

GMT 中文攻略 1.0 β 版

北京大学地球与空间科学学院
地球物理系
佟啸鹏

目录

前言	1
1 简介	1
2 GMT 概览和快速索引	3
3 GMT 的基本特征	5
3.1 GMT 中的单位制	5
3.2 GMT 的默认设置	5
3.3 命令行的参数选项	5
3.4 命令行的历史	6
3.5 用法信息, 语法和出错信息	6
3.6 标准输入(stdin)和头文件	6
3.7 PostScript 特征	7
3.8 覆盖和连续模式	7
3.9 指定线条属性	7
3.10 指定面积填充属性	8
3.11 调色板	8
4 GMT projection (GMT 中的投影)	9
4.1 非地图投影	9
4.2 圆锥投影	11
4.3 方位角投影 (Azimuthal Projection)	14
4.4 圆柱投影	18
5 食谱 Cook-book	21
6 附录	29
A. GMT 补充包	29
B. GMT 的文件格式	31
C. 制作 GMT 的 EPS 文件 (Encapsulated PostScript)	32
D. GMT 程序和相关资源的下载	33
K. 关于分辨率的一点说明	33
后记	34

前言

GMT 是一款与地球物理学联系紧密的绘图软件，它可以在地图上绘制地震，断层，画等值线，地形图，还可以进行简单数字运算，甚至谱分析等。它相对于其他软件（matlab,origin）的最大优势在于它针对地球工作者，提供了大量相关的功能和各种地图和断层数据。

目前，在网上关于 GMT 的网站大多是英文的，笔者根据英文的资料和自己的经验，翻译了英文手册（Generic Mapping Tools Technical Reference and Cookbook）中的主要内容，期望对后来人有所裨益。

GMT 的英文资源可以在 <http://gmt.soest.hawaii.edu> 中的镜像网站中找到。遇到问题可以联系作者： ttxpp@gmail.com

有一些说法，是我翻译过来的，不一定那么合适，所以我就想解释解释。文中经常提到数据和文件，其实我指的是一回事，程序和命令也是一回事。网格文件都是二维的，有时候省略二维，简写网格数据。

1 简介

大多数科学家一定熟悉从原始数据到分析处理再到图像显示这个过程。为了发表论文，准备报告，许多科学家花费大量的时间制作图示。这个过程很单调，还要用手工来完成，因为现有的软件只可以完成部分工作。为了克服这方面困难，我们推荐免费的 GMT 软件，它可以处理排成列的表格似的数据，时间序列，和定义在格子上的数据（grided data），还可以用不同的方式显示，有 x-y 坐标图，地图，3 维视图，有阴影的地貌图。GMT 用的是 PostScript 语言：PostScript page description language。用 PostScript，多种图形文件可以叠加，从而画出任意复杂的图形。GMT 相当于 UNIX 下的一个工具软件，（当然现在也有 windows 版本），并且它是独立的，有文档说明的。

最原始的 GMT1.0 是作者（Paul Wessel at School of Ocean and Earth Science and Technology and Walter H.F.Smith at Laboratory for Satellite Altimetry）在 1988 年从 Lamont-Doherty Earth Observatory of Columbia Univ. 毕业时发布的，后来 L-DEO 把它的运行环境改成 Unix，所以作者又写了 GMT 的 Unix 版本。它在那里很成功，后来又传播到美国，加拿大，日本的一些机构中去。由于过去用户的大量的建议，现在的版本有了很大改进，它包含了 50 种工具，25 种投影方式，使用起来更灵活。GMT 为科学家们提供了处理，显示数据的多种方式，包括数据采样，滤波，谱分析，计算时间序列的趋势，对空间分布的数据的网格化和插值，拟和数据等等。

GMT 是用具有高可移植度的 ANSI C 语言编写，与 POSIX 标准相适应。它经过少许改动就可以在所有运行 UNIX 软件的硬件上运行。在 GMT 的编写中，我们遵循了标准的 UNIX 的编写原则：把原始数据—分析处理—图像显示的流程分解为一系列基本的步骤，每一个步骤都可以用一个 UNIX 或 GMT 工具来完成。这个方法带来了如下的好处：（1）只需要少量的程序，（2）每一个程序简洁，并且易于更新，（3）每一个步骤都是独立的，因此可以有多种应用（每一步骤都可以任意组合），（4）程序可以用 shell 或其他的管道（pipes）连接到一起，这样就可以根据需要建立自己想要的数据处理流程。因为每一个科研机构的数据库结构不同，这样的话，数据获取和后续的处理，作图之间的解耦就很重要。如果我们想用自定义（custom）的数据库，我们就可以自己编一个程序把这种数据库的格式转化为 GMT 可读的格式。接下来就可以用标准的 GMT 函数了。

GMT 充分利用了 PostScript 语言，可以输出彩图。我们可以让一些专门的公司帮我们打彩页（应该是在美国）。而且，有一种软件 general-purpose PostScript raster image processors(RIPs)，可以生成 raster（光栅）图像，并在显示器，打印机上输出。因为画彩图的费用高，GMT 提供了 90 种图案，包括地质符号和灰色调的阴影图案。用户还可以自己绘制图案符号。

GMT 有完备的说明文档系统，包括技术参考手册和 cookbook(食谱)，他们详细说明很多软件包的特征和功能，为了使用户尽快熟悉系统，还提供了许多例子。cookbook 中有每一个例子的 shell 脚本和 PostScript 文件。每一个程序都有帮助页，可以用网页方式或者 UNIX 的 man 命令查询。对于用户的非法命令，系统会给出简短的提示，帮助用户自己改正。

GMT 里的处理和显示功能都很全面，可以处理(x,y),(x,y,z)的输入数据形式。很多(x,y)坐标用于显示经纬度，但他们也可以表示其他变量（波长，能量谱）。GMT 实际上是将(x,y)坐标用不同的变换方式投影到图上，所以对于输入数据，只要求他是(x,y),(x,y,z)的样子。为了简化和使输入输出标准化，GMT 只用了两种文件格式。第一种是 ASCII 码文件格式(x,y)或(x,y,z)数据按行和列的顺序排列，即表格数据。每一个坐标表示在一列中。（x 对应第一列，y 对应第二列）这种格式直观，方便用户用 UNIX 的工具 cut,paste,grep,sed,awk 对数据进行简单的处理。第二种是 grdfile。他就像是二维的数据分布在一个等距离的格子上。（网格化数据）在 GMT 里，我们就用 grdfile 读写他。这种基于 XDR(External Data Representation)格式的文件与本身的机器无关，这种二进制的文件就可以从一种机器系统到另一种机器系统之间任意传递。GMT 里，有程序专门用来把 ASCII 格式转化到网格化文件（grdfile），如 surface。

每一个程序都可能输出数据，这些数据可以分成四类，也有一些程序同时输出几种类型的数据。

1. 1-D ASCII 表格。例如，一个(x,y)序列被滤波，输出滤波后的结果。
2. 2-D 二进制文件（grdfile）。例如，把 ASCII 网格化或直接在 grdfile 上操作的程序都会输出 grdfile。
3. PostScript 简单说，就是图像文件。像 JPEG,bmp 那样的。

4. Report 执行 GMT 程序后的说明，报告输入数据和统计数据。几乎每一个文件都有 verbose 选项，报告程序的计算流程。

2 GMT 概览和快速索引

下面是 GMT 提供的所有的程序的简单介绍。想要知道更多的内容，请查阅 GMT 手册和帮助网页。后面的内容和例子也会较详细地解释。有些命令的中文解释我实在没有想好，所以会翻译的不好，请不要着急，因为读者可以通过后面的内容和一些实践理解。

一维，二维数据的滤波	
Blockmean	网格数据的算术平均
Blockmedian	网格数据的中位数平均
Blockmode	网格数据的模数平均
Filter1d	滤波一维数据（时间序列）
Grdfilter	滤波二维数据

一维，二维数据的作图	
Grdcontour	画二维网格数据（文件）的等值线
Grdimage	用二维网格数据制作图像
Grdvector	用二维网格数据制作矢量场
Grdview	用二维网格数据制作 3D 的图像
Psbasemap	建立边框
Psclip	根据一个多边形文件，剪切出一块图形
Pscoast	绘制海岸线，陆地，河流，政治边界
Pscontour	直接用三角差值（triangulation）处理 xyz 数据
Pshistogram	画等高线
Psimage	绘制彩色图
Psmask	一种选择合适数据的方法
Psrose	绘制玫瑰图（在风向风强表示中）
Psscale	画彩色或灰色标尺
Pstext	绘制文本
Pswiggle	在地图上画波形文件，见例子
Psxy	利用二维 xy 数据绘制：线，多边形，符号
Psxyz	利用三维 xyz 数据绘制：线，多边形，符号

表格数据（xy,xyz）的网格化	
Nearneighbor	最近点方法
Surface	连续曲率方法
Triangulate	最优化 Delauney 三角化方法

一维，二维数据的取样(sample)	
--------------------	--

Grdsample	对原先的二维网格数据的再次采样
Grdtrack	沿一条直线对原先的二维网格数据的再次采样
Sample1d	对原先的一维网格数据的再次采样

投影和地图转化	
Grdproject	将二维网格数据变化到另一个新坐标下
Mapproject	将表格数据变化到另一个新坐标下
Project	将数据投影到直线或者大弧上

信息查询	
Gmtdefaults	列出当前的默认设置
Gmtset	用命令行编辑 .gmtdefaults 中的参数
Grdinfo	得到网格化数据的内容
Minmax	得到表格数据的最大最小值

多彩多样的工具	
Gmtmath	对于表格数据的, Reverse Polish Notation (RPN) 格式的计算器
Makecpt	制作 GMT 的调色板文件, 为图像设置颜色
Spectrum1d	时间序列的谱分析
Triangulate	最优化 Delauney 三角化方法

数据的转化和提取	
Gmtconvert	将表格数据从一种格式转化到另一种格式
Gmtselect	依据多种的空间的标准从表格数据中选择子数据
Grd2xyz	将网格数据转化成表格数据
Grdcut	从网格数据剪切出一部分
Grdpaste	将两个网格数据按共同边界粘和
Grdreformat	将网格数据从一种格式转化到另一种格式
Splitxyz	把表格数据分开成几段
Xyz2grd	将表格数据转化成网格数据

在一维, 二维数据中的拟和	
Fitcircle	找到最佳拟和的小圆或大圆
Grdtrend	多项式拟和网格数据 ($z=f(x,y)$)
Trend1d	多项式或者傅立叶拟和 ($y=f(x)$)
Trend2d	多项式拟和数据 ($z=f(x,y)$)

其他对网格数据的操作	
Grd2cpt	根据网格数据制作调色板文件
Grdclip	对于网格数据, 在 z 轴上加限制
Grdedit	编辑网格数据的头文件 (header info)
Grdfft	从频域对网格数据操作

Grdgradient	从网格数据中计算方向导数
Grdhisteq	画网格文件的等高线
Grdlandmask	根据海岸线，决定某个点在海上，还是在陆上
Grdmask	将网格数据中一条剪切线的内部或者外部的格点置为常数
Grdmath	对于网格数据的，Reverse Polish Notation (RPN) 格式的计算器
Grdvolume	轮廓图中，计算某一个表面下的体积

3 GMT的基本特征

这一部分的内容很重要，介绍了在 GMT 中的对许多程序通用的特征，总结了这个软件的理念。记住程序的语法是次要的，主要的是理解记住下面的东西。再配合例子我们就可以轻松掌握 GMT 了。

3.1 GMT中的单位制

GMT 接受许多单位制，如 **cm,inch,meter,point**. 有两种办法可以让 GMT 知道你想使用的单位。

- 1 直接在数字后面附加上你的单位。例如表示 4 厘米：4c.
- 2 在 default 文件中，设置 MEASURE_UNIT 为你想要的单位，那么以后的数据就都是这个单位了，除非你用 1 中的方法再次指定单位。

