext** 的标准输入, 依次为 X 坐标, Y 坐标, 字体大小 (p), 角度, 字符集编号, 字符对齐选择 (5 为中中对齐), 字符内容。

HAWAIIAN RIDGE



grdview 设 Z 轴的刻度为每单位 0.34i ; -R195/210/18/25/-6/4 :

```

x1/x2/y1/y2/z1/z2; -N -6/200/200/200: z=-6 为网格图底面, 灰色; -Qsm: 设置
3D 图类型, s 表面图 (类似 Surfer 软件的最小曲率算法), m 网格图, i 影象图;
-B2/2/2:"Topo (km)":neswZ: x,y,z 刻度间隔均为 2, 只标注 Z 轴轴名
echo 3.25 5.75 .....
3.25 5.75 60 0.0 33 2 H@#awaiian@# R@#idge      作为 pstext 的标准输入, @#字
母大小写转换开关
del zero.cpt 删删除色谱文件

```

3.4 3-D 照明表面图

批处理文件如下

```

grdmath -R-15/15/-15/15 -I0.3 X Y HYPOT DUP 2 MUL PI MUL 8 DIV COS
EXCH NEG 10 DIV EXP MUL = sombrero.grd
echo -5 128 5 128 >gray.cpt
grdgradient sombrero.grd -A225 -Gintensity.grd -Nt0.75
grdview sombrero.grd -JX6i -JZ2i -B5/5/0.5SEwnZ -N-1/255/255/255 -Qs
-Iintensity.grd -X1.5i -K -Cgray.cpt -R-15/15/-15/15/-1/1 -E120/30
-U-1.25i/-0.75i/"Example 5 in Cookbook" >example_05.ps
echo 4.1 5.5 50 0 33 2 z(r)=cos(2@~p@~r/8)*e@+r/10@+ | pstext -R0/11/0/8.5
-Jx1i -O >> example_05.ps

```

部分命令注释

grdmath 把方程式 $Z(y)$
 $=\cos(2\pi y/8) \times e^{-y/10}$, 用
RPN 语法写出 [(HYPOT
DUP 2 MUL PI MUL 8 DIV)
COS) (EXCH NEG) 10
DIV) EXP) MUL], 计算.grd

文件 sombrero.grd

echo -5 128.....建立灰度

文件 gray.cpt

grdgradient 由 sombrero.grd 计算 225° 的方向梯度文件 intensity.grd。-Nt 见
Figure 6.2 注释。

grdview 设 Z 轴高度为 6i; xyz 坐标的刻度单位分别为 5、5、0.5; -N -

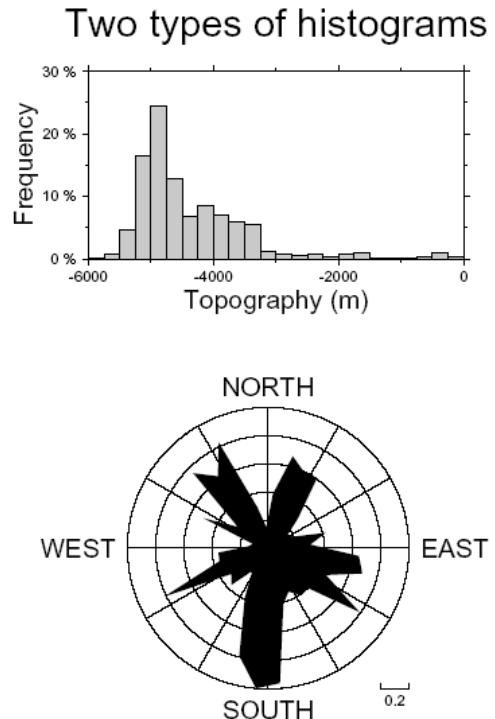
1/255/255/255: z=-1 为网格图底面，白色； -Qs: 设置 3D 图类型为表面图（类似 Surfer 软件的最小曲率算法）；光照强度由文件 intensity.grd 给出； 灰度由灰度文件 gray.cpt 给出； -E120/30: 视点转角/倾角

echo 4.1 5.5 作为 pstext 的标准输入，@#字母大小写转换开关。写图名
 $Z(\gamma) = \cos(2\pi\gamma/8) * e^{-\gamma/10}$

3.5 两类直方图

批处理文件如下

```
psrose fractures.d      -A10r      -S1.8in
-U/-2.25i/-0.75i/"Example 6 in Cookbook"
-P      -G0      -R0/1/0/360    -X2.5i      -K
-B0.2g0.2/30g30 > example_06.ps
pshistogram   -Ba2000f1000:"Topography
(m)":/a10f5:"Frequency":,%::."Two types
of histograms":WSne
v3206.t -R-6000/0/0/30 -JX4.8i/2.4i -G200
-O -Y5.5i -X-0.5i -L0.5p -Z1 -W250 >>
example_06.ps
del .gmt*
```



部分命令注释

psrose - A10r: 玫瑰花图分区为 10 度； - S1.8in: 半径 1.8i, 投影标准化为 0~1； - R0/1/0/360: 极坐标, 半径方向 0~1, 圆周方向 0~360 度

pshistogram 直方图, -B: x 轴标注间隔、刻度、轴名, y 轴标注间隔、刻度、轴名, 图名

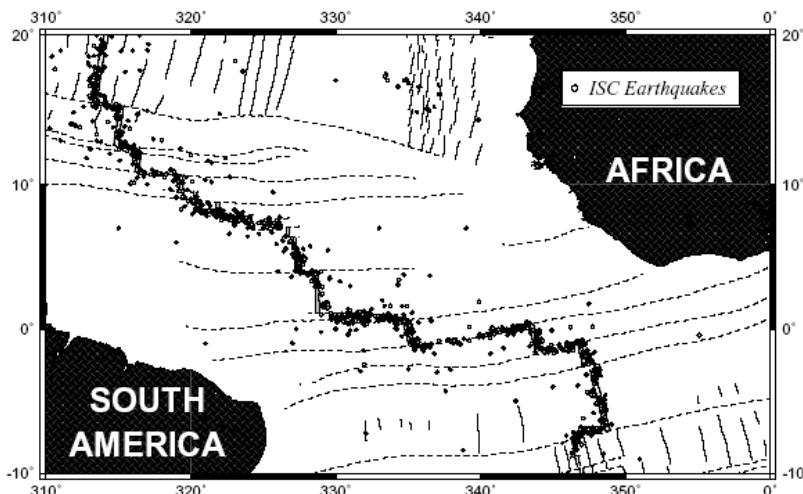
3.6 一个典型的位置图

```
pscoast -R-50/0/-10/20 -JM9i -K -GP300/26 -Dl -W0.25p -B10 -U"Example 7 in
Cookbook" > example_07.ps
```

```

psxy1 -R -JM -O -K -M fz.xy -W0.5pta >> example_07.ps
gawk "{print $1 -360.0, $2, $3*0.01}" quakes.xym | psxy -R -JM -O -K -H1 -Sci
-G255 -W0.25p >> example_07.ps
psxy2 -R -JM -O -K -M isochron.xy -W0.75p >> example_07.ps
psxy3 -R -JM -O -K -M ridge.xy -W1.75p >> example_07.ps
echo -14.5 15.2 > tmp
echo -2 15.2 >> tmp
echo -2 17.8 >> tmp
echo -14.5 17.8 >>
tmp
psxy4 -R -JM -O -K
-G255 -W1p -A tmp
>> example_07.ps
echo -14.35 15.35 >
tmp
echo -2.15 15.35 >>
tmp
echo -2.15 17.65 >>
tmp
echo -14.35 17.65 >> tmp
psxy5 -R -JM -O -K -G255 -W0.5p -A tmp >> example_07.ps
echo -13.5 16.5 | psxy -R -JM -O -K -Sc0.08i -G255 -W0.5p >> example_07.ps
echo -12.5 16.5 18 0 6 5 ISC Earthquakes | pstext -R -JM -O -K >> example_07.ps
echo -43 -5 30 0 1 6 SOUTH > tmp
echo -43 -8 30 0 1 6 AMERICA >> tmp
echo -7 11 30 0 1 6 AFRICA >> tmp
pstext -R -JM -O -S0.75p -G255 tmp >> example_07.ps
del .gmt*

```



部分命令注释

`pscoast - GP300/26`: 陆地填充用模板 26, 反相, 深灰 (300) 底色。

文件 `fz.xy` 内容如下:

```

> 629 OCEAN: 2 FEATURE: 6 NAME: KANE
284.74011 35.04543

```

284.82080 34.99857

.....

> 445 OCEAN: 2 FEATURE: 6 NAME: FIFTEEN TWENTY

298.96030 18.06335

299.04568 18.01421

.....

