
M_Map: 用户指南1.4版



1 入门

首先，获取并解压缩 [zip 压缩包](#) 或 [gzip 压缩的 tar 文件](#)，得到所有文件。如果你解压了 zip 文件，**请确保你还解压了子目录**！现在，启动 Matlab（版本号为 5.0 或者更高）并确保工具箱在你的路径中。通过 CD 命令可以很简单的到的正确的目录。

另外，如果你已经将他们解压到/users/rich/m_map (和 /users/rich/m_map/private)目录下，那么可以将这些加入到你的搜索路径中：

```
path(path, '/users/rich/m_map');
```

或者

```
addpath /users/rich/m_map
```

按照这份文档，你可能需要使用一个网页浏览器打开 file:/users/rich/m_map/map.html，也就是本 HTML 文档。（译者注：原版说明是网页格式的）

注：你可能需要需要将 m_map 安装成为对所有用户都可用的工具箱。为了实现这一点，需要将文件解压缩至 \$MATLAB/toolbox/m_map，将此目录添加到 \$MATLAB/toolbox/local/pathdef.m 中定义的列表并使用 rehash toolboxcache 更新缓存文件。

在[第9章](#)（译者注：实际上应为第 10 章，下同）给出了高分辨率水深数据库（可选）的安装说明，在[第10章](#)给出了高分辨率 GSHHS 的海岸线数据库（译者注：实际上应为第 11 章，下同）。当然，我们需要首先检查这些基本设置都是没问题的。

让我们来看一个地图示例，试试这个：

```
m_proj('oblique mercator');
m_coast;
m_grid;
```

这是俄勒冈州/不列颠哥伦比亚省海岸的一个线图，使用斜墨卡托投影（如果想生成一些更复杂的地图，可以运行演示功能 `m_demo`）。

第一行初始化投影。不同的投影已经被设置了默认值，因此你可以很轻松的看到一个特定的投影是什么样的。然而所有的投影都有一系列可选参数。如果要在不使用默认值的情况下得到相同的地图，你可以使用

```
m_proj('oblique mercator', 'longitudes', [-132 -125], 'latitudes', [56
40], 'direction', 'vertical ', 'aspect', .5);
```

这些选项的确切含义在[第2章](#)被给出。然而，需要注意经度用符号进行划分—东经为正，西经为负。还有一点需要注意的是角度用十进制表示，因此东经 120°30'W 被指定为-120.5。第二行绘制海岸线，使用 1/4 度的数据库。更高的分辨率的海岸线可以使用自己的数据库（见[第7章](#)（译者注：实际上应为第8章））。`m_coast` 可以调用不同的参数。例如，

```
m_coast('linewidth', 2, 'color', 'r');
```

这就画出了一个粗的红色海岸线。填充的海岸线也可以画出来，使用 `'patch'` 选项（后面跟着其通常的属性和值）

```
m_coast('patch', [.7 .7 .7], 'edgecolor', 'none');
```

这就绘制出灰色填充并且无边框绘制的海岸线。

第三行叠加网格。尽管有很多可用于自定义网格外观可选的选项，然而却总可以使用默认值（例如本例）。这些选项在[第4章](#)讨论。你可以使用 GET 语法得到一系列选项列表：

```
m_grid get
```

这有点类似常规绘图中的 `get(gca)` 语法。

最后，假设你想在 129°W, 48°30'N 显示和标注位置一个锚：

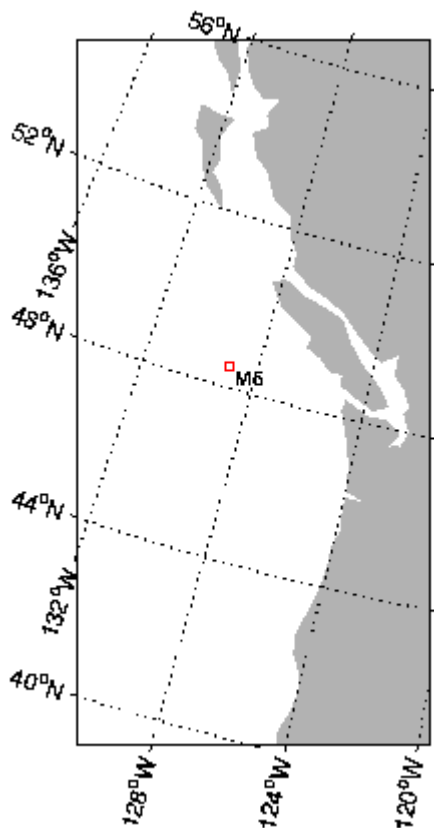
```
[X,Y]=m_ll2xy(-129,48.5);
line(X,Y, 'marker', 'square', 'markersize', 4, 'color', 'r');
text(X,Y, ' M5', 'vertical', 'top');
```