3.2 GMT的默认设置

在 GMT 里面有关于各种图形，数据的五十多个参数。当一个程序被运行，首先它会把这些参数都设置为默认值。（在 `gmt.conf` 中可以设置根据国际单位制还是美国单位制）接着，它会在当前目录中或者你的主目录中查找 `.gmtdefaults` 文件。如果成功找到，它会根据该文件重新设置参数。利用它，你可以决定地图中线条的粗细，颜色，注释中的字体，图像的 dpi 等等。从帮助手册中你可以找到更详细的参数描述。你可以自己改变这里的参数（运行 `gmtdefaults` 或 `gmtset`），来实现自己想要的效果。请注意：你在命令行中的任何参数设置都会覆盖你的原先的设置。例如，你在 `.gmtdefaults` 中设置 x 的间隔为 1i，命令行中的 `-X1.5i` 选项会覆盖原先的 1i。

3.3 命令行的参数选项

每一个程序的运行都要求你提供一定的参数选项。大多数程序是区分大小写的；几乎所有的选项都以短线符和一个大写字母开头，紧接着是具体的数字或字符。短线符，大写字母，数字或字符中间没有空格，而选项之间一定要有空格。下面

的例子是画一个海岸线的轮廓出来。

```
pscoast -R0/20/0/20 -G200 -JM6i -W0.25p -B5 -V > map.ps
```

选项	意思
-B	为边框，轴线定义tickmarks ¹ ，注解，标签
-H	表示输入的表格文件有头文件
-J	选择一种地图投影或者非地图投影方式
-K	允许在这个作图命令后面追加新作图命令（连续）
-O	在一个已经存在的图上继续作图（覆盖）
-P	选择图的摆放方式为肖像图方向（默认的是风景图方向），前者是竖直，后者是水平
-R	定义图的边界区域。（下/上/左/右）
-U	画一个时间戳
-V	verbose.报告程序的计算流程
-X	设置图的坐标原点的位置 x 方向
-Y	设置图的坐标原点的位置 y 方向
-c	设置作图的份数
:-	输入的地理坐标是（纬线/经线）而不是默认的经线/纬线

3.4 命令行的历史

在 GMT 中，输入的命令行的参数会被记录下来。比如你之前输入了：

```
-Joc190/25.5/327/56/1:500000
```

当你再次作图，需要输入上面同样的数据时，你只要输入-Jo 就可以了。之前的命令被存放到了 .gmtcommands 文件中。在每一个你运行过 GMT 的目录中都有这样的文件。注意这个记忆只能记最近的一条参数。

3.5 用法信息，语法和出错信息

每一个程序都携带了用法信息。如果你只输入程序名，那么就会显示这一个程序的完整信息。包括每一个参数的详细用法。如果你输入程序名后跟了一个短线符，那么只会显示出一个简化的信息。如果你输入错了，系统还会给出相应的简短的提示。

3.6 标准输入(stdin)和头文件

大多数的程序可以从文件中或者标准输入²中读取数据。如果你在程序中，没有指定文件名，那么系统会从标准输入中读数据。在程序中，短竖线所引导的内容是参数，如果没有短竖线，系统会理解它为文件名。

¹ 图表或轴上的小竖线

² 一般的就是指屏幕输入

如果你的 ASCII 文件带有一个或多个头文件（这里的头文件不是通常讲的文件，而是数据前面一些用来解释说明的行），你一定要设 `-H` 参数项。头文件中记录的数目可以在 `.gmtdefaults` 中更改也可以直接用 `-Hn_header_recs` 更改。在 ASCII 文件中，有可能在文件的中间也有头文件记录分割数据文件。对于网格数据，没有头文件。

3.7 PostScript 特征

PostScript (PS) 是一种图形设备的命令语言。它的一个很大的优点是可以让用户轻松地修改已经创建的图形。高级的用户还可以通过 PS 文件携带的信息，了解图形是如何被创建的。另外，用户可以更改 `.gmtdefaults` 中的 `PAPER_MEDIA` 设置，使得输出的文件是 Encapsulated PostScript (EPS)。这样的文件可以用软件 **IslandDraw** 或者 **Adobe Illustrator** 进一步的修饰。

3.8 覆盖和连续模式

一个典型的 PS 文件有开头，中间和结尾。开头定义了建立图形必须的一些基本信息，中间是命令，结尾告诉图形设备输出图像并重置状态为原始值。有时候，一个最终图像是几个图像的叠加。如果是这样，第一个文件只需要一个开头，而没有结尾，最后一个文件正相反，有结尾，没开头，其他的文件就只有一个中间部分。为了达到这个目的，就要用到覆盖和连续模式 (`-O`, `-K`)。覆盖选项告诉系统这次作图的结果覆盖在前一次之上，这样文件就没有开头；连续选项告诉系统这次作图后，还有下一张图会继续画，这样这个文件就没有结尾。当然，如果你只用一个命令行，那么你就没必要考虑这个。

3.9 指定线条属性

在 GMT 中，线条有三种属性：粗细，颜色和形式。基本上所有的程序都用这样的一种格式：

`-Wwidth[/color][ttexture][p]`

粗细的单位是根据当前显示设备的分辨率而定义的。（由 `.gmtdefaults` 中 `DOTS_PR_INCH` 的值决定）。例如你的仪器分辨率是 300dpi,那么粗细的单位就是一英寸的 1/300。如果你加了 **p** 字母，表示你定义粗细的单位就是一英寸的 1/72，而不管仪器的分辨率是什么。例如 `-W5` 就是 5 个粗细单位，而 `-W5p` 就是 5/72 英寸。0 是最细的笔了，当然它也与仪器有关。

颜色可以用灰色系的 0-255 来设定。也可以用 RGB 彩色系来定：`r/g/b`。

形式 (texture) 控制着线条的样子。画一条点线，用 `-to`；画一条虚线，用 `-ta`。你还可以利用 `tstring:offset` 的语法，自己定义线条的样子。举例来说，画一条宽为 2，黄色的虚线，线的间隔是 10。虚线的样子是：20 有，10 空，5 有，10 空。

(这些数的单位是就粗细的单位) 应该这样设定: -W2/255/255/0t20_10_5_10:10

3.10 指定面积填充属性

在画图中, 可能会遇到画符号或者多边形, 为他们填颜色的格式有两种:

-Gfill

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

第一种情况和线条时的一样: 可以填充灰色的 (0-255), 也可以填充彩色的 (r/g/b)。第二种情况更复杂一些, 也不常见。笔者不打算详细地介绍。简单地说, 它是用一种类似瓷砖的小图填充到原图中去, 就像给地面铺瓷砖一样。(类似于地质符号) 如果您想知道具体细节, 请查阅英文版的手册。(附录 E)

3.11 调色板

有一些程序读入网格化数据, 创建彩色图形或者有阴影的地形图。在这个过程中, 他要知道使用什么颜色或者颜色的范围。为了实现这样的目的, 我们需要调色板文件。(Color Palette Table, cpt-file) 在 psxy,psxyz 命令中, 也可能用到这样的文件。我们可以改变.gmtdefaults 中的参数 COLOR_MODEL, 来选择使用 RGB, 还是 HSV 颜色系统。使用 RGB, cpt 文件的格式是这样的:

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]

z 定义了一个坐标, 在这个上面, 颜色可以是常数, 也可以线形变化。[A]是调色板的一个注解。他可能是 L, U, B。分别表示 z 坐标的上界, 下界, 或既是上界也是下界。

同上一个填色属性一样, 一些程序中可以用一些“图样”来填充 (grdview,psyscale,psxy)。举个例子:

```
30  p200/16    80    -
80   -         100    -
100  255 0 0    200  255 255 0
```

表示: 当 $30 < z < 80$, 图形被 200dpi 的 16 号图样填充, 当 $80 < z < 100$, 没有任何的设定, 当 $100 < z < 200$, 被填充的颜色从红逐渐过渡到黄。

一些程序如 **grdimage** 和 **grdview** 利用人工的光源效果创建有阴影的地形图。我们可以找到某一个方向上的方向导数, 再按一定的方法归一化到 $<-1, +1>$ 。这样的数据被用来附加到调色板文件上面。0 对应原始颜色, 高值对应颜色变亮, 低值对应颜色变暗。这样的方向导数的数据可以用 **grdgradient** 得到, 用 **grdhisteq** 修改。它 (梯度文件) 和 **cpt** 文件是两个不同的独立的文件。例如我们可以绘制海底地形图及对应海底深度的散射强度 (通过声纳系统), 再用前者制造某一个方向的照明效果; 我们可以绘制地形图和对应的磁异常强度, 用后者提供颜色再用

前者提供阴影效果。

4 GMT projection (GMT 中的投影)

GMT 程序需要知道如何把输入的坐标转化到图上的点，我们要赋予它一个特定的投影方式。这一节的目的就是总结 GMT 中的所有的投影方式：他们的参数，如何用他们建立图形。我们在以下的例子中会用到 `psxy`, `pscoast` 的命令。这两个命令很好学，所以我想读者可以通过例子掌握，没必要具体的介绍了。注意：这些例子全都是用 `inch(i)` 做单位，读者可以自己改变成 `cm`, `m`, `points(p)`。

注意：

在设定投影时，要是想定义图的比例尺，用小写字母；要是想定义图的宽度，用大写字母。

通常尺度的选取是根据经验，都在 `0.1i`—`10i` 之间吧。屏幕不过 `14i`, `17i` 嘛。

`-Amin-vlaue` : `pscoast` 在画内陆的湖泊时，会用闭合的多边形。`-A` 指示程序不画面积比 `min-vlaue` 小的多边形。单位是平方千米。

4.1 非地图投影

其实就是与地图无关的投影，类似于数学课上的作图。（姑且称之为数学投影吧）投影方式可以分成线性，对数，指数投影。`Y` 轴和 `x` 轴可以用不同的投影方式。为了演示例子，我们先来建立一个小的数据。用到了 `gmtmath`，运算的顺序是 RPN (Reverse Polish Notation), (RPN 是一种计算表达式的规则，一个波兰人搞的，在计算机里常用，如 `1 2 3 + *`，就表示 $(2+3)*1$)

```
#!/bin/sh
#
#   $Id: GMT_dummydata.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#
#   This script makes the dummy data sets needed in Section 5.1

gmtmath -T0/100/1 T SQRT = sqrt.d
gmtmath -T0/100/10 T SQRT = sqrt.d10
```

上面的两个命令创建了一个小数据，第一个表示：`T` 从 100 增加到 1 步长 1，并且计算 `T` 的平方，存储在 `sqrt.d` 中。

4.1.1 笛卡尔坐标下的线性投影（`-Jx -JX`）

用这种投影，我们要给出每个单位数对应的长度（`-Jx`）或者总的轴长（`-JX`）。如果 `Y` 轴和 `x` 轴的标尺不一样，我们可以这样表示：`-Jx0.1i/0.5i` 或 `-JX8i/5i`。

例子如下：

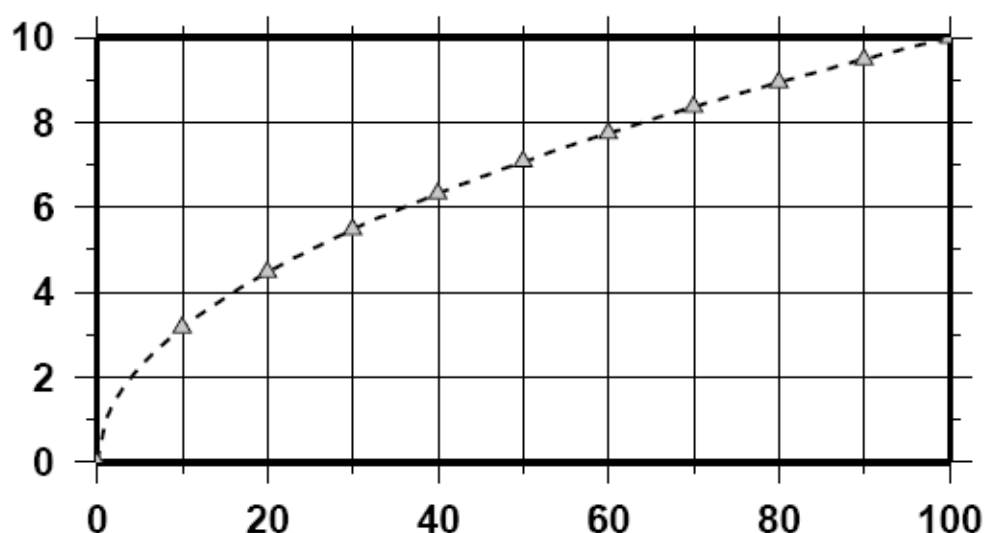


Figure 5.1: Linear transformation of coordinates

```
#!/bin/sh
# $Id: GMT_linear.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#
psxy -R0/100/0/10 -JX3i/1.5i -Ba20f10g10/a2f1g2WSne -Wlt3_3:0p -P -K sqrt.d > GMT_linear.ps
psxy -R -JX -St0.075i -G200 -W -O sqrt.d10 >> GMT_linear.ps
```