文件 fz.xy 为区内大洋转换断层的数据，为多段数据格式（-M）。每一段数据由一行“>”开头的文字和一系列 x、y 坐标组成，用来绘制一条断层，文字行包含了断层名字和 x、y 坐标组数等信息。

Psxy¹ 用文件 fz.xy 的数据来绘制多条转换断层

文件 quakes.xym 内容如下

lon lat mag1

338.80 -8.4 4.8

346.68 -8.36 4.4

.....



gawk 命令由文件 quakes.xym 中读取大洋中脊地震的 x、y 坐标和震级，作为 psxy 绘制地震分布图的输入数据。

文件 isochron.xy 和 ridge.xy 分别为区内“等时”断层和大洋中脊断层的数据，数据格式与 fz.xy 相同。

psxy² 用文件 isochron.xy 的数据来绘制“等时”断层

psxy³ 用文件 ridge.xy 的数据来绘制大洋中脊断层

echo -14.5 15.24 行 echo 语句给文件 tmp 赋值

psxy⁴ 用 tmp 数据绘制矩形内框

echo -14.35 15.354 行 echo 语句给文件 tmp 重新赋值

psxy⁵ 用 tmp 数据绘制矩形外框

echo -13.5 16.5 | psxy在矩形框内绘制地震震中图例（圆圈）,echo 给出圆心位置

echo -12.5 16.5 18 0 6 5 ISC Earthquakes | pstext..... 在图例(圆圈)右面写“ISC Earthquakes”，echo 后面参数依次为 x 坐标、y 坐标、字符大小（字号）、角度（0）、字体（6）和对齐方式，5 表示坐标值为字串的中心。

echo -43 -5.....3 行 echo 语句给文件 tmp 重新赋值，pstext 用 tmp 数据分别写“SOUTH”，“AMERICA”，“AFRICA” 3 个字串。对齐方式 6 表示坐标值为字串右边中点。

3.7 3-D 柱状图

```

grd2xyz topo.grd > tmp.d
psxyz tmp.d      -B1/1/1000:"Topography ( m ) "::ETOPO5:WSneZ+
-R-0.1/5.1/-0.1/5.1/-5000/0 -JM5i -JZ6i -E200/30 -So0.083333ub-5000 -P
-U"Example 8 in Cookbook" -W0.25p -G240 -K > example_08.ps
echo 0.1 4.9 24 0 1 9 This is the surface of cube | pstext -R -JM -JZ -Z0 -E200/30
-O >> example_08.ps

```

部分命令注释

grd2xyz 把 2 进位的 .grd 文件
topo.grd 转变为 ASCII 文件 (x、y、
z) tmp.d

tmp.d 文件内容如下：

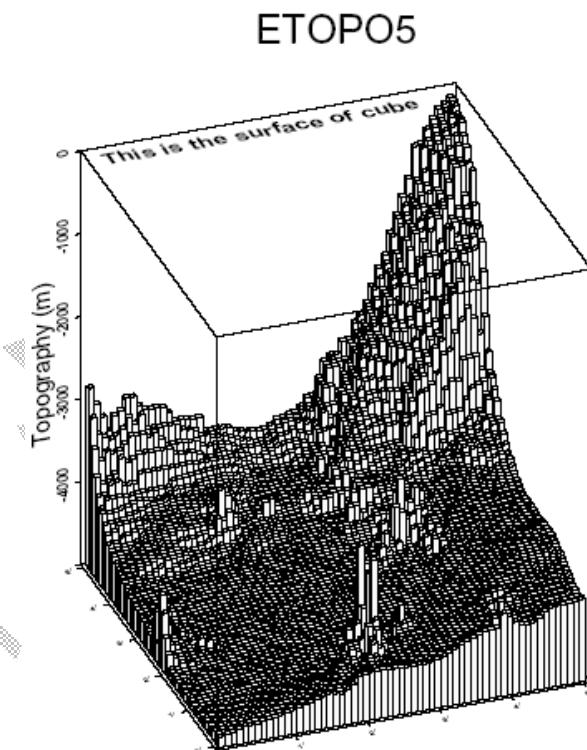
0 5	-2823
0.0833333 5	-3027
0.1666667 5	-3195
0.25 5	-3384

psxyz 读数据文件 tmp.d, 设定 x、
y、z 轴刻度标注的间隔, z 轴标注
轴名 “Topography (m) ”, 图名
“ETOPO5”, 标注 WSZ;

设定 x、y、z 的范围, 墨卡托
投影边长 5i, Z 轴长 6i,
-S0.083333ub-5000: 画由基点

到 z 的 3D 柱, 柱宽 0.083 个 x 轴单位, 基点设为 -50000; -E200/30 视点转
角/倾角。

echo 0.1 4.9 在立方体上表面标注文本



3.8 沿测线变化的时间 - 序列图

Time-series as ``wiggles" along a track

```
pswiggle all.xys -R185/250/-68/-42 -U"Example 9 in Cookbook" -K -Jm0.13i
```

```

-Ba10f5 -G0 -Z2000 -W0.25p -M -S240/-67/500/@~m@~rad > example_09.ps
psxy1 -R -Jm -O -K ridge.xy -W1.75p >> example_09.ps
psxy2 -R -Jm -O -K -M fz.xy -W0.5pta >> example_09.ps
REM Plot labels
gawk "{if (NF = 5 ) print $3, $4, 10, 50, 1, 7, $2}" all.xys | pstext      -R -Jm
-D-0.05i/-0.05i -O >> example_09.ps
del .gmt*

```

部分命令注释

pswiggle 用数据文件 (all.zys) 绘制测线。

文件 all.xys 为区内诸条测线的数据，为多段数据格式 (-M)。

> 107 222.267 -66.2309

-3

241.434 -56.3094

0

241.37 -56.3591

-50

241.306 -56.4088

-104

.....

222.267 -66.2309

-3

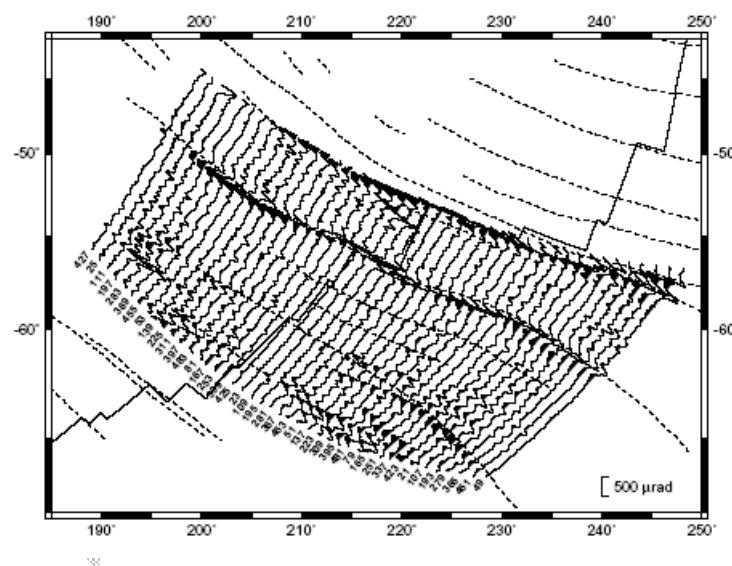
> 109 204.893 -63.2221 -28

220.648 -52.6068 0

220.594 -52.6583 -25

220.54 -52.7099 360

.....



每一段数据由一行 “>” 开头的文字和一系列 x、y、z 值组成。文字行包含了测线编号和该测线最后一点的 x、y、z 值。段内每一测点的 x、y、z 值中，x、y 为该点经纬度，z 值为该点测值。每一段数据绘制一条测线。-Z2000 为异常下限，z 值超过 2000 的部分着黑色 (-G0)。

文件 ridge.xy 和 fz.xy 分别为区内大洋中脊断层和转换断层的数据。fz.xy 为多段数据格式 (-M)。

psxy¹ 用文件 ridge.xy 的数据来绘制大洋中脊断层

psxy² 用文件 isochron.xy 的数据来绘制多条转换断层

gawk 语言中“NF”为当前输入纪录的字段数。多段数据文件 all.xys 每一段数据开头为一行“>”开头的文字，该行由 5 个字段组成，该行以后的数据行，每行由 3 个字段组成。条件语句 if (NF = 5) 逐行检查每行的字段数，每段的首行字段数条件为真时，取第 3、4 号字段数内容为待写字符的起点经纬度，字段数即测线编号为待写字符。

3.9 地理坐标棒图

```
pscoast -R-180/180/-90/90 -JX8i/5id -Dc -G0 -E200/40 -K -U"Example 10 in
Cookbook" > example_10.ps
psxyz agu.d -R-180/180/-90/90/1/100000 -JX -JZ2.5il -So0.3ib1 -G140 -W0.5p
-O -K -E200/40 -B60g60/30g30/a1p:Memberships:WSneZ >> example_10.ps
gawk "{print $1-10, $2, 20, 0, 0, 7, $3}" agu.d | pstext -R-180/180/-90/90 -JX -O
-K -E200/40 -G255 -S0.5p >> example_10.ps
echo 4.5 6 30 0 5 2 AGU 1991 Membership Distribution | pstext -R0/11/0/8.5 -Jxli
-O >> example_10.ps
del .gmt*
```

部分命令注释

pscoast - E200/40

视点转角/倾角。

psxyz agu.d

agu.d 文件内容如

下：

-100 42 21666

-60 -20 235

20 10 132

90 30 1331

10 50 2936

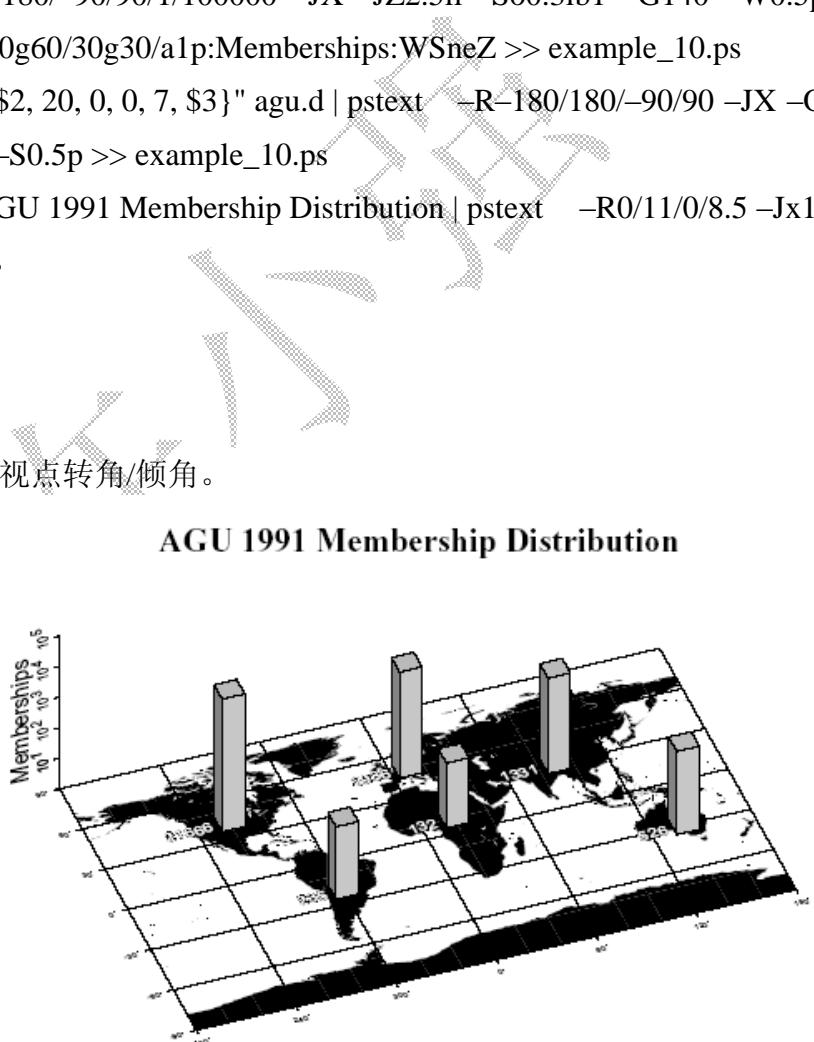
140 -30 526

读 数据文件

agu.d, 设定 x、y、

z 的范围, z 轴长 2.5i, 对数坐标, - So0.3ib1: 画由基点到 z 的 3D 柱, 宽 0.3i, 基点设为 1; WsneZ z 轴用指数方式标注, 加轴名, 标注 WSZ。