`m_ll2xy`（和与其相反的 `m_xy2ll`）用于转换经度/纬度坐标去匹配投影。在转换投影坐标的过程中也可以指定各种裁剪选项。如果你愿意接受默认的剪切设置，你可以使用内置的函数 `m_line` 和 `m_text`：

```
m_line(-129,48.5, 'marker', 'square', 'markersize', 4, 'color', 'r');
m_text(-129,48.5, ' M5', 'vertical', 'top');
```

最后 (!), 我们可能需要稍微改变网格的细节。需要注意的是, 一部给定的地图只能被初始化一次。

```
clf
m_coast('patch',[.7 .7 .7],'edgecolor','none');
m_grid('xlabelldir','end','fontsize',10);
m_line(-129,48.5,'marker','square','markersize',4,'color','r');
m_text(-129,48.5,' M5','vertical','top');
```



2 指定投影

可以使用 `m_proj get` 或 `m_proj('set')` 得到投影的列表。
一般会返回:

Available projections are:

- Stereographic
- Orthographic
- Azimuthal Equal-area
- Azimuthal Equidistant
- Gnomonic
- Satellite

Albers Equal-Area Conic
 Lambert Conformal Conic
 Mercator
 Miller Cylindrical
 Equidistant Cylindrical
 Oblique Mercator
 Transverse Mercator
 Sinusoidal
 Gall-Peters
 Hammer-Aitoff
 Mollweide
 Robinson
 UTM

(译者注：具体含义及翻译见后文)

如果你想得到这些投影可能的选项，可以在上述命令中添加其名称，例如：

```
m_proj('set','stereographic');
  'Stereographic'
<,'lon<itude>','center_long>
<,'lat<itude>','center_lat>
<,'rad<ius>','( degrees | [longitude latitude] )>
<,'rec<tbox>','( 'on' | 'off' )>
```

你也可以得到关于投影的更多细节。例如为了查看正弦投影的默认参数，我们出现需要初始化，然后使用'set'选项：

```
m_proj('sinusoidal');
m_proj get
Current mapping parameters -
Projection: Sinusoidal (function: mp_tmerc)
longitudes: -90 30 (centered at -30)
latitudes: -65 65
Rectangular border: off
```

(翻译：

当前映射参数-

投影：正弦（功能：mp_tmerc）的

经度：-90 30 （中心在-30）

纬度：-65 65

矩形边框：关闭)

为了初始化投影，你通常需要指定一些定义投影几何形状（经线限制，中心平行等）的位置参数，以及定义地图范围（是否在直角坐标系内，边界点是什么等）的参数。这些在不同的投影中略有不同。

投影有两个有用的属性，一个是能够在不同的小区域内保证角度不变，二是保证等积。符合第一个条件的投影叫做 *等角（保形）投影*，符合第二种条件的投影叫做 *等积投影*。没有同时符合两个条件的投影。许多投影（特别是全球投影）其实一个条件都不符合，而是用一种介于二者之间的美学平衡来代替。

注：大部分投影都是 *球形* 而非 *椭球形*。UTM（通用横轴墨卡托）是一种椭球形投影，如果需要的话，兰伯托的等角特性和阿尔伯斯的等积特性都可以在椭球上满足。这在你拥有加拿大各省或美国各州数据时会很有用。因为这些数据经常使用上述投影中的一种进行绘制。在正常的使用中这不见得是一个问题。

2.1 方位投影

方位投影是将地球上所有点都投影到一个平坦的切平面的投影。使用这种投影的地图能够将从中心点到其它所有点的方向或者方位角显示正确。穿越中心点的大圆路线显示为直线（尽管不穿过中心点的大圆可能会出现弯曲）。这些地图通常用圆形边界绘制。定义一个方位投影需要使用下列参数：

```
<,'lon<itude>',center_long>
<,'lat<itude>', center_lat>
```