正常地，X 轴会向右增加，Y 轴会向上增加。如果需要的话，我们可以指定 X 轴向左，Y 轴可以向下。只要加个负号就可以了。（-Jx-0.1i/-0.5i）

4.1.2 对数投影

想画对数图，只要在-Jx -JX 后面加一个小写的 l 就可以了。如果两个轴都是对数的，就要加两遍。-Jx0.1il/0.5il

4.1.3 指数投影

它可以显示 x^a 与 y^b 。意思就是 Y 轴和 x 轴可以使指数增加的。我们可以在-Jx -JX 后面加一个 pvalue (value 表示 a, b 的大小)。如我们想让 x 轴以 0.5 为幂级数增加，y 轴线性增加，就可以这样输入：-Jx0.3ip0.5/0.15i
这样的话，x 的平方根就和 y 的值在一条直线上了。

4.1.4 地理坐标的线性投影

虽然这种线性投影主要用在广泛的数学投影上，有时候，也需要用线性投影的方

式作地图投影。这时候我们只需要附加一个 **d**，就可以了。如-Jx0.14id。

4.1.5 极坐标投影

很多情况下，我们用极坐标投影比用直角坐标好。用法如下：

-Jp inches/unit or -JP length (-Jp0.1i -JP4i)

可选项:加一个 a, 表示角度增加的方向不是默认的逆时针方向，而是顺时针方向。(-Jp0.1ia -JP4ia)

可选项加一个/角度，可以指定角度的起始点。(-Jp0.1ia/45)

例子：

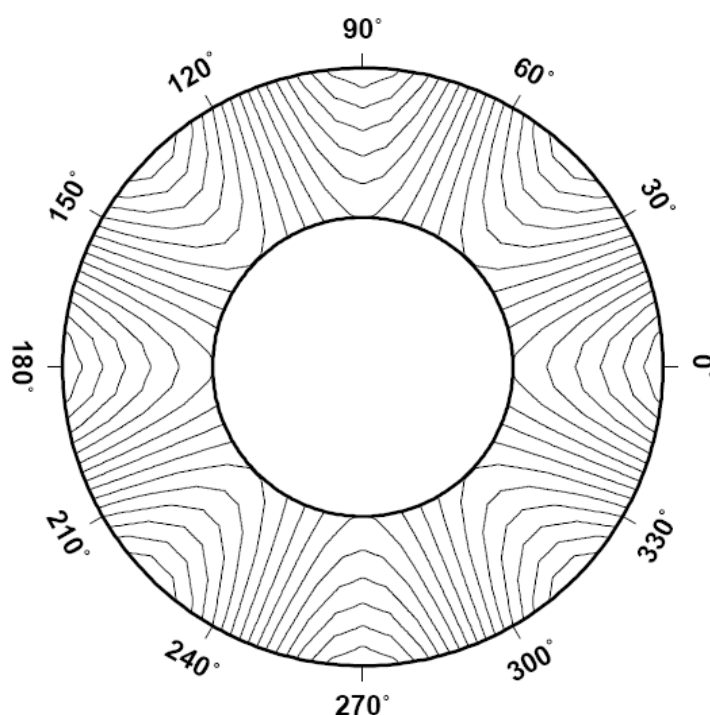


Figure 5.5: Polar (Cylindrical) transformation of (θ, r) coordinates

```
#!/bin/sh
# $Id: GMT_polar.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

grdmath -R0/360/2/4 -I6/0.1 X 4 MUL PI MUL 180 DIV COS Y 2 POW MUL = test.grd
grdcontour test.grd -JP31 -B30Ns -P -C2 -S4 > GMT_polar.ps
```

4.2 圆锥投影

4.2.1 Albers Conic Equal-Area Projection(-Jb -JB)

Albers 在 1805 年提出了这种投影，它主要用在东西向距离较大时，特别是对于

美国那样的。(跨纬度 20 几度, 跨经度 60 多度)。这是一个圆锥等面积投影, 平行线 (即纬度线) 是一个同心圆环的一部分弧。他们是不等距的, 南北极的纬度线间距较小。(如图) 子午线 (经线) 是由一个公共点发射出来的直线, 与纬度线垂直相交。在各个投影中都会有近似。那么, 在这个投影中会有两条标准的平行线, 这两条线的形状和尺度都是准确的。在这两条线之间的平行线, 尺度被缩短了; 在这两条线之外的平行线, 尺度被拉长了。对于子午线情况也是一样的。我们需要输入:

投影中心的经纬度

两条平行的纬度线

投影的标尺 (inch/degree or 1:xxxxx, map width)

例子中我们选定投影中心的经纬度: 125E/20N 两条平行的纬度线: 25N, 45N. 我们定图的宽度是 5 英寸。(1 英寸等于 2.54cm)

```
#!/bin/sh
# $Id: GMT_albers.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset GRID_CROSS_SIZE 0
pscoast -R110/140/20/35 -JB125/20/25/45/51 -B10g5 -D1 -G200 -W0.25p -A250 -P > GMT_albers.ps
```

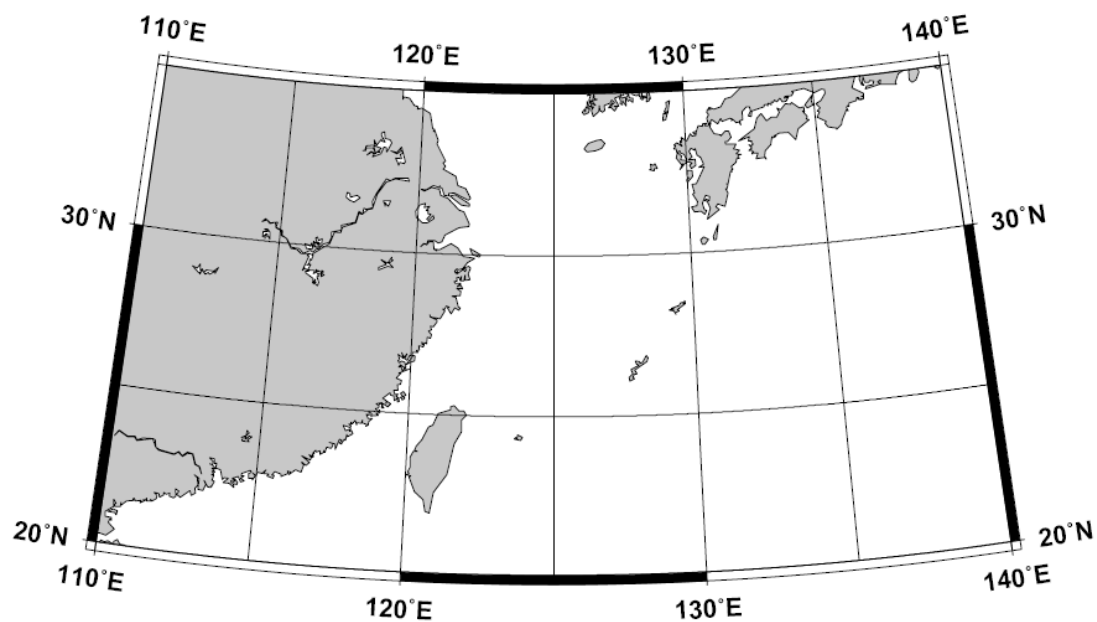


Figure 5.6: Albers equal-area conic map projection

4.2.2 Lambert Conic Conformal Projection (-J1 -JL)

与 Albers 类似, Lambert 投影也是用在东西向距离占主导时。但是它不是等面积投影, 而是正形 (conformal) 的, 他的输入参数与 Albers 一样。

投影中心的经纬度

两条平行的纬度线

投影的标尺 (inch/degree or 1:xxxxx, map width)

例子中 Lambert 投影被用来画美国大陆的部分。

注意: 投影中心的经纬度不影响投影的方式, 但是决定了那一条子午线在图上是

垂直的。

```
#!/bin/sh
# $Id: GMT_lambert_conic.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset BASEMAP TYPE FANCY DEGREE FORMAT 3 GRID_CROSS_SIZE 0.051
pscoast -R-130/-70/24/52 -Jl-100/35/33/45/1:50000000 -B10g5 -Dl -N1/1p -N2/0.5p -A500 -G200 \
-W0.25p -P > GMT_lambert_conic.ps
gmtset GRID_CROSS_SIZE 0
```

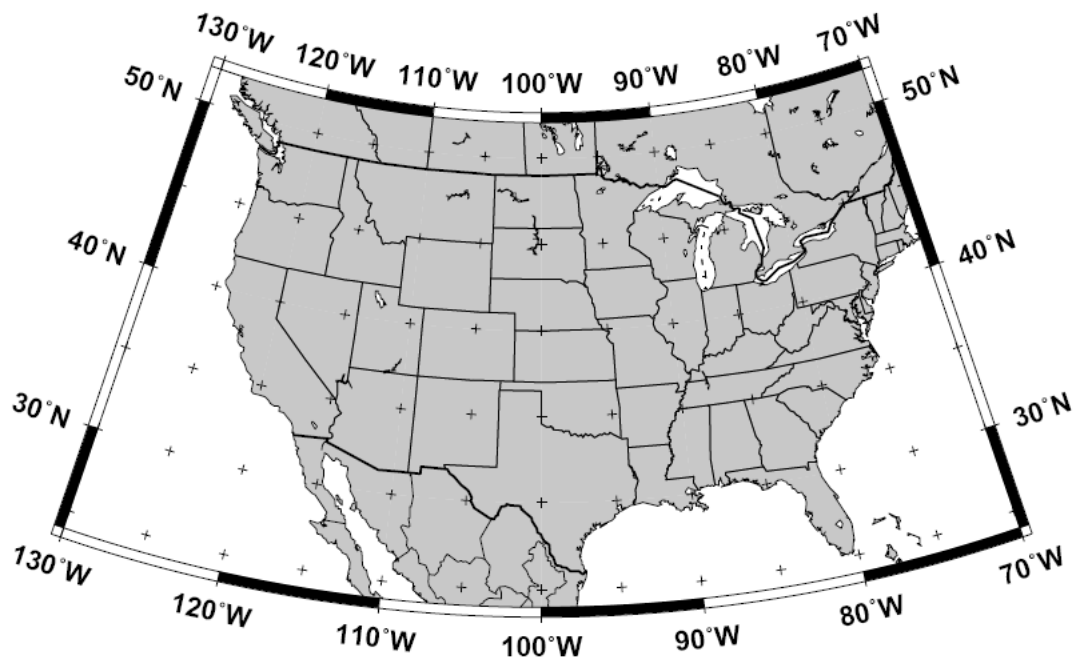


Figure 5.7: Lambert conformal conic map projection

4.2.3 Equidistant Conic Projection (-Jd -JD)

在公元 150 年，希腊哲学家 Claudius Ptolemy 描述过这种投影方式。他既不是正形的，也不是等面积的，而是介于前两种投影之间。我们经常用它绘制东西向跨度较大的小国家的地图。用法与上面的一样。

```
#!/bin/sh
# $Id: GMT_equidistant_conic.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset DEGREE_FORMAT 3 GRID_CROSS_SIZE 0.051
pscoast -R-88/-70/18/24 -JD-79/21/19/23/4.51 -B5g1 -Dl -N1/1p -G200 \
-W0.25p -P > GMT_equidistant_conic.ps
gmtset GRID_CROSS_SIZE 0
```