gawk 读 agu.d 中的数据, 依次为 x 坐标 (\$1)、y 坐标 (\$2)、z 值 (\$3), 加上



另外 4 个常数，作为 `pstext` 要求标注每个 3D 柱 z 值所需要的 7 列数据。依次为 x 坐标 (-10), y 坐标, 字体大小 (点), 角度, 字体集编号, 字串对齐方式 (左下) 和待标注内容。文本轮廓线宽 0.5 点

`echo 4.5 6 写标题`

3.10 数据最佳三角形线性内插 (Optimal triangulation of data)

```
triangulate table_5.11 -M > net.xy
psxy1 -R0/6.5/-0.2/6.5 -JX3.06i/3.15i -B2f1WSNe -M net.xy -W0.5p -P -K
-Y4.65i > example_12.ps
```

```
psxy2 table_5.11 -R -JX -O -K -Sc0.12i -G255 -W0.25p >> example_12.ps
gawk1 "{print $1, $2, 6, 0, 0, 6, NR -1}" table_5.11 | pstext -R -JX -N -O -K >>
example_12.ps
```

REM Then draw network and print the node values

```
psxy3 -R -JX -B2f1eSNw -M net.xy -W0.5p -O -K -X3.25i >> example_12.ps
```

```
psxy4 -R -JX -O -K table_5.11
```

```
-Sc0.03i -G0 >> example_12.ps
```

```
gawk2 "{print $1, $2, 6, 0, 0, 5, $3}"
```

```
table_5.11 | pstext -R -JX -O -K
-W255o -C0.01i/0.01i -D0.08i/0i -N
>> example_12.ps
```

REM Then contour the data and draw triangles using dashed pen; use

"minmax" and "makecpt" to make a color palette (.cpt) file

```
makecpt -Cjet -T675/975/25 >
topo.cpt
```

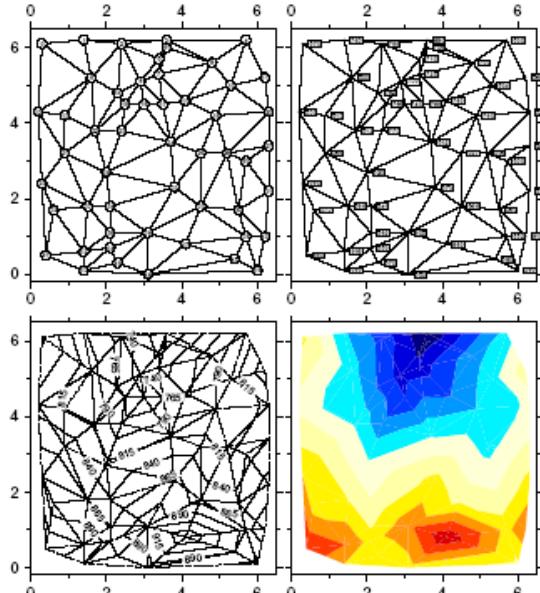
```
pscontour1 -R -JX table_5.11
-B2f1WSne -W0.75p -Ctopo.cpt
```

```
-L0.25pta -G1i -X-3.25i -Y-3.65i -O -K -U"Example 12 in Cookbook" >>
example_12.ps
```

REM Finally color the topography

```
Pscontour2 -R -JX table_5.11 -B2f1eSNw -Ctopo.cpt -I -X3.25i -O -K >>
example_12.ps
```

Delaunay Triangulation



```

echo 3.16 8 30 0 1 2 Delaunay Triangulation | pstext -R0/8/0/11 -Jxli -O -X-3.25i
>> example_12.ps
REM
del .gmt*

```

部分命令注释

table_5.11 文件格式

0.3 6.1 870.0

1.4 6.2 793.0

2.4 6.1 755.0

net.xy 文件格式

> Edge 0-1

0.3 6.1

1.4 6.2

> Edge 0-5

0.3 6.1

1.6 5.2

triangulate table_5.11 -M > net.xy

读 xy (z) 文件 table_5.11，对 xy 平面上的离散点执行最优 Delaunay 三角形化，即构筑尽可能等边的三角形网。结果用多段文件形式写入文件 net.xy。每一段表示三角形 1 条边两个端点的 xy 坐标。

psxy¹ 绘左上图图框、标注；读取多段文件 net.xy，绘各条边组成三角形网

psxy² 读取文件 table_5.11，在三角形每个角顶画一个小圆圈，用白色填充。

gawk¹ 在小圆中心写点号（由 0 开始的自然数）

psxy³ 以左上图为相对坐标，绘右上图图框、标注；绘各条边组成三角形网

psxy⁴ 在三角形顶点（即原始数据点）绘制黑色小圆点

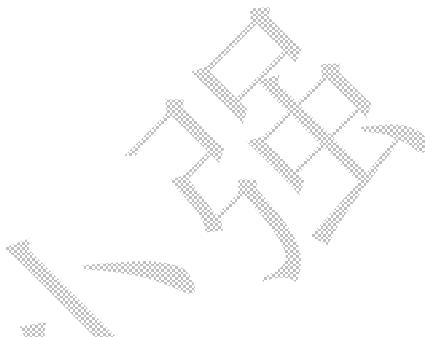
gawk² 在小圆点右面 0.08i 标注文件 table_5.11 的 z 值，z 值数字外 0.01i 画黑色矩形框。

makecpt 调用 GMT 库中的.cpt 文件 jet.cpt，建立等值线色谱文件 topo.cpt

pscontour¹ 以右上图为相对坐标，绘左下图图框、标注；绘各条边组成三角形网，虚线；根据 topo.cpt 的间隔画等值线，标注间隔 1i。

Pscontour² 以左下图为相对坐标，绘右下图图框、标注；根据 topo.cpt 的间隔画等值线和填充颜色

echo 3.16 8写图名。



3.11 数据网格化和趋势面

```

gmtset GRID_PEN 0.25pt
psxy1 table_5.11 -R0/7/0/7 -JX3.06i/3.15i -B2f1WSNe -Sc0.05i -G0 -P -K -Y6.45i
> example_14.ps
gawk1 "{print $1+0.08, $2, 6, 0, 0, 5, $3}" table_5.11 | pstext -R -JX -O -K -N >>
example_14.ps
blockmean table_5.11 -R0/7/0/7 -I1 > mean.xyz
REM Then draw blocmean cells
psbasemap -R0.5/7.5/0.5/7.5 -JX -O -K -B0g1 -X3.25i >> example_14.ps
psxy2 -R0/7/0/7 -JX -B2f1eSNw mean.xyz -Ss0.05i -G0 -O -K >> example_14.ps
gawk2 "{print $1+0.1, $2, 6, 0, 0, 5, $3}" mean.xyz | pstext -R -JX -O -K -W255o
-C0.01i/0.01i -N >> example_14.ps
REM Then surface and contour the data
surface mean.xyz -R -I1 -Gdata.grd
grdcontour1 data.grd -JX -B2f1WSne -C25 -A50 -G3i/10 -S4 -O -K -X-3.25i
-Y-3.55i >> example_14.ps
psxy3 -R -JX mean.xyz -Ss0.05i -G0 -O -K >> example_14.ps
REM Fit bicubic trend to data and compare to gridded surface
grdtrend data.grd -N10 -Ttrend.grd
grdcontour2 trend.grd -JX -B2f1wSne -C25 -A50 -G3i/10 -S4 -O -K -X3.25i >>
example_14.ps
project -C0/0 -E7/7 -G0.1 -Fxy > track
psxy4 -R -JX track -W1pto -O -K >> example_14.ps
REM Sample along diagonal
Grdtrack1 track -Gdata.grd | gawk "{print $3, $4}" > data.d
Grdtrack2 track -Gtrend.grd | gawk "{print $3, $4}" > trend.d
REM minmax data.d trend.d -I0.5/25
REM Use result of minmax manually in -R below:
Psxy5 -R0/10/775/1050 -JX6.3i/1.4i data.d -W1p -O -K -X-3.25i -Y-1.9i
-B1/50WSNe >> example_14.ps
Psxy6 -R -JX trend.d -W0.5pta -O -U"Example 14 in Cookbook" >> example_14.ps
del .gmt*

```

部分命令注释

psxy¹ 绘左上图图框、标注；读数据文件 table_5.11；在数据点绘制小圆，黑色填充。

gawk¹ 标注数据点的 z 值，标注起点的 x 轴位置向右偏移 0.08 单位

blockmean 数据预处理，读离散的 ASCII 数据文件 table_5.11，以 1×1 个单位为网格（块），计算每块的平均值及点坐标，并写到输出文件 mean.xyz 中，没有原始数据点的块空白

psbasemap 绘制并标注右上图框，图内绘制 1×1 的网格

psxy² 读文件 mean.xyz，绘数据点，符号为黑色小正方形

gawk² 标注数据点的 z 值，在小正方形右面 $0.1i$ 标注文件 table_5.11 的 z 值，z 值数字外 $0.01i$ 画黑色矩形框。

surface mean.xyz -R -I1 -Gdata.grd

由处理后的离散数据 mean.xyz 产生 2 进位的 data.grd 文件，grd 范围同上（-R0/7/0/7），间隔为 1×1 个单位。

grdcontour¹ 绘制并标注左中图框，由 data.grd 绘制等值线。-G3i/10-S4：同一等值线上标注间隔和宽度（点），大致每 4 个等值线间隔重复采样。

psxy³ 读文件 mean.xyz，画数据点，符号为小正方形。

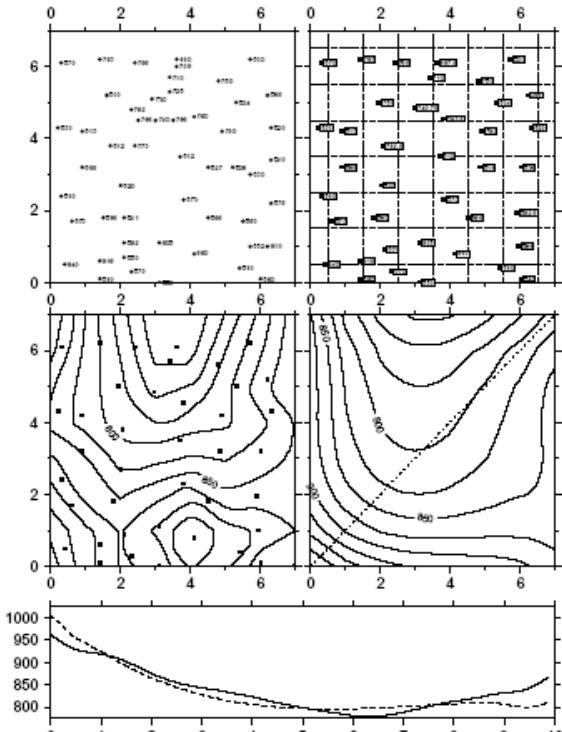
grdtrend data.grd -N10 -Ttrend.grd

读 data.grd 文件，采用最小二乘法，用低次多项式趋势拟合 data.grd 数据，趋势面由双三次方程决定。设置拟合模型参数数目为 10。输出拟合后的 trend.grd 文件。

Grdcontour² 绘制并标注右中图框，由 trend.grd 绘制等值线。

project 创建对角线的剖面线文件，起点坐标 0,0，终点坐标 7,7，每 0.1 个单位采点一个，把每个点的 x,y 写入数据文件 track。

Psxy⁴ 绘制剖面线，虚线



Grdtrack¹ track - Gdata.grd | gawk "{print \$3, \$4}" > data.d

沿剖面线方向(文件 track)从 data.grd 中采样, 结果作为 gawk "{print \$3, \$4}" 的输入数据, 再写到数据文件 data.d 中

grdtrack² track - Gtrend.grd | gawk "{print \$3, \$4}" > trend.d

沿剖面线方向(文件 track)从 trend.grd 中采样, 结果作为 gawk "{print \$3, \$4}" 的输入数据, 再写到数据文件 trend.d 中

psxy⁵ 绘制并标注下面图框, 由 data.d 的 x,y 绘制实线段。

Psxy⁶ 由 trend.d 的 x,y 绘制虚线段。

3.12 更多的数据网格化方法

```
gmtset MEASURE_UNIT INCH ANOT_FONT_SIZE 9
pscontour -R0/6.5/-0.2/6.5 -Jx0.45i -P -K -Y5.5i -Ba2f1WSne table_5.11
-Cex16.cpt -I > example_16.ps
echo 3.25 7 18 0 4 CB pscontour      (triangulate) | pstext -R -Jx -O -K -N >>
example_16.ps
REM
surface table_5.11 -R -I0.1 -Graws0.grd
grdview1 raws0.grd -R -Jx -Ba2f1WSne -Cex16.cpt -Qs -O -K -X3.5i >>
example_16.ps
echo 3.25 7 18 0 4 CB surface      (tension = 0 ) | pstext -R -Jx -O -K -N >>
example_16.ps
REM
surface table_5.11 -R -I0.1 -Graws5.grd -T0.5
grdview2 raws5.grd -R -Jx -Ba2f1WSne -Cex16.cpt -Qs -O -K -Y-3.75i -X-3.5i
>> example_16.ps
echo 3.25 7 18 0 4 CB surface      (tension = 0.5 ) | pstext -R -Jx -O -K -N >>
example_16.ps
REM
triangulate table_5.11 -Grawt.grd -R -I0.1 > NUL
grdfilter rawt.grd -Gfiltered.grd -D0 -Fc1
grdview filtered.grd -R -Jx -Ba2f1WSne -Cex16.cpt -Qs -O -K -X3.5i >>
example_16.ps
echo 3.25 7 18 0 4 CB triangulate @~\256@~ grdfilter | pstext -R -Jx -O -K -N >>
example_16.ps
```

```

echo 3.2125 7.5 32 0 4 CB Gridding of Data | pstext      -R0/10/0/10 -Jxli -O -K -N
-X-3.5i >> example_16.ps
psscale -D3.21/0.35/5/0.25h -Cex16.cpt -O -U"Example 16 in Cookbook" -Y-0.75i
>> example_16.ps
del .gmt*

```

部分命令注释

pscontour 绘制并标注左上图框，由 x,y,z 数据文件 table_5.11 绘制等值线。等值线间隔及充填颜色调用色谱文件 ex16 cpt，给出前景、背景及 NaN 的填充方式

echo 3.25 7 写字符串
pscontour (triangulate) , 底中对齐

surface table_5.11 - R - I0.1 -
Graws0.grd

由离散数据文件 table_5.11 产生 2 进位的 raws0.grd 文件，间隔为 0.1×0.1 个单位，张力系数 0

grdview¹ 绘制并标注右上图框，由 grd 文件 raws0.grd 及色谱文件 ex16.cpt 绘制等值线并填充。选择 surface 方式

echo 3.25 7 写字符串 surface (tension = 0) , 底中对齐

surface 由离散数据文件 table_5.11 产生 2 进位的 raws0.grd 文件，张力系数 0.5

grdview² 绘制并标注左下图框，由 grd 文件 raws5.grd 及色谱文件 ex16.cpt 绘制等值线并填充。

echo 3.25 7 写字符串 surface (tension = 0.5) , 底中对齐

triangulate table_5.11 - Grawt.grd - R - I0.1 > NUL

执行 Delauney 三角型法构筑 grd 文件（尽可能等边），

grdfilter rawt.grd - Gfiltered.grd - D0 - Fc1

读入 grd 文件 rawt.grd，按时间域，用多种方法进行筛选 - D0: x,y 网格与宽度单位一致，采用笛卡儿距离； - Fc1：采用余弦 Arch 进行筛选；输出文件 filtered.grd

grdview 绘制并标注右下图框，由网格文件 filtered.grd 及色谱文件 ex16.cpt 绘制

等值线并填充。

echo 3.25 7 写字符串 triangulate→grdfilter, 底中对齐
 echo 3.2125 7.5 写图名字符串 Gridding of Data, 底中对齐 (疑为 CT, 顶中对齐)
 psscale 画图例, 绘制系统时间标志。

3.13 体积计算和特定地区的圈定

gmtset ELLIPSOID Sphere

REM Define location of Pratt seamount

echo -142.65 56.25 > pratt.d

makecpt -Crainbow -T-60/60/10 -Z > grav.cpt

grdgradient ss_grav.grd -Nt1 -A45 -Gss_grav_i.grd

grdimage ss_grav.grd -Iss_grav_i.grd -JM5.5i -Cgrav.cpt -B2f1 -P -K -E100 -X1.5i -Y5.85i > example_18.ps

pscoast¹ -R-149/-135/52.5/58 -JM -O -K -Di -G160 -W0.25p >> example_18.ps

psscale -D2.75i/-0.4i/4i/0.15ih -Cgrav.cpt -B20f10/:mGal: -O -K >> example_18.ps

echo {print \$1, \$2, 12, 0, 1, "LB", "Pratt"} > t

gawk¹ -f t pratt.d | pstext -R -JM -O -K -D0.1i/0.1i >> example_18.ps

gawk² "{print \$1, \$2, 0, 200, 200}"

pratt.d | psxy -R -JM -O -K -SE

-W0.25p >> example_18.ps

grdcontour¹ ss_grav.grd -JM -C20

-B2f1WSEn -O -K -Y-4.85i

-U/-1.25i/-0.75i/"Example 18 in

Cookbook" >> example_18.ps

grdcontour² ss_grav.grd -JM -C10

-L49/51 -O -K -Dsm

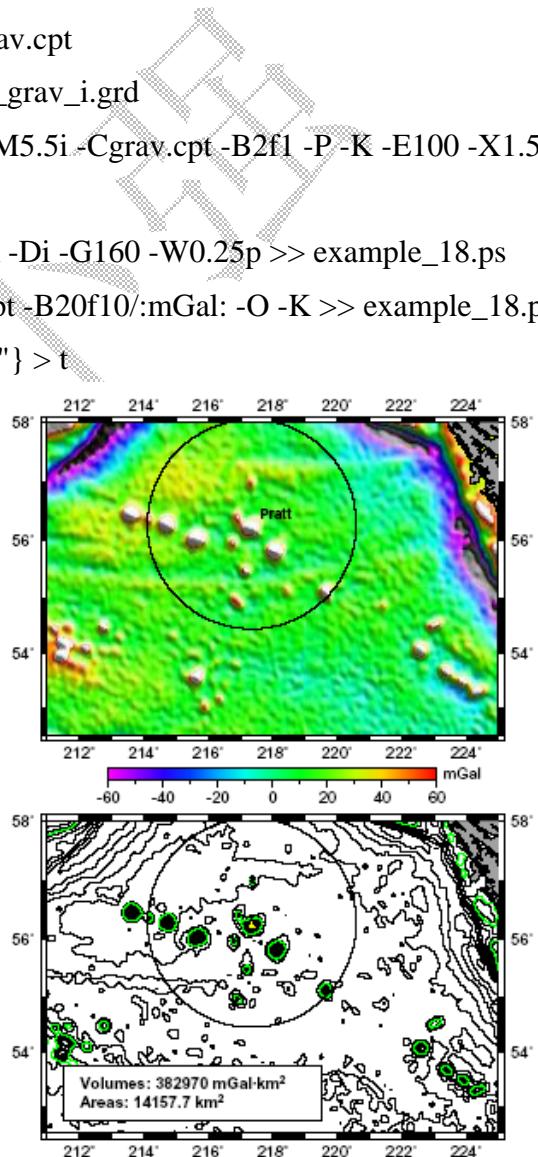
-Wc0.75p/0/255/0 >> example_18.ps

pscoast² -R -JM -O -K -Di -G160

-W0.25p >> example_18.ps

gawk³ "{print \$1, \$2, 0, 200, 200}"