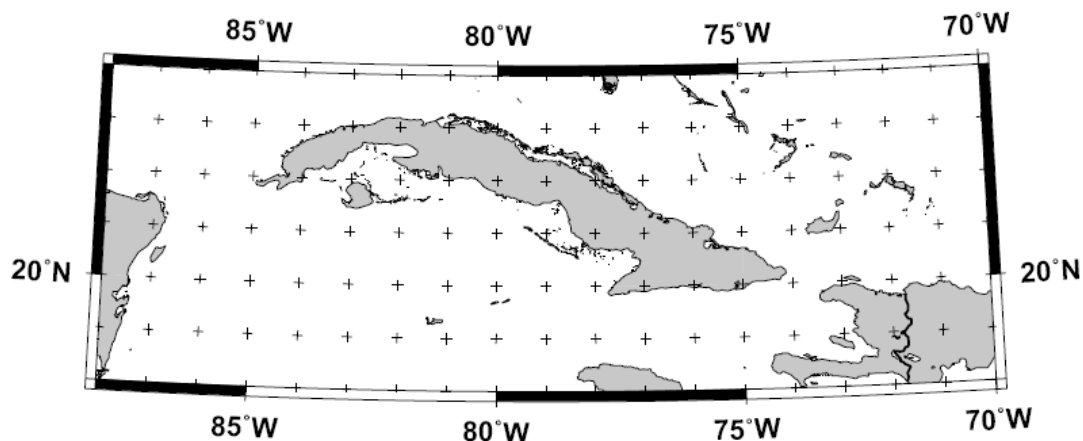


Figure 5.8: Equidistant conic map projection

4.3 方位角投影（Azimuthal Projection）

4.3.1 Lambert Azimuthal Equal-Area (-Ja -JA)

这种投影是由 Lambert 在 1772 年提出的，经常被用来大范围的投影（半球或是整个大陆）它是方位角的等面积的，但不是透视的。其实它就是我们地质学家说得 Schmidt 投影。在图的中心投影最精确，离它越远扭曲越大。我们需要输入：

投影中心的经纬度

投影的标尺（radius/latitude or l:xxxxx or map width）

根据画图区域（-R 选项控制）的不同，我们可以有两种类型：

长方形

通常情况下，我们用 -R 定义地图的东西南北（东西用经线，南北用纬线）。但是在这种投影方式下，-R 只是定义了地图的两个角（加一个 **r**）。这么定义的原因是经纬线在这种投影下，已经不再是直线了，不能用来做地图的边界线了。在例子中，我们定义 0E/40S 是图的左下角，60E/10S 是图的右上角。

```
#!/bin/sh
# $Id: GMT_lambert_az_rect.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset DEGREE FORMAT 0 GRID CROSS_SIZE 0
pscoast -R0/-40/60/-10r -JA30/-30/4.51 -B30g30/15g15 -D1 -A500 -G200 -W0.25p -P > \
GMT_lambert_az_rect.ps
```

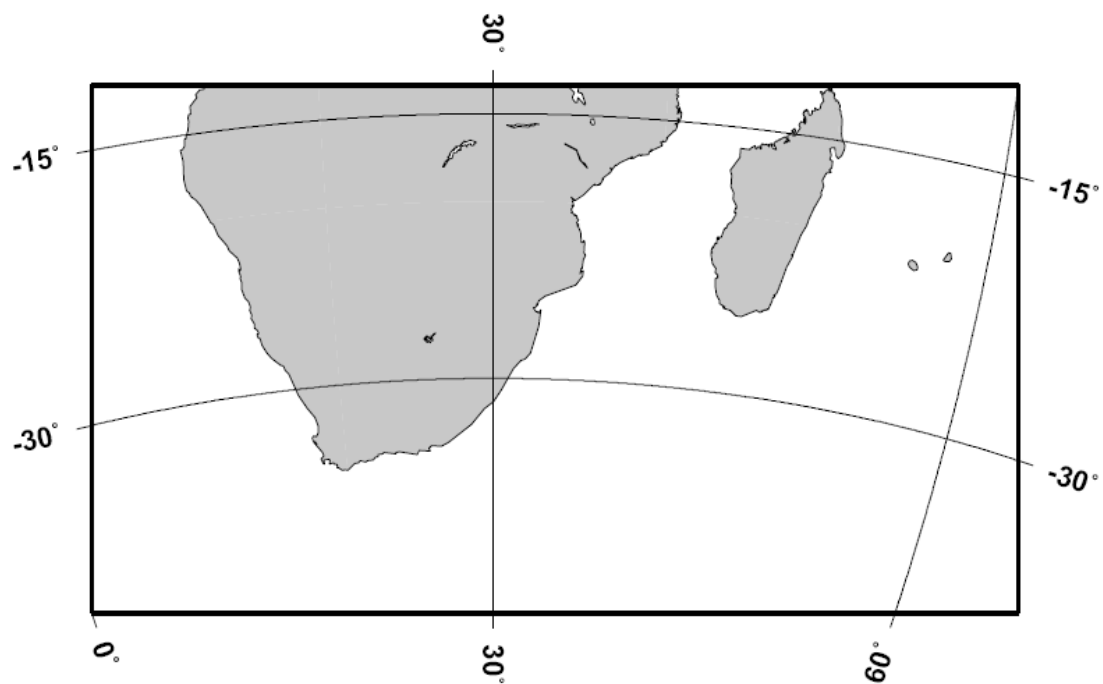


Figure 5.9: Rectangular map using the Lambert azimuthal equal-area projection.

半球

如果把画图区域扩大到全球（-R0/360/-90/90），我们就得到了一张半球投影。

```
#!/bin/sh
# $Id: GMT_lambert_az_hemi.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#
pscoast -R0/360/-90/90 -JA280/30/3.51 -B30g30/15g15 -Dc -A1000 -G0 -P > GMT_lambert_az_hemi.ps
```



4.3.2 立体等角度投影 Stereographic Equal-Angle Projection (-Js -JS)

这种投影同样可以追溯到希腊，它主要用来画极区（高纬度）的地图。

我们需要输入：

投影中心的经纬度

投影的标尺（radius/latitude or 1:xxxxx or slat/1:xxxx or map width）

立体等角度投影也有两种类型

我们可以利用-R 的改变来定义它。如果是-R-30/30/60/72 就会画出来一个楔形的，如果是-R-25/59/70/72r 就是画一个长方形，两个角是给定的。

```
#!/bin/sh
# $Id: GMT_stereographic_polar.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset DEGREE_FORMAT 1
pscoast -R-30/30/60/72 -Js0/90/4.51/60 -Ba10g5/5g5 -D1 -A250 -G0 -P > GMT_stereographic_polar.ps
```

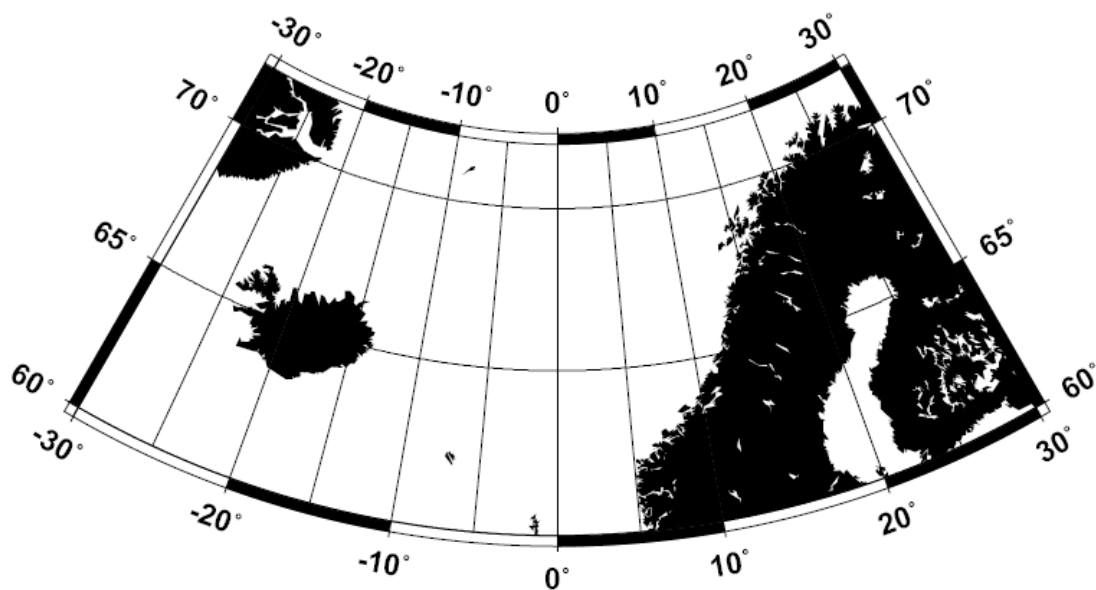


Figure 5.12: Polar stereographic conformal projection.

```
#!/bin/sh
# $Id: GMT_stereographic_rect.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset DEGREE_FORMAT 1 OBLIQUE_ANOTATION 30
pscoast -R-25/59/70/72r -JS10/90/11c -B30g10/5g5 -D1 -A250 -G200 -W.25p -P > \
GMT_stereographic_rect.ps
```

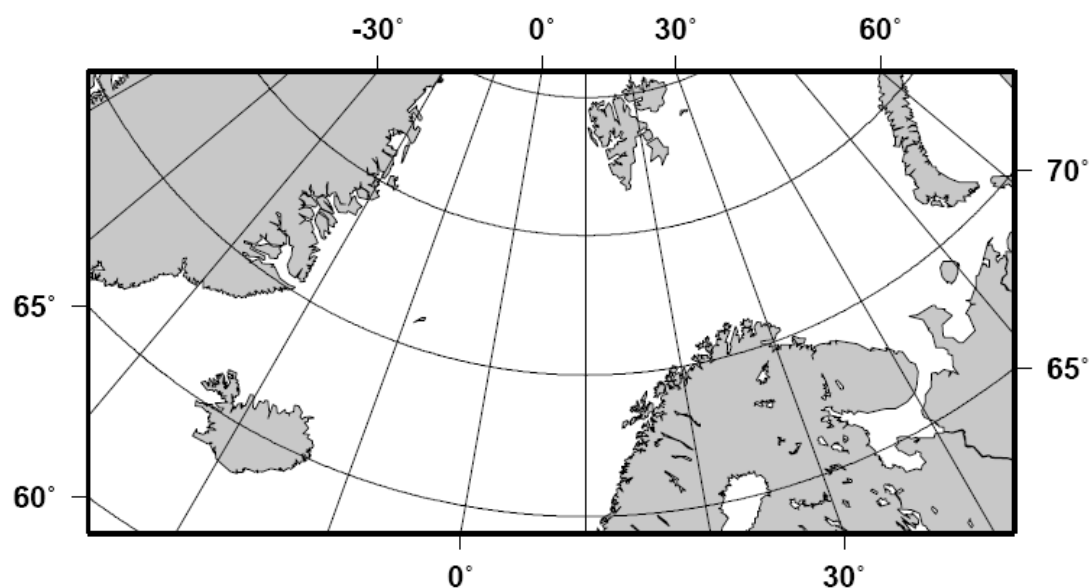


Figure 5.13: Polar stereographic conformal projection with rectangular borders.

4.3.3 Orthographic Projection (-Jg -JG)

这种投影是以一个无限远的视角来看地球，常常被用来表示从太空上看到的地球。它只可以显示一个半球。我们需要告诉 GMT：

投影中心的经纬度

投影的标尺 (radius/latitude or l:xxxxx or or map width)

```
#!/bin/sh
# $Id: GMT_orthographic.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

pscoast -R0/360/-90/90 -JG-75/40/4.51 -B15g15 -Dc -A5000 -G0 -P > GMT_orthographic.ps
```



Figure 5.15: Hemisphere map using the Orthographic projection.