pratt.d | psxy -R -JM -O -K -SE -W0.25p >> example_18.ps



REM Only consider closed contours

REM Make simple gawk script to calculate average position of locations

```
echo BEGIN { > center.awk
echo x = 0 >> center.awk
echo y = 0 >> center.awk
echo n = 0 >> center.awk
echo } >> center.awk
echo { >> center.awk
echo x += $1 >> center.awk
echo y += $2 >> center.awk
echo n++ >> center.awk
echo } >> center.awk
echo END { >> center.awk
echo print x/n, y/n >> center.awk
echo } >> center.awk
```

REM Now determine centers of each enclosed seamount exceeding 50 mGal but
REM only plot the ones within 200 km of Pratt seamount.

REM Now determine mean location of each closed contour and
REM add it to the file centers.d using center.awk script

```
for %%f in (sm_*.xyz) do gawk -f center.awk %%f >> centers.d
```

REM Only plot the ones within 200 km

```
gmtselect -R -JM -C200/pratt.d centers.d > tmp2
```

```
psxy1 tmp2 -R -JM -O -K -SC0.07i -G255/0/0 -W0.25p >> example_18.ps
```

```
psxy2 -R -JM -O -K -ST0.1i -G255/255/0 -W0.25p pratt.d >> example_18.ps
```

REM Then report the volume and area of these seamounts only

REM by masking out data outside the 200 km-radius circle

REM and then evaluate area/volume for the 50 mGal contour

```
Grdmath1 -R -I2m -F -142.65 56.25 GDIST = mask.grd
```

```
grdclip mask.grd -Sa200/NaN -Sb200/1 -Gmask.grd
```

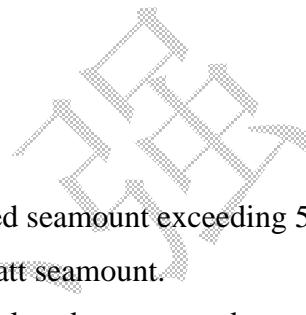
```
grdmath2 ss_grav.grd mask.grd MUL = tmp.grd
```

```
echo -148.5 52.75 > tmp1
```

```
echo -140.5 52.75 >> tmp1
```

```
echo -140.5 53.75 >> tmp1
```

```
echo -148.5 53.75 >> tmp1
```



```
psxy3 -R -JM -A -O -K -L -W0.75p -G255 tmp1 >> example_18.ps
echo {printf "-148 53.08 14 0 1 LM Areas: %%s km@+2@+\n-148 53.42 14 0 1 LM
Volumes: %%s km@+3@+\n", $2, $3} > t1
grdvolume tmp.grd -C50 -Sk | gawk -f t1 | pstext -R -JM -O >> example_18.ps
del .gmt*
```

部分命令注释

gmtset ELLIPSOID Sphere 改缺省的 ELLIPSOID WGS-84 为 Sphere
 echo -142.65 56.25 > pratt.d 把 Pratt 海山位置（经纬度）写入文件 pratt.d
 makecpt -Crainbow -T-60/60/10 -Z > grav.cpt 以色谱文件库的 rainbow 为标准建立
 连续色谱文件 grav.cpt
 grdgradient ss_grav.grd -Nt1 -A45 -Gss_grav_i.grd

由 ss_grav.grd 计算 45° 方向的方向梯度文件 ss_grav_i.grd。-Nt 见 Figure 6.2
 注释。

ss_grav.xyz 文件如下

```
-148.983 57.9833 99.8354
-148.95 57.9833 98.0651
.......
```



ss_grav_i.xyz 文件如下

```
-148.983 57.9833 -0.449835
-148.95 57.9833 -0.520409
.......
```

注意两个网格文件 x 的增量均为 30 分之 1 度，即 2 分。

grdimage 用 ss_grav.grd ss_grav_i.grd 和 grav.cpt 绘制影象图(上图)，设置 grd 的分辨率 100dpi。

pscoast¹ 绘制、标注上图图框。

psscale 绘制、标注图例条。以图左下角为原点，水平图例框 (h) 中心点 X 坐标（该图宽 5.5i，图例中心 X 坐标 5.5/2=2.75i）；图框中心的 Y 坐标（图框下边界下方 0.4i），图例的长、宽（4i、0.15i）。图例的标记刻度非标记刻度（20、10），图例的右边标“mGal”，。

echo {print \$1,把 “Pratt” 写入文件 t

gawk¹ 根据文件 t 和数据文件 pratt.d 写 “Prett” 海山名字。

gawk² 画椭圆命令，6 个参数，中心经、纬度、角度、长轴、短轴长度 (km)。

例中，椭圆中心坐标由 pratt.d 里读出，由于长轴、短轴长度一样，实际是以

Prett 海山为圆心，画半径 200km 的圆。

grdcontour¹ 按照 ss_grav.grd 绘制全部等值线图，间距 20。

grdcontour² 按照 ss_grav.grd 绘制等值线图，间距 10，限定只画 49-51 范围内的等值线（因此只有 50 一条），分别将每一条 50 等值线的 x、y、z 值建立一个 xyz 文件存盘，用 sm_50_n_i 来命名，其中 n 为序号，i 表示为封闭的等值线。等值线为绿色 (0/255/0)。

pscoast² 绘制、标注下图图框。

gawk³ 以 Prett 海山位置为圆心，画半径 200km 的大圆。

echo x = 0 用 13 行 echo 写一段 gawk 程序，存入 center.awk。程序为
BEGIN {

```
x = 0
y = 0
n = 0
}      (变量赋初值)
```

```
{
x += $1      (x 坐标累加)
y += $2      (y 坐标累加)
n++         (n 记数器累加)
}
```

```
END {
print x/n, y/n    (求算并输出等值线中心经纬度坐标)
}
```

~~for %%f in (sm_*.xyz) do gawk -f center.awk %%f >> centers.d~~

顺序读取文件 sm_*.xyz，执行 gawk，即把所有封闭的超过 50 mGal 等值线中心坐标（由执行 center.awk 得）建立一个数据文件 centers.d。本例中有 33 个等值线中心坐标满足条件。

gmtselect -R -JM -C200/pratt.d centers.d > tmp

筛选，从 centers.d 中选择距 Pratt，即大圆圆心的距离在 200 公里以内的封闭 50 等值线。存入临时文件 tmp。本例中有 11 个等值线中心坐标满足条件。

psxy¹ 以 tem 中的坐标为圆心绘制直径 0.04i 的红色 (255/0/0) 圆

psxy² 以 Pratt 海山为中心，画边长 0.1i 的黄色 (255/255/0) 三角形

grdmath -R -I2m -F -142.65 56.25 GDIST = mask.grd

以 2 分为网格间隔，计算由 Pratt 海山到每个节点的距离。生成一个新的网格文件 mask.grd

```
grdclip mask.grd -Sa200/NaN -Sb200/1 -Gmask.grd
```

对 mask.grd 进行处理，将 >200 的节点改为空白，将 <200 的节点改值为 1。处理后的网格文件另存为 mask.grd

grdmath ss_grav.grd mask.grd MUL = tmp.grd 两个网格文件相乘，生成临时网格文件 tmp.grd

```
echo -148.5 52.75 .....
```

psxy 用上面 4 行 echo 给出的坐标画矩形框

echo {printf 用 echo 写两个字串 Areas: km² 和 Volumes: km³，存入 ft，分别作为面积和体积的标注。

```
grdvolume tmp.grd -C50 -Sk | gawk -f t | pstext -R -JM -O >> example_18.ps
```

计算给定条件，本例为距 Pratt 海山 200km 以内，等值线 z=50 以上的（山体）体积和表面积。单位度转换为 km。计算结果通过管道命令作为 gawk -f t 的输入，即分别给体积和表面积赋值；最后通过 pstext 命令输出到.ps 文件。

3.14 彩色图案模版的使用

```
gmtset COLOR_MODEL rgb
```

```
grdmath1 -R-180/180/-90/90 -I1 -F Y COSD 2 POW = lat.grd
```

```
grdmath2 -R-180/180/-90/90 -I1 -F X = lon.grd
```

```
echo 0 255 255 255 1 0 0 255 > lat.cpt
```

```
makecpt -Crainbow -T-180/180/60 -Z > lon.cpt
```

```
grdimage1 lat.grd -JI0/6.5i -Clat.cpt -P -K -Y7.5i -B0 > example_19.ps
```

```
pscoast1 -R -JI -O -K -Dc -A5000 -Gc >> example_19.ps
```

```
grdimage2 lon.grd -JI -Clon.cpt -O -K >> example_19.ps
```

```
pscoast2 -R -JI -O -K -Q >> example_19.ps
```

```
pscoast3 -R -JI -O -K -Dc -A5000 -W0.25p >> example_19.ps
```

```
echo 0 20 32 0 1 CM FIRST INTERNATIONAL | pstext -R -JI -O -K -G255/0/0
```

```
-S0.5p >> example_19.ps
```

```
echo 0 -10 32 0 1 CM GMT CONFERENCE | pstext -R -JI -O -K -G255/0/0 -S0.5p
```

```
>> example_19.ps
```

```
echo 0 -30 18 0 1 CM Honolulu, Hawaii, April 1, 2000 | pstext -R -JI -O -K
```

```
-G0/255/50 -S0.25p >> example_19.ps
```

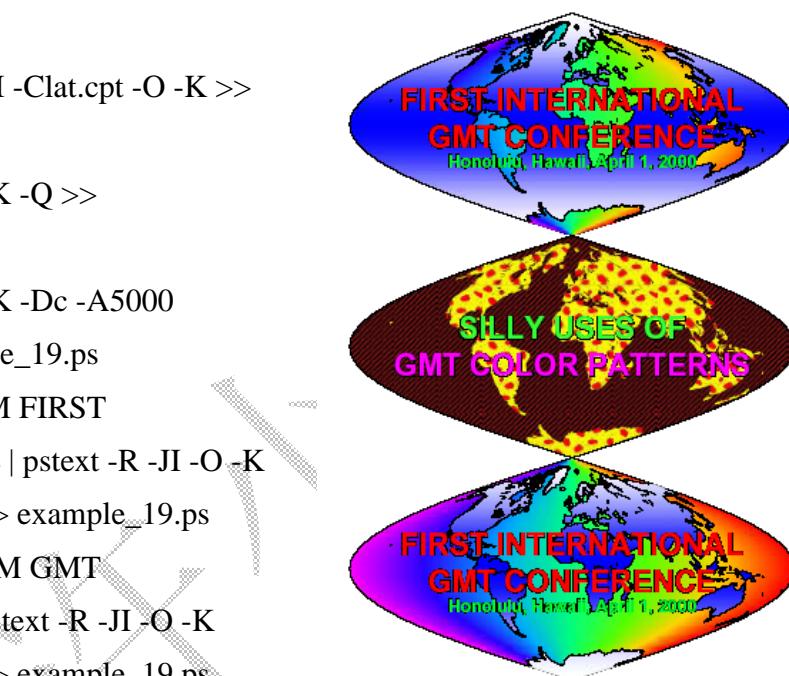
REM Then show example of color patterns

```
pscoast4 -R -JI -O -K -Dc -A5000 -Gp100/86:F255/0/0B255/255/0
```

```

-Sp100/7:F255/0/0B0/0/0 -B0 -Y-3.25i >> example_19.ps
echo 0 15 32 0 1 CM SILLY USES OF | pstext -R -JI -O -K -G50/255/50 -S0.5p >>
example_19.ps
echo 0 -15 32 0 1 CM GMT COLOR PATTERNS | pstext -R -JI -O -K -G255/0/255
-S0.5p >> example_19.ps
REM Finally repeat 1st plot but exchange the patterns
grdimage3 lon.grd -JI -Clon.cpt -O -K -Y-3.25i -B0 -U"Example 19 in Cookbook" >>
example_19.ps
pscoast5 -R -JI -O -K -Dc -A5000 -Gc >>
example_19.ps
grdimage4 lat.grd -JI -Clat.cpt -O -K >>
example_19.ps
pscoast6 -R -JI -O -K -Q >>
example_19.ps
pscoast7 -R -JI -O -K -Dc -A5000
-W0.25p >> example_19.ps
echo 0 20 32 0 1 CM FIRST
INTERNATIONAL | pstext -R -JI -O -K
-G255/0/0 -S0.5p >> example_19.ps
echo 0 -10 32 0 1 CM GMT
CONFERENCE | pstext -R -JI -O -K
-G255/0/0 -S0.5p >> example_19.ps
echo 0 -30 18 0 1 CM Honolulu, Hawaii, April 1, 2000 | pstext -R -JI -O -G0/255/50
-S0.25p >> example_19.ps
del .gmt*

```



GMT EXAMPLE 19 注释

Grdmath¹ -R-180/180/-90/90 -I1 -F Y COSD 2 POW = lat.grd

按照表达式 $\cos^2(y)$ 计算每 1 度纬度 (y) 的换算值，作为 z 值，写入 lat.grd

```

-172.5 82.5    7.61524e-005
-178.5 82.5    7.61524e-005
.....

```

grdmath² -R-180/180/-90/90 -I1 -F X = lon.grd

把每 1 度经度 (x) 的值, 写入 lon.grd

-172.5 82.5 -172.5

-178.5 82.5 -178.5

.....

echo 0 255 255 255 1 0 0 255 > lat.cpt

makecpt -Crainbow -T-180/180/60 -Z > lon.cpt

建立 2 个.cpt 文件

grdimage¹ 采用正弦曲线混合投影画上面影像图。由 lat.grd 和 lat.cpt 绘制彩色影象, z 值 0、1 对应颜色白、蓝。

pscoast¹ 绘制海陆界线, -Gc 剪裁去陆地区, 即只填充水体彩色影像。

grdimage² 由 lon.grd 和 lon.cpt 绘制陆地彩色影象, z 值由-180 度到 180 对应彩虹色紫到红。

pscoast² -Q 恢复 pscoast¹ 剪裁去的陆地区

pscoast³ 绘制海陆界线, 填充陆地彩色影象。

echo 0 20 分别书写 “FIRST INTERNATIONAL” 、“GMT CONFERENCE” 和 “Honolulu, Hawaii, April 1, 2000” 3 行文字

pscoast⁴ 下移 3.25i 画中间图。陆地填充分辨率 100dpi, 采用彩色模板 86, 前景红色, 背景黄色; 水体填充分辨率 100dpi, 采用模板 7, 前景红色, 背景黑色。;

echo 0 15 分别书写 “SILLY USES OF ” 和 “GMT COLOR PATTERNS” 2 行文字

grdimage³ 再下移 3.25i 画下图。由 lon.grd 和 lon.cpt 绘制彩色影象, z 值由-180 度到 180 对应彩虹紫到红。

pscoast⁵ 绘制海陆界线, -Gc 剪裁去陆地区, 即只填充水体彩色影像。

grdimage⁴ 由 lat.grd 和 lat.cpt 绘制陆地彩色影象, z 值 0、1 对应颜色白、蓝。

pscoast⁶ -Q 恢复 pscoast⁵ 剪裁去的陆地区

pscoast⁷ 绘制海陆界线, 填充陆地彩色影象。

echo 0 20..... 分别书写 “FIRST INTERNATIONAL” 、“GMT CONFERENCE” 和 “Honolulu, Hawaii, April 1, 2000” 3 行文字

4. GMT 缺省设置文件 (.gmtdefaults) 注释

ANOT_MIN_ANGLE = 20 ° (0° -90°) 标注坐标经纬度的最小角度

ANOT_FONT = Helvetica 标注坐标经纬度的字体, 可选 39 种字

体。

ANOT_FONT_SIZE = 14p	(>0) 标注坐标经纬度字符的大小 (像素数)
ANOT_OFFSET = 0.075i	(or 0.2c) 标注坐标经纬度的偏移量 (距图框距离)
BASEMAP_AXES = WESN	底图标注形式。大写: 画轴同时标注经纬度; 小写: 只画轴。
BASEMAP_FRAME_RGB = 0/0/0	底图图框颜色设定, RGB 模式, 每个颜色范围 0~255。
BASEMAP_TYPE = fancy	(/plain) 底图图框类型设定, 黑白相间或单线
COLOR_BACKGROUND = 0/0/0	背景色, z 值小于色谱表最小值时用
COLOR_FOREGROUND = 255/255/255	前景色, z 值大于色谱表最大值时用
COLOR_NAN = 128/128/128	, z 值为 NaN 时的颜色设置
COLOR_IMAGE = adobe	(/tiles)
COLOR_MODEL = rgb	(/HSV) 颜色模式设定
D_FORMAT = %lg C	语言语法, 输出精度为双精度浮点。
DEGREE_FORMAT = 0	(1/2/3/4/5), 经纬度标注格式
	0 经度 0~360, 纬度 -90~90。
	1 经度 -180~180, 纬度 -90~90。
	2 经度无符号 0~180, 纬度无符号 0~90。
	3 同 2, 但在经纬度后加 W、E、S、或 N。
	4 同 0, 但用十进制代替度、分、秒。
	5 同 1, 但用十进制代替度、分、秒。
DOTS_PR_INCH = 300	(dpi) 分辨率
ELLIPSOID = WGS	- 84 椭球模型
FRAME_PEN = 1.25p	单线型图框线宽, p 或 dpi。
FRAME_WIDTH = 0.075i	黑白相间型图框框宽, i 或 c。
GLOBAL_X_SCALE = 1	(>0)
GLOBAL_Y_SCALE = 1	(>0)
GRID_CROSS_SIZE = 0i	设经纬度线长度, 设为 0 时画连续的经纬度线。
GRID_PEN = 0.25p	设经纬度线线宽, p 或 dpi。
GRIDFILE_SHORTHAND = FALSE	(/TRUE), 如果设为 FALSE, 网格文件无扩展名
HEADER_FONT = Helvetica	图标题字体选择