4.3.4 方位角等距离投影(Azimuthal Equidistant Projection)(-Je -JE)

这个投影的一大特点是：从中心量的距离是真实的。因此，我们在图上以中心为原点画圆，这个圆上的点就真实的表示了地球上离圆心等距离的那些点。而且方向也是真实的。它经常被用来在全球尺度上表示一些点与特定点的位置关系。(不明白吗？有一个例子就是在研究面波时，要给处远震的分布，就可以利用这个投影，看看方位角的覆盖啊，距离啊，什么的)

```
#!/bin/sh
# $Id: GMT_az_eqidistant.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#
pscoast -R0/360/-90/90 -JE-100/40/4.5i -B15g15 -Dc -A10000 -G200 -W0.25p -P > GMT_az_eqidistant.ps
```

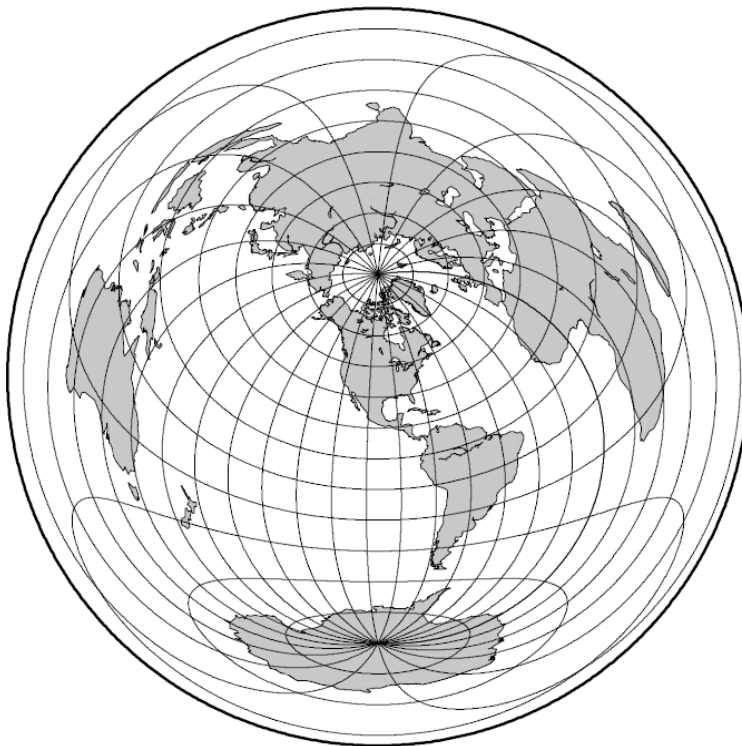


Figure 5.16: World map using the equidistant azimuthal projection.

4.4 圆柱投影

4.4.1 墨卡特投影(Mercator Projection -Jm -JM)

这种投影因他的发明者而得名，(1569)它可能是最著名的一种投影了。它是一种圆柱投影，正形的，在赤道上面精确投影。它在航海上有一个显著的优点，就是等角度的方向线在地图上是一条直线。如果我们想从一个地方航行到另外一个地方，只需要连线，确定方向，并且在整个航行中都保持这条路线。(要知道航海上定方向是很重要的，用这样的地图，我们可以在地图定一个方向，如 NNW30，然后用地图上的标示物指引我们)墨卡特投影被广泛的应用，但我们要记住这种

投影把高纬度的区域扭曲了，变大了。这就给我们一种错觉。例如，格陵兰岛比南美洲还大，实际上南美洲是格陵兰岛的八倍！还有前苏联看起来比非洲或者南美洲大。也许正是这种错觉影响了美国的对外政策！

普通的墨卡特投影需要我们输入：括号里的是默认值

中心子午线（地图的中心）

精确的纬线（赤道）

赤道上的标尺或地图的宽度，inch/degree or 1:xxxx or map width

```
#!/bin/sh
# $Id: GMT_mercator.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#

gmtset DEGREE FORMAT 1 BASEMAP_TYPE FANCY
pscoast -R0/360/-70/70 -Jm1.2e-21 -Ba60f30/a30f15 -Dc -A5000 -G0 -P > GMT_mercator.ps
```

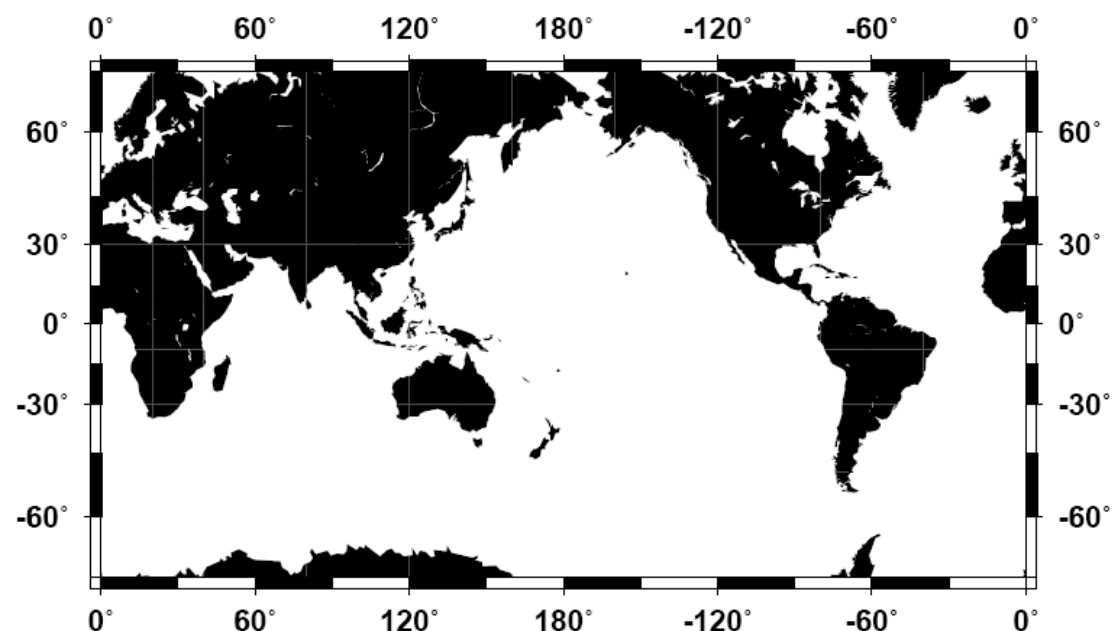


Figure 5.18: Simple Mercator map

4.4.2 横向墨卡特投影 Transverse Mercator (-Jt -JT)

在这种投影方式中，有一条子午线是没有扭曲（no distortion）的。中心的子午线，与之相距 90 度的子午线，赤道线是直线，其他的经纬线的投影都是复杂的曲线。用法如下：

中心子午线

纬线的起始

赤道上的标尺或地图的宽度

```
#!/bin/sh
# $Id: GMT_transverse_merc.sh,v 1.1 2001/03/21 04:10:21 pwessel Exp $
#
pscoast -R20/30/50/45r -Jt35/0.181 -B10g5 -D1 -A250 -G200 -W0.25p -P > GMT_transverse_merc.ps
```

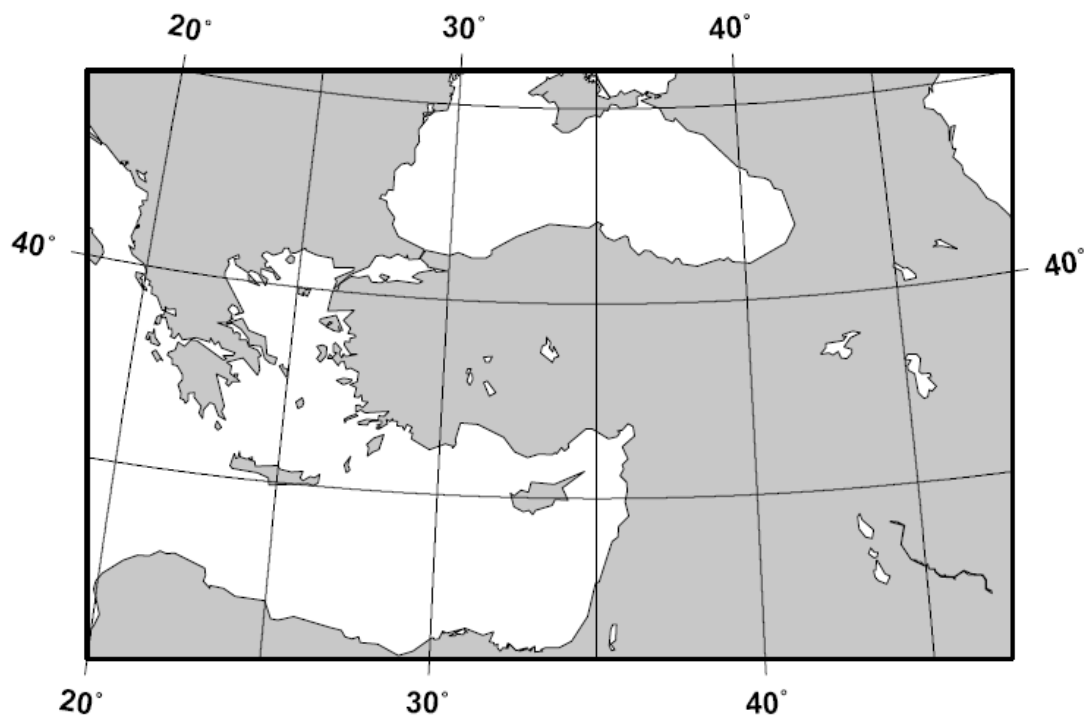


Figure 5.19: Rectangular Transverse Mercator map

后面讲的投影方式还有很多，笔者想列表总结一下，主要是列出他们的名称，特点和用法。例子，具体的细节就不讲了。有兴趣的读者可以自己查阅英文版手册。

名称	特点	用法
Universal transverse Mercator UTM (-Ju -JU)	被美国军方采用，全球被分成了 60 个区域，每个区域都有一个中心子午线。	UTM zone (1-60) 赤道上的标尺或地图的宽度
倾斜的墨卡特投影 (-Jo -J0)	特别用在投影斜的地形区域时	-Joa or -J0a 投影中心的经纬度 倾斜赤道的方位角 赤道上的标尺或地图的宽度
Cassini 圆柱投影 (-Jc -JC)	适合于南北跨度大的投影	投影中心的经纬度 标尺或地图的宽度
圆柱等距离投影 (-Jq)	经纬线全部是直线，	中心子午线

-JQ)	线性投影	标尺或地图的宽度
广义的圆柱投影 (-Jy -JY)	等面积, non-conformal 经纬线全部是直线	中心子午线 标准的平行线 标尺或地图的宽度
Miller Cylindrical Projection (-Jj -JJ)	经纬线全部是直线, 避免了奇异的极点	中心子午线 标准的平行线 标尺或地图的宽度

多姿多彩的全球投影 (Miscellaneous Projections)
 注意这些投影全都用到球形的近似, 而不是精确的椭圆公式。

名称	特点	用法
Hammer Projection (-Jh -JH)	边界是个椭圆	中心子午线 标尺或地图的宽度
Mollweide Projection (-Jw -JW)	这个投影主要被用来在 全球范围内显示数据分布的	中心子午线 标尺或地图的宽度
Winkel Tripel Projection (-Jr -JR)		中心子午线 标尺或地图的宽度
Robinson Projection (-Jn -JN)		中心子午线 标尺或地图的宽度
Eckert Projection (-JK , -Jk)		中心子午线 标尺或地图的宽度
Sinusoidal Projection (-Ji -JI)		中心子午线 标尺或地图的宽度
Van der Grinten Projection (-Jv -JV)	可以把整个地球投在一个圆内	中心子午线 标尺或地图的宽度

5 食谱 Cook-book

用食谱也许是学习 GMT 最好的方式了。本章中, 就给出了十余个具体的例子, 涉及到了 GMT 的主要功能。它分为四个部分:

1. 介绍我们要做什么事情
2. 给出 shell 脚本
3. 解释一些命令行
4. 给出绘图结果

这里面没有给出所有命令行的详细解释, 读者可以自己查阅手册中的语法说明。shell 脚本有 bash 和 csh。我们这里的例子全用的是 csh。所有的程序都是没有

屏幕反馈的，也就是没有报告中间进程的输出。而且，他会清除用过的多余的数据文件。还要注意一点：程序有时用到 `awk` 命令，因为不清楚是 `nawk` (另一种版本的 `awk`) 还是 `awk` 就用 `$AWK` 表示了，真正用到时，用户自己改一下就可以了。作为翻译，我想最好的方式是把每一个命令行和选项的意思解释明白，这样初学者可以很快的明白。可是，这样工作量有点大。我只想把我选的书中的五个例子详细的翻译一下。其余的例子请读者自己学习英文版。其实，大多数命令都会重复，所以五个例子就应该使初学者领悟了。(尤其是中国人，呵呵。)

我选的例子是 6.1 6.2 6.7 6.9 6.14

6.1 画等值线图

我们想用 Hammer 等面积投影作两张低精度的 `geoid` 等值线图。我们的输入数据是网格文件：`osu91alf_16.grd` 它包含了 1 度*1 度的全球的 `geoid` 等值线图。(至于这个文件是怎么得到的我们通过后面的例子讲)我们让一张图的中心线是格林威治线 (本初子午线)；另一张图的是国际日期变更线。正的等值线我们用实线，负的用虚线。每五十米间隔的等值线要有一个数值注释。图中大陆的颜色是浅灰的，最后呢，我们需要一个边框。