HEADER_FONT_SIZE = 36p (>0) 图标题字大小
 HSV_MIN_SATURATION = 1 (0-1) 采用 HSV 颜色模型时最小饱和度
 HSV_MAX_SATURATION = 0.1 (0-1) 采用 HSV 颜色模型时最大饱和度
 HSV_MIN_VALUE = 0.3 (0-1) 采用 HSV 颜色模型时最小值
 HSV_MAX_VALUE = 1 (0-1) 采用 HSV 颜色模型时最大值
 INTERPOLANT = akima (/linear/cubic), 1-D 内插模型设定
 IO_HEADER = FALSE
 N_HEADER_RECS = 1 输入/输出 ASCII 码文件时, 文件标头记录数或无。
 LABEL_FONT = Helvetica 标注字体
 LABEL_FONT_SIZE = 24p (>0) 标注字大小
 LINE_STE = 0.01i (或 0.025c, >0) 用多段线绘制曲线时, 单个线段最大长度。
 MAP_SCALE_FACTOR = 0.9996 (>0) 图比例尺系数
 MAP_SCALE_HEIGHT = 0.075i (or 0.2c, >0) 图比例尺高度
 MEASURE_UNIT = inch (/cm/m/point) 用户单位
 N_COPIES = 1 拷贝数
 OBLIQUE_ANOTATION = 1 (/0), 标注倾斜坐标(经纬度)标记的方式
 0, 经度标记只标注在上下图框, 纬度标记只标注在左右图框。
 1, 标注在交切图框边界的任何地方。
 PAGE_COLOR = 255/255/255 纸面颜色设定
 PAGE_ORIENTATION = 横向 (/竖向) 打印纸方向设定
 PAPER_MEDIA = letter (/A4/ ..., 见 GMTDEFAULT) 打印纸型号选择
 PSIMAGE_FORMAT = hex (/bin) ps 图形文件格式
 TICK_LENGTH = 0.075i (或 0.2c) 坐标标记线长度
 TICK_PEN = 0.5p (p 或 dpi) 坐标标记线线宽
 UNIX_TIME = FALSE (/TURE)
 UNIX_TIME_POS = -0.75i/-0.75i (或 -2c/-2c) 绘制机内日期时间的起点坐标
 VECTOR_SHAPE = 0 (/1 绘制矢量图时箭头形状设定
 0, 箭头形状为三角形
 1, the head of a vector will be arrow .Inter mediate settings
 gives something in between
 VERBOSE = FALSE

```

WANT_EURO_FONT = FALSE      (/TURE)
X_AXIS_LENGTH = 9i          (or 25c) X 轴长度
Y_AXIS_LENGTH = 6i          (or 15c) Y 轴长度
X_ORIGIN = 1i               (or 2.5c) ,X 轴起点, overlay 方式, 缺省设置为 0.
Y_ORIGIN = 1i               (or 2.5c) ,Y 轴起点, overlay 方式, 缺省设置为 0.
XY_TOGGLE = FALSE           (x,y/TUREy,x)
Y_AXIS_TYPE = hor_text     (/ver_text)

```

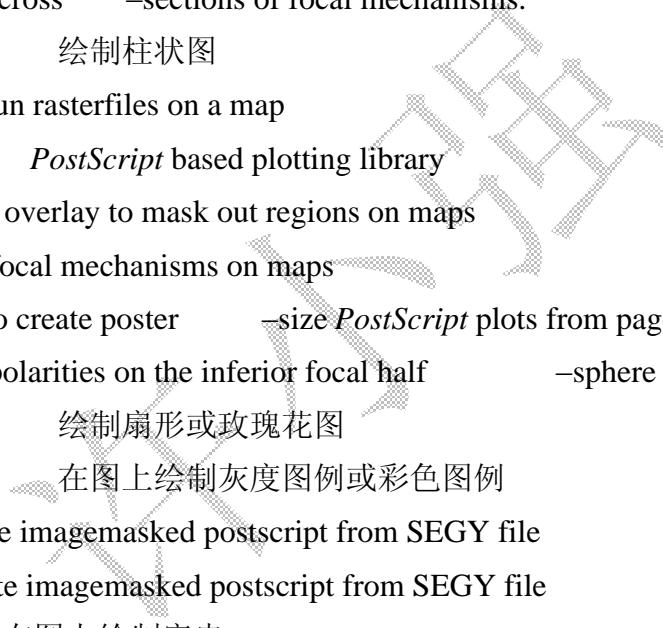
5. GMT summary

下面是 GMT 所提供的所有命令和其功能的很简单的说明。更详细的解释见使用手册或在线文件。

backtracker	Forward and Backward flowlines and hotspot tracks
binlegs	维护 GMT-SYSTEM 索引文件
blockmean L2	(x,y,z) table data filter/decimator
blockmedian L1	(x,y,z) table data filter/decimator
blockmode	Mode estimate (x,y,z) table data filter/decimator
cps	把 shell scripts 和数据文件插入到 Complete PostScript(CPS) 文件
cpsdecode	由 Complete PostScript (CPS) 文件提取 shell scripts 和数据文件
cpsencode	把 shell scripts 和数据文件插入到 Complete PostScript (CPS) 文件
dat2gmt	把一个 ASCII 文件转换为二进位 gmt 文件
filter1d Filter 1	-D table data sets (time series)
fitcircle	得到一组数据点的最佳拟合大圆或小圆
gmt GMT	数据处理和显示软件包
gmt2bin	由 gmt 文件创建二进制索引文件
gmt2dat	二进制 gmt 文件转换为 ASCII 文件
gmtconvert ASCII	和二进制 1-D 表文件之间转换
gmtconvert	数据表文件格式转换
gmtdefaults	当前缺省设置列表
gmtinfo	得到单独航次的信息
gmtlegs	查明一个给定地区的航次

gmtlist	由<legid>.gmt 文件提取数据
gmtmath	使用 Polish Notation 语法对表格数据进行代数操作
gmtpath	获得 gmt 文件的完全路径
gmtselect	选择基于多空间标准的数据子集
gmtset	改变当前的 .gmtdefaults 文件所选择的参数
gmttrack	一个航迹绘图程序
grd2cpt	由一个网格文件创建色谱表
grd2xyz 2	-D 网格文件转换为表格数据
grdclip	限定网格数据组的 z 值范围
grdcontour	绘制 2-D 网格数据组等值线
grdcut	由一个网格文件抽取一个子区
grdedit	改变一个 2-D 网格文件的标头信息
grdffft	Perform operations on gridded files in the frequency domain
grdfilter Filter 2	-D gridded data sets in the space domain
grdgradient	由网格文件计算方向梯度
grdhisteq	网格文件柱状图均化 (equalization)
grdimage	由一个 2-D 网格数据组创建影像图
grdinfo	得到网格文件的信息
grdlandmask	Create masking gridded files from shoreline data base
grdmask	Reset grid nodes in/outside a clip path to constants
grdmath	对网格文件进行代数操作
grdpaste	沿一条公共边界合并多个网格文件
grdproject	把网格数据组投影到一个新的坐标系
grdraster	Extract subregion from a binary raster and write a grd file
grdreformat	把网格文件转换为另一种格式
grdsample	对一个 2-D 网格文件重新采样创建一个新的网格
grdtrack	沿 1-D 轨迹对一个 2-D 网格数据组重新采样
grdtrend	对网格文件进行多项式趋势拟合
grdvector	绘制 2-D 网格矢量区
grdview	由一个 2-D 网格数据组创建 3-D 透视影像图
grdvolume	计算给定等值线以下的体积
hotspotter	Create CVA image from seamount flowlines
img2mercgrd	Extract region of img, preserving Mercator, save as grd
makecpt	创建色谱表文件

makepattern Make GMT color pattern from b/w pattern or icon
 mapproject Transformation of coordinate systems for table data
 mgd77togmt Convert an MGD -77 ascii file to a binary gmt file
 minmax 报告数据表文件极大极小值
 nearneighbor Nearest -neighbor gridding scheme
 originator Associate seamounts with hotspot point sources
 project Project table data onto lines or great circles
 psbasemap 创建底图
 psclip Use polygon files to define clipping paths
 pscoast 在图上绘制（和充填）海岸线、国界线和河流
 pscontour Contour or image raw table data by triangulation
 pscoupe Plot cross -sections of focal mechanisms.
 pshistogram 绘制柱状图
 psimage Plot Sun rasterfiles on a map
 pslib v3.2 A PostScript based plotting library
 psmask Create overlay to mask out regions on maps
 psmeca Plot focal mechanisms on maps
 psmegaplot To create poster -size PostScript plots from page-size plot
 pspolar Plot polarities on the inferior focal half -sphere on maps
 psrose 绘制扇形或玫瑰花图
 psscale 在图上绘制灰度图例或彩色图例
 pssegy Create imagemasked postscript from SEGY file
 pssegyz Create imagemasked postscript from SEGY file
 pstext 在图上绘制字串
 psvelo Plot velocity vectors, crosses, and wedges on maps
 pswiggle 沿地图上一条规机绘制根据时间序列数据绘制曲线
 psxy 在图上绘制符号、多边形和线条
 psxyz 在图上绘制 3-D 符号、多边形和线条
 sample1d 表格数据组重新采样
 spectrum1d Compute various spectral estimates from time -series
 splitxyz 把 xyz 文件分为几段
 surface 一种连续弯曲的网格算法
 trend1d Fits polynomial or Fourier trends to $y = f(x)$ series
 trend2d Fits polynomial trends to $z = f(x,y)$ series



triangulate Perform optimal Delauney triangulation and gridding
X_EDIT Convert between binary and ASCCI xocer correction tables.
XINIT Initialization of Cross –Over Data Bases.
XLIST To extract Cross –Over Information from the Data Base.
XOVER Find and compute Cross –Over Errors
XREMOVE Remove COEs for Selected Cruises from the Data Base
XREPORT Report Cross –Over Error Statistics for Cruises
X_SETUP Determine Pairs of Legs that need Cross –Over Checking
X_SOLVE_DC_DRIFT Find Best –fitting D.C.–shift and Drift Estimates
X_SYSTEM A Cross –Over Error Analysis Tool
X_UPDATE Archiving of Cross –Over Information
x2sys_cross Find and compute Cross –Over Errors
x2sys_datalist A generic data –extractor ascii or binary files
xyz2grd Convert an equidistant table xyz file to a 2 –D gridded file

6. GMT 常用命令介绍

6.1 psbasemap GMT 画底图的指令

一定要有的参数

-B 设定图边界尺标的间隔

a 文字批注的间隔

f 边框标记的间隔

g 隔线的间隔

如果要针对 X 轴或 Y 轴做不同的设定，就在参数间加斜线 /

NEWS 每个轴加上标记

news 每个轴不加标记

:.title: 在图上面加 title 这个标题

:title: 在轴上加 title 这个标题（只有在-Jx 的情况下有用）

例如-BNEWSf2a3g2:xx:/f3a3g2:yy:.title:

-Jx (-JX) 线性投影(Linear Projection)一般绘图用

-Jx 后面接的数字代表每单位的长度（预设长度单位为英吋，加 c 的话就变公分）

-JX 后面接的数字代表轴的全长

如果要针对 X 轴或 Y 轴做不同的设定，就在数字间加斜线 /

-Jm (-JM) 麦卡托投影(Mercator Projection)地图绘图用

-Jm 后面接的数字代表每度的长度（预设长度单位为英吋，加c的话就变公分）

-JM 后面接的数字代表整张图的宽

如果要针对X轴或Y轴做不同的设定，就在数字间加斜线/

-R 地图的经纬度范围

西/东/南/北 依序填入经纬度

>out.ps 把画出来的图输出到out.ps这个档案

不一定要有的参数

-E 旋转地图的视野方向

-G 把图着色，用r/g/b调色

-K 在这一行下面还有后续的指令

-O 把这一行指令绘制的图迭加到先前画的图上

-P 把图旋转90度

-V 画面上会显示本行指令的执行结果

-X 移动整个图的水平位置

-Y 移动整个图的垂直位置

表示方式

```
psbasemap -JM7 -R117/126/18/26 -BNEWsf2a3g2:xx:/f3a3g2:y:::test: -P -V  
-K >123.ps
```



6.2 pscoast GMT画陆块海洋和海岸线的程序

一定要有的参数

-J 投影法，和前面的一样

-R 地图的经纬度范围，和前面的一样

不一定要有的参数

-E 旋转地图的视野方向

-G 把干地着色，用r/g/b调色

-S 把湿地着色，用r/g/b调色

-D 图的分辨率(f, h, i, l, c)左往右为分辨率高至低

-I 加河流上去(1-10)大河至小河(a, r, i, c)大小河都画

-W 画海岸线

-N 画边界(1.国界，2.州界，3.海洋，a.全部)

-K 在这一行下面还有后续的指令

-O 把这一行指令绘制的图迭加到先前画的图上

-P 把图旋转90度

-V 画面上会显示本行指令的执行结果

-X 移动整个图的水平位置

-Y 移动整个图的垂直位置

表示方式

pscoast -JM7 -R117/126/18/26 -G50/50/50 -S150/150/150 -W1 -Di -V -O
-K >>123.ps

6.3 psxy GMT 在地图上画点线的程序

一定要有的参数

- J 投影法，和前面的一样
- R 地图的经纬度范围，和前面的一样

不一定要有的参数

- A 画大圆线
- B 和先前 psbasemap 的一样
- C 对照颜色，后面要接文件名称
- G 把符号着色，用 r/g/b 调色
- K 在这一行下面还有后续的指令
- N 不要跳过超出范围的点
- O 把这一行指令绘制的图迭加到先前画的图上
- P 把图旋转 90 度
- M 线分段
- Sa 在图上画星星，后面要接大小
- Sc 在图上画圆圈，后面要接大小
- Sd 在图上画钻石，后面要接大小
- Se 在图上画椭圆形，在资料的第 3, 4, 5 栏要有方向长轴短轴
- Sh 在图上画六角形，后面要接大小
- Ss 在图上画正方型，后面要接大小
- St 在图上画三角形，后面要接大小
- Si 在图上画反三角，后面要接大小
- Sv 在图上画向量，在数据的第 3, 4 栏要有方向长度
- Sx 在图上画叉叉，后面要接大小
- V 画面上会显示本行指令的执行结果
- W 笔宽
- X 移动整个图的水平位置
- Y 移动整个图的垂直位置



表示方式一 把数据直接打在档案里

```
psxy -JM7 -R117/126/18/26 -W1 -O -K -V<<! >>123.ps
120 23
124 27
!
```

表示方式二 把数据另外打在数据文件里

```
psxy 123.dat -JM7 -R117/126/18/26 -W1 -M -V -O -K >>123.ps
在 123.dat 的档案里是这样（这是画线条用的档案）
```

```
>
120 25
123 25
>
```

121 23

123 27

在 123.dat 的档案里是这样（这是点用的档案）

120 25

123 25

121 23

123 27

表示方式三 把更多的数据打在数据文件里

`psxy 123.dat -R117/126/18/26 -JM7 -Sc -Ccpt >> 123.ps`

在 123.dat 的档案里要有四栏数据，依序为 X, Y, 颜色代码，圆的大小

6.4 grdimage GMT 画二维等高图的指令

一定要有的参数

`grd` `grd` 档案的档名

`-C` 对照颜色，后面要接文件名称

`-J` 投影方式，和前面一样

不一定要有的参数

`-B` 和先前 `psbasemap` 的一样

`-K` 在这一行下面还有后续的指令

`-O` 把这一行指令绘制的图迭加到先前画的图上

`-P` 把图旋转 90 度

`-R` 地图的经纬度范围，和前面的一样

`-V` 画面上会显示本行指令的执行结果

`-X` 移动整个图的水平位置

`-Y` 移动整个图的垂直位置

表示方式

`grdimage file.grd -J0c122/21/22/69/4.8i -R117/126/18/26 -Cfile.cpt
-P -K -V >> 123.ps`



6.5 pscale GMT 画比例尺的指令

一定要有的参数

`-D` 定义比例尺的位置和大小

如果比例尺是水平的 `-D` 距中央/距顶端/长度/宽度 h

如果比例尺是垂直的 `-D` 距左端/距中央/长度/宽度

不一定要有的参数

`-B` 定义比例尺的间隔，和前面一样

`-C` 对照颜色，后面要接文件名称

`-K` 在这一行下面还有后续的指令

`-O` 把这一行指令绘制的图迭加到先前画的图上

- P 把图旋转 90 度
- V 画面上会显示本行指令的执行结果
- X 移动整个图的水平位置
- Y 移动整个图的垂直位置

表示方式

```
psscale -Cfile.cpt -D2.8/-1/4/0.22h -Bf1000a2000:"Elevation (m)":  
-O -K >> 123.ps
```

6.6 grdcontour GMT 画二维等高线图的指令

一定要有的参数

- grd** **grd** 档案的档名
- C** 等高线的间距
- J** 投影方式, 和前面一样

不一定要有的参数

- A** 等高线数值标示的间隔
- B** 和先前 psbasemap 的一样
- E** 视角
- G** 同一条等高线数值标示的间隔
- K** 在这一行下面还有后续的指令
- L** 等高线最低或最高的限制
- N** 等高线的单位 (如果有下-A 的参数才有效)
- O** 把这一行指令绘制的图迭加到先前画的图上
- P** 把图旋转 90 度
- R** 地图的经纬度范围, 和前面的一样
- T** 高地 (-T+) 或低地 (-T-) 加上标记
- V** 画面上会显示本行指令的执行结果
- X** 移动整个图的水平位置
- Y** 移动整个图的垂直位置

表示方式

```
grdcontour file.grd -JM7 -R117/126/18/26 -C0.025 -W1 -A1 -P -K -V >>  
123.ps
```



6.7 grd 数据的处理

转换成 xyz 值

```
grd2xyz file.grd -R117/126/18/26 > file.xyz
```

块状平均 (I 是精确度每多少度一格)

```
blockmean file.xyz -R117/126/18/26 -I0.1 > file2.xyz
```

转换成 grd 档

```
surface file2.xyz -R117/126/18/26 -I0.1 -Gfile3.grd
```

查询档案信息

```
minmax file.xyz
grdinfo file.grd
```

7. 实例

本想多做些实例，但由于时间关系，只能以后搞了。这里仅以中上扬子航磁为例。

```
#=====
#Created by Xu Xiaoqiang
#Northwest University,China
#=====

#=====hangci=====

#!/bin/sh
#=====
gmtset ANOT_FONT_SIZE 10
gmtset DOTS_PR_INCH 300
gmtset MEASURE_UNIT cm
gmtset PAPER_MEDIA a4
#=====

#sed -i '/Line/d' To.xyz

#=====
outps=To.ps
indata=To.xyz
outgrd=To.grd
region1="/0/18/0/17"
region2="101.04/117.96/21.52/35"
ticks="-B2g1"
proj1="x1c"
proj2="L110/20/25/47/16c"
wgjg="-I0.04"
tscpt=hangci.cpt

#=====

#xyz2 grd 把规整的（即等间距的 xyz 数据）xyz 数据（indata）转换成 GMT 格式的 grd 数据
#(outgrd)
xyz2grd $indata -R$region2 -G$outgrd $wgjg
#grd2cpt 根据（outgrd）产生一个颜色文件（tscpt），该文件可以手动编辑
grd2cpt $outgrd -V >$tscpt
#psbasemap 画一个底图，可有可无，视情况而定
```

```

psbasemap -B0 -R$region1 -J$proj1 -K -V>>$outps
#grdgradient Compute directional derivative or gradient from 2-D grid file representing z(x,y)
grdgradient $outgrd -A0 -Nt -Gtmp_int.grd
#由 grd 文件画彩色 image, 加-I 参数生成彩色阴影图, I 后数据为 grdgradient 生成的数据
grdimage $outgrd -J$proj2 -R$region2 $ticks -Itmp_int.grd -X1 -Y2.5 -C$tscpt -O -K -V>>$outps
#pscoast 画海岸线、国界等
pscoast -R$region2 -J$proj2 -Na -I1 -W1.5 -Dh -Cwhite -S30/144/255 -V -O -K -V>>$outps
#psscale 画 colorbar
psscale -C$tscpt -D8/-1/10/0.6h -I -V -O -K -B150/:nT: >>$outps
#grdcontour 画等值线
#grdcontour $grdfile $proj2 $region2 -S15 -C5 -V -P -K -O >>$outps
#psxy 画标记符号, 如地名符号, 该图画的就是地名符号 (小圆圈)
psxy cities.dm -R$region2 -J$proj2 -K -O -Sc0.15c -W2 -V >>$outps
#pstext 标记文字, 该图是标记地名, 地名在文件 cities.dm 里
gawk '{print $1, $2+0.1, $5+2, $6, $4, $7, $3}' cities.dm | pstext -R$region2 -J$proj2 -O -K
-V>>$outps

```

rm -f *.gmt* *.grd *.cpt