下面的中文解释是按照脚本的顺序一行一行地解释，选项也是按前后顺序一个个地解释：(有时候重复的还是会省略。)

```
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
pscoast -R-180/180/-90/90 -JH0/61 -X0.25i -Y0.5i -O -K -Bg30 -Dc -G200 >> example_01.ps
grdcontour -R osu91alf_16.grd -JH -C10 -A50f7 -G4i -L-1000/-1 -Wc0.25pta -Wa0.75pt2_2:0 -O -K \
-T0.1i/0.02i >> example_01.ps
grdcontour -R osu91alf_16.grd -JH -C10 -A50f7 -G4i -L-1/1000 -O -K -T0.1i/0.02i >> example_01.ps
pscoast -R0/360/-90/90 -JH180/61 -Y4i -O -K -Bg30:."Low Order Geoid": -Dc -G200 >> example_01.ps
grdcontour osu91alf_16.grd -JH -C10 -A50f7 -G4i -L-1000/-1 -Wc0.25pta -Wa0.75pt2_2:0 -O -K \
-T0.1i/0.02i:--+ >> example_01.ps
grdcontour osu91alf_16.grd -JH -C10 -A50f7 -G4i -L-1/1000 -O -T0.1i/0.02i:--+ >> example_01.ps
\rm -f .gmtcommands
```

置网格交叉点的大小为 0，注释字体大小为 10。

psbasemap 画一张长方形的边框：区域范围是 X 轴从 0 到 6.5，Y 轴从 0 到 9；投影方式是直角坐标投影，一个单位长度是一英寸；设置边框 (轴) 上的一些细节 (轴上的标记位置的短竖线，数值注释等等)，这里是设置他们全为 0，即消失；设置肖像画方式 (纵向)；设置后面还会追加命令；设置一个标题，和一个时间戳；强制输出到 `example_01.ps`

pscoast 画海岸线：区域范围是从西经 180 度到东经 180 度，从南纬 90 度到北纬 90 度；投影方式是 Hammer 等面积投影，中心子午线是 0 度，宽 6 英寸；图的坐标原点在 x 轴 0.25 英寸；y 轴 0.5 英寸；覆盖在上一个命令上输出；设置后面还会追加命令；经纬圈每隔 30 度画一条；分辨率为粗糙；颜色为 200 (255 黑 0 白)；追加输出到 `example_01.ps`

grdcontour 画等值线：区域范围与上一个命令一样；输入数据是网格文件：`osu91alf_16.grd`；投影方式不变；设置等值线间隔为 10；设置每五十米间隔的

等值线要有一个数值注释，注释字体大小为 7；同一条等值线上注释间隔为 4 英寸；设置画等值线的一个范围，只画-1000 到-1 的等值线；设置一般等值线属性，0.25 的粗细，ta 表示画虚线；设置有注释的等值线属性 0.75 的粗细，t2_2:0 表示另一种类的虚线；-T 表示在最低的等值线处画短竖线，指向下降的方向。后面的-T 选项还有在局域的高值或低值处加画+，-。

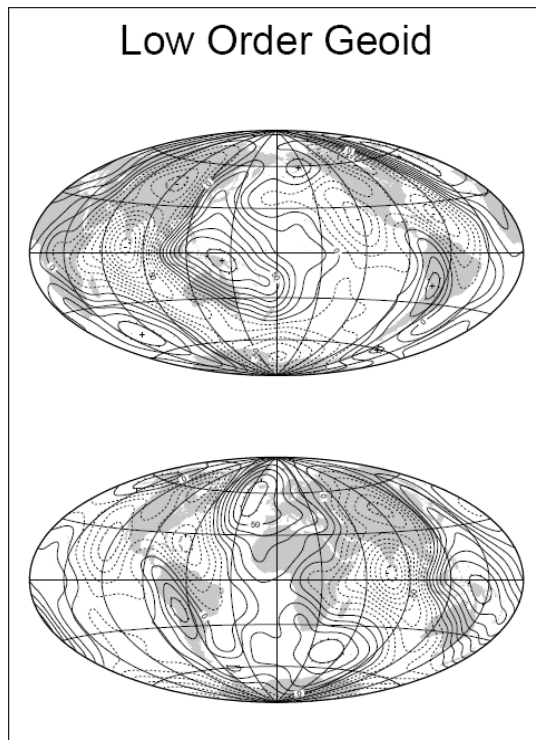


Figure 6.1: Contour maps of gridded data

6.2 画一个夏威夷的彩色地形图和 geoid

第二个例子我们将告诉你如何用网格文件画彩图。我们将用到补充包里的工具 `grdraster` 提取二维的海底地形的网格文件和 `geoid` 的高度，然后把他们放到一张图里面。我们感兴趣的是夏威夷岛。因为岛屿是倾斜走向的，我们用倾斜的墨卡特投影，并用热点的极点定义投影的极点（68W, 69N），我们选（190，25.5）为投影的中心，以（160, 20）（220, 30）定义长方形区域的角。`grdimage` 命令可以绘制彩图。

```

gmtset HEADER_FONT_SIZE 30 OBLIQUE_ANNOTATION 0 DEGREE_FORMAT 0
makecpt -Crainbow -T-2/14/2 >! g.cpt
grdimage HI_geoid2.grd -R160/20/220/30r -JOC190/25.5/292/69/4.5i -E50 -K -P -B10 -Cg.cpt \
-U/-1.25i/-1i/"Example 2 in Cookbook" -X1.5i -Y1.25i >! example_02.ps
psscale -Cg.cpt -D5.1i/1.35i/2.88i/0.4i -O -K -L -B2:GEOID:/m: -E >> example_02.ps
grd2cpt HI_topo2.grd -Crelief -Z >! t.cpt
grdgradient HI_topo2.grd -A0 -Nt -GHI_topo2_int.grd
grdimage HI_topo2.grd -IHI_topo2_int.grd -R -JO -E50 -B10:."H@#awaiian@# T@#opo and @#G@#eoid:" -O -K \
-Ct.cpt -Y4.5i >> example_02.ps
psscale -Ct.cpt -D5.1i/1.35i/2.88i/0.4i -O -K -I0.3 -B2:TOPO:/km: >> example_02.ps
cat << EOF | pstext -R0/8.5/0/11 -Jx1i -O -N -Y-4.5i >> example_02.ps
-0.4 7.5 30 0.0 1 2 a)
-0.4 3.0 30 0.0 1 2 b)
EOF
\rm -f .gmtcommands HI_topo2_int.grd ?.cpt

```

makecpt 制作调色板文件：彩虹模式，颜色对应的值从-2 到 14，数值对应着离散的颜色，间隔为 2； 强制输出到 g.cpt 文件。

grdimage 画彩图：输入文件 HI_geoid2.grd；以(160, 20) (220, 30)定义长方形的角；设置投影为倾斜的墨卡特投影，(190, 25.5)为投影的中心，极点(68W, 69N)，图的宽度是 4.5 英寸；设分辨率是 50dpi，-B 设置边框上每隔 10 度有一个标记，设置调色板文件为 g.cpt 文件。

psscale 画颜色标尺：设置调色板文件为 g.cpt 文件；设置标尺的位置为 (5.1i, 1.35i)，长 1.35i，宽 0.4i，-L 设置颜色标尺中每一个颜色块体等大小，-B 设置数值标记的间隔，文字，-E 表示有一个小三角在颜色标尺的上下两端（真是无聊的选项）

以上画的是下面的一个 geoid 的彩图，下面是地形图

grd2cpt 从网格文件中获取调色板文件：输入文件 HI_geoid2.grd，设置浮雕模式，设置为连续变化的调色板

grdgradient 制作梯度信息的网格文件（得到某一个方向的方向导数）：输入文件 HI_geoid2.grd，设置方向导数的方向为 0 度，（北方），-N 表示归一化，后面的 t 表示一种归一化算法，输出的文件为 HI_topo2_int.grd

grdimage 画彩图：输入文件 HI_topo2.grd，梯度信息的文件是 HI_topo2_int.grd(其实就是增加了立体的效果，好像有一束光从北方射过来，地形的不同造成了不同的效果)。

Cat << EOF | pstext……在某一个位置画标记 a) b) cat 命令是一个 shell 命令，这个命令大概表示：在坐标位置 (-0.4, 7.5)，画大小为 30，倾角为 0.0，字体号为 1 的符号：“a)”，最后一个 2 表示字体在该坐标的细致位置。

HAWAIIAN TOPO AND GEOID

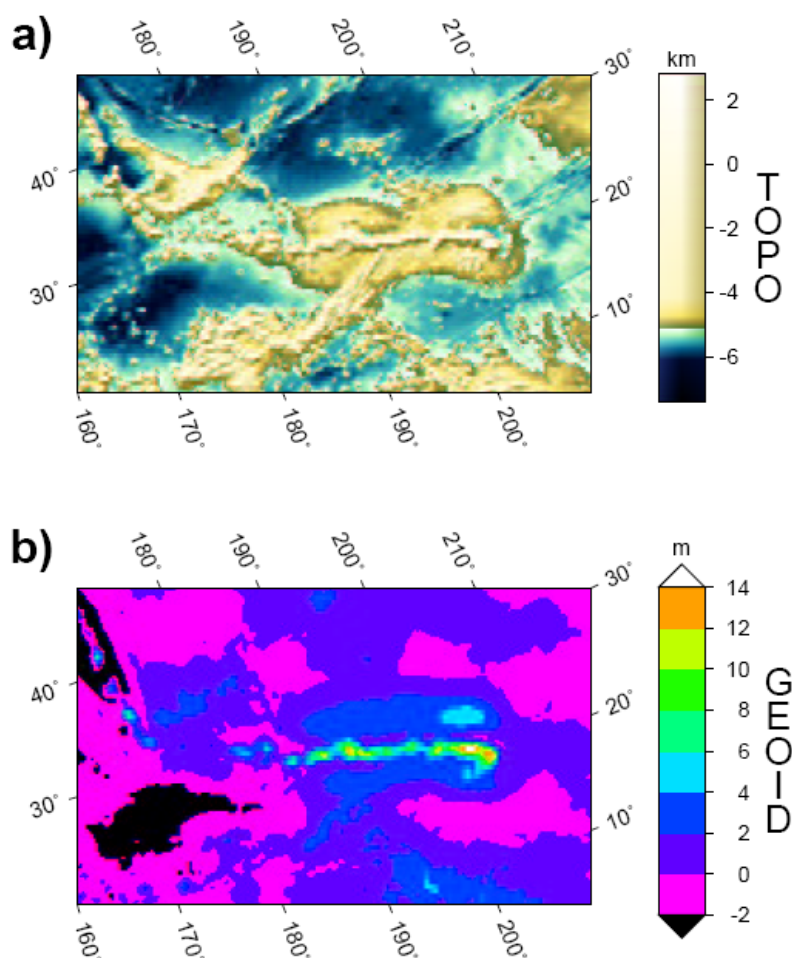


Figure 6.2: Color images from gridded data

6.7 一个简单的地震定位图

许多论文都会对位置感兴趣，如显示区域的某些特征和标记。这个例子显示了赤道附近的大西洋，不仅划出了大洋中脊和 fracture zone, 还划出了地震位置和 isochron。我们已经有有了一个包含地震位置和震级大小的文件：quakes.xy。我们把震级大小除以 100，用来表示地震图标的大小。Fracture zone 的轨迹是 fz.xy。

```
pscoast -R-50/0/-10/20 -JM91 -K -GP300/26 -D1 -W0.25p -B10 -U"Example 7 in Cookbook" >! example_07.ps
psxy -R -JM -O -K -M fz.xy -W0.5pta >> example_07.ps
$AWK '{print $1-360.0, $2, $3*0.01}' quakes.xy | psxy -R -JM -O -K -H1 -Sc1 -G255 -W0.25p \
>> example_07.ps
psxy -R -JM -O -K -M isochron.xy -W0.75p >> example_07.ps
psxy -R -JM -O -K -M ridge.xy -W1.75p >> example_07.ps
psxy -R -JM -O -K -G255 -W1p -A << END >> example_07.ps
-14.5 15.2
-2 15.2
-2 17.8
-14.5 17.8
END
psxy -R -JM -O -K -G255 -W0.5p -A << END >> example_07.ps
```

```

-14.35    15.35
-2.15     15.35
-2.15     17.65
-14.35    17.65
END
echo "-13.5 16.5" | psxy -R -JM -O -K -Sc0.081 -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
pstext -R -JM -O -S0.75p -G255 << END >> example_07.ps
-43 -5 30 0 1 6 SOUTH
-43 -8 30 0 1 6 AMERICA
-7 11 30 0 1 6 AFRICA
END
\rm -f .gmtcommands

```

画坐标点：-M 设置画的线是一段一段的 输入文件是 fz.xy，查看该文件可以发现用来标记分段的符号。默认的应该是>。

Awk 命令是把 quakes.xym 作了一个处理，第一字段（经度）减 360，第二字段（纬度）不变，第三字段（震级）除以 100，然后送给 psxy 处理，-H1 表示有一行头文件，-S 标示只画点，不连线 c 标示画的是圆圈，i 是单位英寸。

后面的命令基本上都是重复，最后的几个命令是在画文字说明和边框：south America Africa isc earthquakes

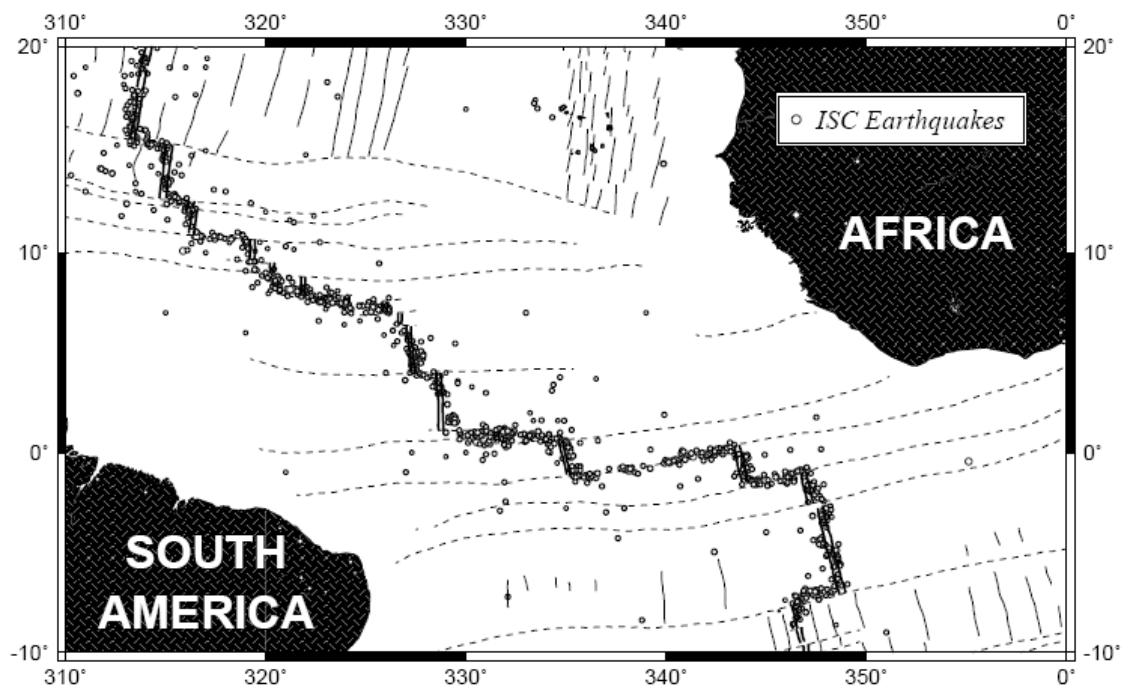


Figure 6.7: A typical location map

6.9 画时间序列 (plotting time-series along tracks)

许多海洋地球物理学家用它来画地磁条带，地震学家则用它在地图上画地震波形。在这个例子中，我们用 pswiggle 工具画出了南太平洋的 geosat sea surface slope profiles 的波浪线。这些数据都存储在 *.xys, fz.xy, ridge.xy 文件中。我们的每一个剖面都有一个 id，我们想要把它提取出来，用 pstext 画出来。我

们还要把 id 的方向也转到与剖面一致的方向。(见图) N40E。我们剪切出*.xys 的最后一列，把它粘贴到文件 tracks.lis 文件中，再用 awk 命令转化到适合 pstext 的格式。注意：pstext 里的选项-D 是让 id 号码离剖面有一个短的距离。

```

pswiggle track_*.xys -R185/250/-68/-42 -U"Example 9 in Cookbook" -K -Jm0.13i -Ba10f5 -G0 -Z2000 \
-W0.25p -S240/-67/500/@~m@~rad >! example_09.ps
psxy -R -Jm -O -K ridge.xy -W1.25p >> example_09.ps
psxy -R -Jm -O -K -M fz.xy -W0.5pta >> example_09.ps
if (-e tmp) then
  \rm -f tmp
endif
foreach file (track_*.xys)      # Make label file
  tail -1 $file >>! tmp
end
ls track_*.xys | $AWK -F. '{print $2}' >! tracks.lis

paste tmp tracks.lis | $AWK '{print $1, $2, 10, 50, 1, 7, $4}' | pstext -R -Jm -D-0.05i/-0.05i -O \
>> example_09.ps
\rm -f tmp tracks.lis .gmtcommands

```

pswiggle 画波浪线：文件名，区域，投影方式，边框上每隔 10 度有个数字，每隔 5 度分开；-Z 给出一个画波浪线的比例，-S 画一个比例尺。

foreach 循环读入 track_*.xys 的最后一列。

后面的命令又涉及到 shell 命令。大概意思就是把 id，和剖面的经纬度合成到一个文件里，这样的话就可以对应的画出 id 了。

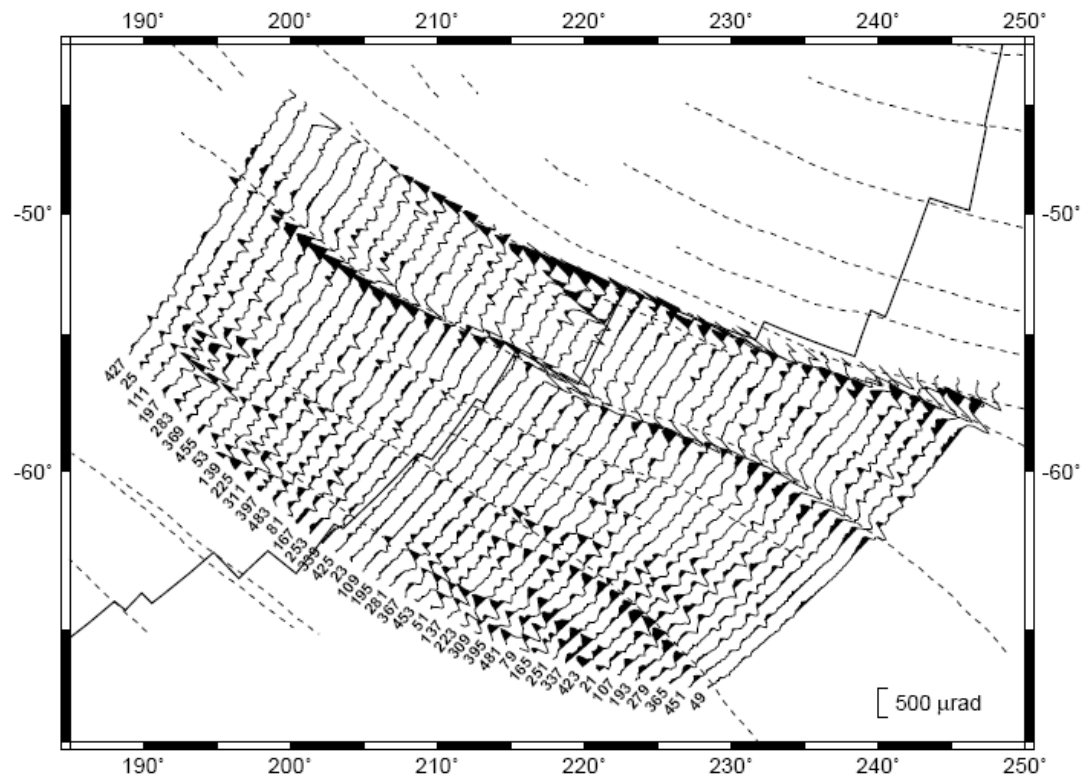


Figure 6.9: Time-series as “wiggles” along a track

6. 14 数据的网格化和画一个趋势面 (trend surface)

前面用到了这么多的网格文件，下面就来告诉你这些文件是怎么得来的。注意方法有很多，我们只讲一种方法。这个例子向我们展示了如何把随机分布的数据变成均匀采样的数据。首先，我们画出原始数据；我们选择的是等距离的网格，用 blockmean 避免重复；虚线网格是我们用 shell 脚本得到的；用 surface 我们把数据网格化，每隔 25 有一条等值线。重要的一点是：blockmean, blockmedian, blockmode 命令一定要在 surface 前运行，而且这两步一定要用同样的网格间隔。我们用 grdtrend 拟和了一组双三次曲线。Grdtrack 从这里面截了一条对角线。比较了双三次曲线和用 surface 得到的曲线。

```
gmtset GRID PEN 0.25pta
psxy table_5.11 -R0/7/0/7 -JX3.061/3.151 -B2f1WSne -Sc0.051 -G0 -P -K -Y6.451 >! example_14.ps

$AWK '{printf "%g %s 6 0 0 5 %g\n", $1+0.08, $2, $3}' table_5.11 | pstext -R -JX -O -K -N \
>> example_14.ps
blockmean table_5.11 -R0/7/0/7 -I1 >! mean.xyz
psbasemap -R0.5/7.5/0.5/7.5 -JX -O -K -B0g1 -X3.251 >> example_14.ps
psxy -R0/7/0/7 -JX -B2f1eSNw mean.xyz -Sc0.051 -G0 -O -K >> example_14.ps
$AWK '{printf "%g %s 6 0 0 5 %g\n", $1+0.1, $2, $3}' mean.xyz | pstext -R -JX -O -K -W255o \
-C0.011/0.011 -N >> example_14.ps
surface mean.xyz -R -I1 -Gdata.grd
grdcontour data.grd -JX -B2f1WSne -C25 -A50 -G31/10 -S4 -O -K -X-3.251 -Y-3.551 >> example_14.ps
psxy -R -JX mean.xyz -Sc0.051 -G0 -O -K >> example_14.ps
grdtrend data.grd -N10 -Ttrend.grd
grdcontour trend.grd -JX -B2f1WSne -C25 -A50 -G31/10 -S4 -O -K -X3.251 >> example_14.ps
project -C0/0 -E7/7 -G0.1 -Fxy >! track
psxy -R -JX track -Wlpto -O -K >> example_14.ps
grdtrack track -Gdata.grd | cut -f3,4 >! data.d
grdtrack track -Gtrend.grd | cut -f3,4 >! trend.d
psxy 'minmax data.d trend.d -I0.5/25' -JX6.31/1.41 data.d -Wlp -O -K -X-3.251 -Y-1.91 -B1/50WSne \
>> example_14.ps
psxy -R -JX trend.d -W0.5pta -O -U"Example 14 in Cookbook" >> example_14.ps
\rm mean.xyz track *.grd *.d .gmt*
```

Blockmean 是一个 surface 前的预处理命令，他读入空间上随机分布的数据，然后根据你定义的区域和网格，保证每个网格输出一个点，这个点是平均后的结果。输入文件是 table_5.11，区域是 0 到 7，0 到 7，网格间距是 1，输出文件是：mean.xyz

Surface 输入文件是：mean.xyz；区域不变；网格间距是 1；输出文件是 data.grd

Grdtrend：输入文件是 data.grd，-N10 指定了拟和函数的参数个数，数字越高，次数也就越高。对三元函数来说，10 对应着 10 个参数，即三次多项式，4 对应着一次多项式，7 对应着二次多项式。输出文件是：trend.grd。

Project 投影：等间隔的取一条对角线，起点是 0,0；终点是 7,7。点点间隔是 0.1。输出文件是 track。

grdtrack 沿着某一条线对一个二维的网格数据取样。取样的轨迹是 track，网格数据是 data.grd，再把输出的文件剪切到 data.d 里面。

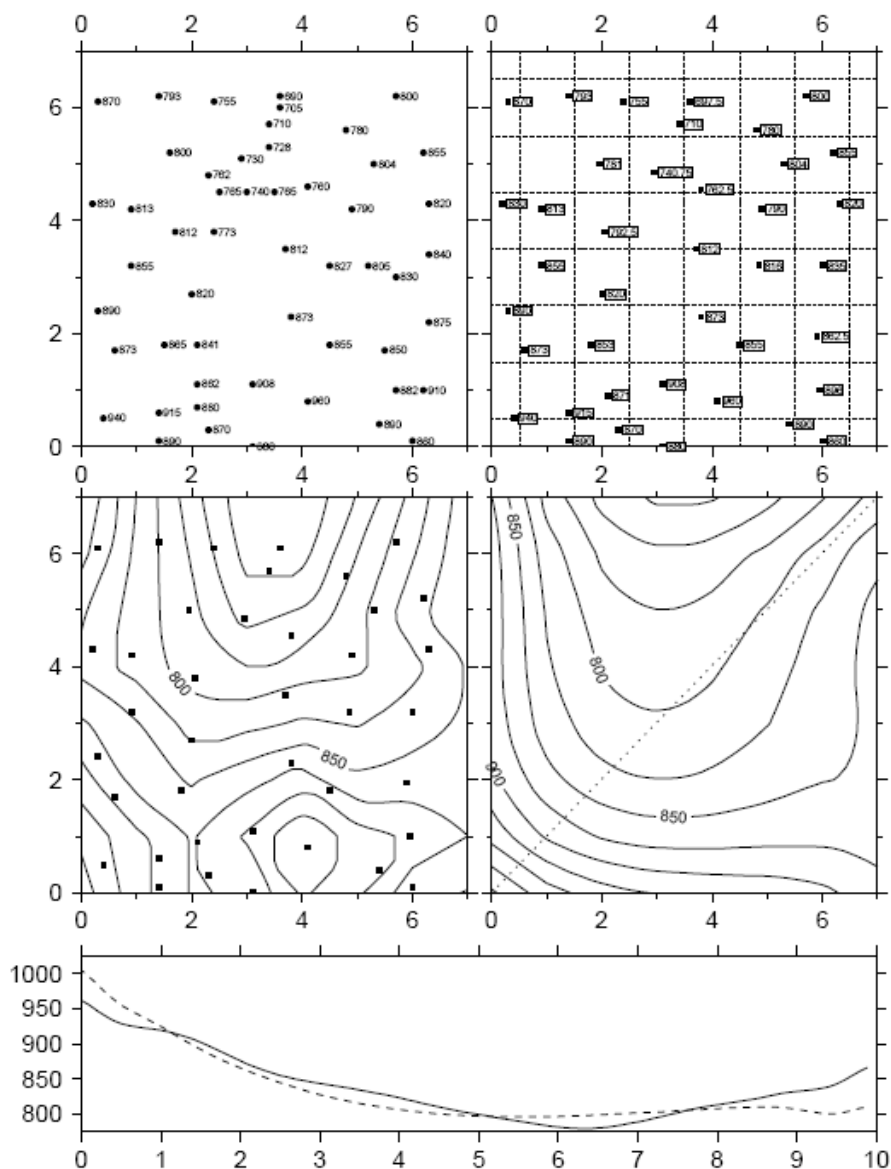


Figure 6.14: Gridding of data and trend surfaces

6 附录

A. GMT 补充包

这个包裹里提供了一些为特定应用设计的小程序。读者可以从网站下载到。GMT_suppl.tar.gz. 作者不保证所有的程序都可以在你的机器上运行，但是大多数情况是可以的。还有一点，程序中用到的数据不包含在这个包裹里，所以具体地应用时，可能还要去找具体的数据库。

A.1 dbase: gridded data extractor

这里面有程序 `grdraster`, 可以让你从全球的网格数据库里面提取数据 (如 NGDC)。我们在第六章的例子中用到了部分的网格。你也可以根据需要定制自己的数据。

A.2 gshhs: GSHHS data extractor

这里面有程序 `gshhs`, 可以让你从 Global Self-Consistent Hierarchical High-resolution Shorelines (GSHHS) 提取海岸线数据。还有 `gshhs_dp`, 可以智能地 decimate (愿意是十中取一, 就是再次挑选的意思吧), `gshhstograss` 将海岸线的片断数据 (segments) 转化到 GRASS 数据库格式。

A.3 imgsrc: gridded altimetry extractor

这里面有 `img2mercgrd` 程序, 可以从卫星测量中得到全球或一部分的重力数据或者预测的地形起伏的数据。

A.4 meca: seismology and geodesy symbols

这里面有 `pscoupe`, `psmeca`, `pspolar`, `psvelo`, 可以画震源球, 误差椭圆, 速度矢量。

A.5 mex: Matlab-GMT interface

这里面有 `grdinfo`, `grdread`, `grdwrite` 我们可以在 `matlab` 中用这些命令读写 `grdfiles` (网格文件)。

A.6 mgg: MGD77 extractor and plotting tools

这个软件包有程序: `binlegs`, `dat2gmt`, `gmt2dat`, `gmtinfo`, `gmtlegs`, `gmtlist`, `gmtpath`, `gmttrack`, `mgd77togmt`. 他们全都与一个海洋地球物理的数据库有关 (MGD-77)。可以维护, 提取, 绘图等等。

A.7 misc: posters, patterns, and digitizing

这里的程序都与制作海报 (poster) 有关。可以制作海报 (`psmegaplot`), 绘制图案 (`makepattern`), 建立表格 (`gmtdigitize`)。

A.8 segyprogs: Plotting SEG Y seismic data

这个软件包可以绘制 SEG Y 格式的地震波数据。Pssegy 建立二维的图像, pssegyz 画三维的图像。

A.9 spotter: backtracking and hotspotting

这个软件包与板块构造和热点有关。Backtracker 可以让你按时间顺序标记地质记号。Hotspotter 可以根据海底山脉的位置和板块运动的极点建立 CVA 网格。Originator 可以分析海底山脉, 找到热点的可能起源。

A.10 x2sys: Track crossover error estimation

这个软件包有 x2sys_datalist, 可以让你从二进制或 ASCII 数据文件中提取资料。X2sys_cross 帮你计算出几条轨迹的交叉点位置和该计算的误差。这个软件包会逐步取代了旧的 x_system 软件包。

A.11 x_system: Track crossover error estimation

这里有 x_edit, x_init, x_over, ……在 x2sys 完整前, 他们起到了分析轨迹交叉点和误差的作用。

A.12 xgrid: visual editor for grdfiles

在这里, 我们可以用 xgridedit 来可视化地编辑网格文件 (grdfiles)。

B. GMT 的文件格式

B.1 表格数据 table data

一般, 该文件有 N 个记录 (就是 N 行), 每行有 M 个字段 (列)。一般的工具都识别多列的文件。只是一般它们就只用到前两列或前三列。(filter1d, sample1d 是个例外) GMT 可以读取 ASCII, 和二进制的表格数据。

B.1.1 ASCII 表格数据

在这种数据前, 可能有一行或几行头文件。所以有时我们要用 -H 选项指示有头文件, 再改变在 .gmtdefaults 中的参数 N_HEADER_RECS, 为头文件的行数。字段可以用空格, TAB 键, 或者逗号分割。字段里可以是整数, 浮点数, 或者地理坐标 [+|-]dd[:mm][:ss][W|S|N|E|w|s|n|e]。有时我们要把一个数据分成几段,

存到不同的文件中，分别画出来。例如画几个断层线时，我们就可以这么做。但是如果断层线很多，文件就会很多，很零碎。我们就可以用选项-M 来解决。这是的文件可以称为 multiple segment file. 在文件中，默认的分割号是 \)。当然，你也可以用-Mstring 来改变这个。例如：-MH 是用 H 作分割号；-M '*' 是用*作分割号。如果断层线的粗细，颜色不一样，我们可以在分割号后面加-W, -G 选项来改变这些属性。

B.1.2 二进制的表格数据

为了加快数据处理的速度，我们用二进制的表格数据。头文件不可用。这时数据要么是单精度，要么是双精度。分隔数据的选项还是可以用的。这时我们要用 -bi[s][n] -bo[s] 来指定输入和输出。s 表示单精度，n 表示字段的个数。

B.2 二维的网格文件

B.2.1 文件内容

网格文件也有一个头文件，里面记录了这个网格文件的属性信息。如表中所示。grdedit 也可以编辑其中的某些属性。

B.2.2 两种节点的排列方式 (Grid line and Pixel registration) (或者称之为布局)

网格文件中数据存储在节点上。我们可以想象有一个 X, Y 坐标构成的表格。y 从下到上增加，x 从左到右增加。但是我们可以有两种不同的布局，如图所示。一般，GMT 工具可以读取这两种类型。对于某些工具，如 grdimage, pixel 的布局更有意义。Grdsample 和 grdproject 可以转化这两种类型的数据。

在 Grid line 布局下，数据分布在交叉点上；而在 pixel 布局下，数据分布在交叉点形成的网格上。注意 pixel 布局比 Grid line 布局的节点数 x 和 y 向各少一个。

B.2.3 网格的边界条件

当我们考虑一些命令：grdsample -L, grdgradient -L, grdgradient -L, grdtrack -L, nearneighbor -L, grdview -L, 我们要在边界插值或求导数时，我们就要注意选取什么样的边界条件。默认的边界条件是自然边界条件，保证插值的曲面有最小的曲率。如果我们的数据有周期性，那么我们可以具体的指出 x 或者 y 的周期性。注意这时候，pixel 布局 and Grid line 布局有一点差别。此外，还有地理的边界条件。感兴趣的读者可以详细地看英文版手册。

C. 制作GMT 的EPS 文件 (Encapsulated PostScript)

GMT 既可以输出 ps 格式文件，又可以输出 eps 格式文件。支持前者的媒介为打印机或者 ps viewer，后者的媒介为另外的文档，当然也可以用打印机或者 ps viewer 来处理。通过改变 gmtdefaults 中的 PAPER_MEDIA，我们可以选择生成的文件。

现在我们可以用 Adobe Illustrator, Aldus Freehand, Corel Draw 来编辑 EPS 文件, 进行剪切, 加入文字等等。

D. GMT 程序和相关资源的下载

所有的源程序, 数据, postscript文件, html 帮助手册, unix 版的帮助都可以从网上下载。 <http://gmt.soest.hawaii.edu>.

GMT_progs.tar 源程序, cpt 文件, ps 文件。

GMT_share.tar 中低分辨率的海岸线数据。

GMT_doc.tar 包括了与 GMT 相关的各种文档

GMT_web.tar html 格式的各种文档

GMT_full.tar 全分辨率的海岸线数据

GMT_high.tar 高分辨率的海岸线数据

GMT_scripts.tar 食谱 Cookbook 中的各种例子的 shell 脚本

GMT_suppl.tar GMT 补充包

Triangle.tar 一个 Delaunay 三角插值程序。

我们还有 windows 的版本:

GMT_exe.zip

GMT_suppl_exe.zip

通过 Unidata (<ftp://unidata.ucar.edu>) , 我们可以取得 netCDF 数据库 (netcdf.tar.Z)。它是与网格文件的输入输出相关的一个重要数据库。

K. 关于分辨率的一点说明

其实, 笔者没有留意这部分内容, 因为觉得用处不大。只想讲一点, 分辨率通过选项-D 控制。一般来讲, 研究的区域越大, 分辨率就可以调低一些: 如-Dl -Dc ; 研究的区域越小, 分辨率就需要越精细。

-Dc crude

-Dl low

-Di intermediate

-Dh high

-Df full

后记

终于大功告成，这个 GMT 的说明文档大概至少也用了 50 几个小时吧。由于水平有限，还是有一些不尽如人意的地方：关于投影技术的细节，本人没有能够很好地解释清楚；有一些英文实在不知道中文怎么说，只能意会，不能言传。这些问题以后也许可以解决，也许不行，看以后的情况了。

但是不管怎么说，毕竟是第一次，所以自己还是挺满意的。我的希望是：只要能使一个人受益，这件事就没有白做，呵呵。