这些参数定义了地图的中心点。由于地图的对称性，指定的经度在地图中央是垂直的，并且其北段在地图顶部（请看下面的 `rotangle` 选项）。

```
<,'rad<ius>', ( degrees | [longitude latitude] )>
```

这个定义了地图的范围。可以给出任何一个以度为单位的角度距离（比如 90 表示一个半球），或者指定一个边界上的点坐标。

```
<,'rec<tbody>', ( 'on' | 'off' | 'circle' )>
```

默认是在圆形边界中放入地图（使用后两种选项中的任意一个），但是也可以指定矩形的边界。然而，矩形地图在类似圆柱或者圆锥投影下通常会表现的更好。

```
<,'rot<angle>', degrees CCW>
```

这个可以用来旋转图形，使得中央经度不再是垂直的。

2.1.1 赤平投影（Stereographic）

赤平投影保形（即等角，下同）不等积，通常被用在极地区域。

2.1.2 正射投影（Orthographic）

正射投影既不保形也不等积，但是其类似于地球的透视图。

2.1.3 方位等积投影（Azimuthal Equal-Area）

有时也被称为兰伯托方位等积投影，该投影等积不保形。

2.1.4 天顶等距投影（等距方位投影）（Azimuthal Equidistant）

此投影既不等积也不保形，但是从中心点出发的距离和长度都是保真的。

2.1.5 球心投影（极平投影）（Gnomonic）

这也是一个既不等积也不保形的投影，但是地图上所有的直线（不只是那些穿过中心的直线）都是大圆路径。然而，由于地图边缘有很大的失真，因此最大半径应该被控制在最大-20 度或 30 度。

2.1.6 卫星投影（Satellite）

这是地球的透视图，仿佛在一颗在指定高度的卫星看到一样。该投影需要指定视角高度而非地图半径。

```
<,'alt<itude>', altitude_fraction >
```

此属性的数值表示视角的高度。该高度以地球半径为单位。例如一颗轨道半径为 3 地球半径的卫星的高度为 2。

2.2 圆柱投影和伪圆柱投影

将一个平面围绕着地球卷起来，只允许其接触到大圆。然后将点都投影到平面上，这就是圆柱投影。这种投影通常用作显示横向范围较大的区域或全球的中纬度地区。在这里还包括了两种伪圆柱投影，正弦投影和彼得斯投影，这两者与圆柱投影有一些类似（详见下文）。这些地图通常使用矩形边界绘制（正弦投影和横向麦卡托投影有时例外）。

2.2.1 麦卡托投影（Mercator）

这是一种基于与赤道相切的圆柱体的保形投影。这种投影上的直线是等方位线（例如方位固定运动的轨迹）。该投影受以下属性影响：

```
<,'lon<itude>', ( [min max] | center)>
```

可以设置任何经度限制。也可以设置中央经度，此时表示全球地图。

```
<,'lat<itude>', ( maxlat | [min max])>
```

纬度限制通常南北一致，指定一个值就可以。也可以在有需求的情况下指定不同的值。

2.2.2 米勒圆柱投影（Miller Cylindrical）

此投影既不等积也不保形，但是在展示世界地图的时候颜值很高。其属性与麦卡托投影一致。

2.2.3 等距圆柱投影（Equidistant cylindrical）

此投影既不等积也不保形。它采取等距的经纬线，经常用于快速绘制数据图。它在这里的意义仅仅是让这种地图可以使用网格生成程序。也被称作可利投影（Plate Carree），属性同上面的麦卡托一样的。

2.2.4 斜麦卡托投影 (Oblique Mercator)

当麦卡托投影切的大圆不是赤道而是倾斜的时候,就会产生斜麦卡托投影。这种投影在漫长的海岸线或者其它不规则形状或者直线型区域等情况下很有用。该投影受以下属性影响:

```
<,'lon<gitude>',[ G1 G2 ]>
<,'lat<itude>',[ L1 L2 ]>
```

两点指定了一个大圆,也限制了地图(假设最短的两段弧之间的区域是必须的)。因此(G1,L1)和(G2,L2)两点在地图的上下或者左右边界中间(由'direction'属性决定)。

```
<,'asp<ect>',value>
```

这决定了地图在所切大圆正交方向的尺寸,并以此作为长度显示的一个比例。纵横比为 1 产生的是正方形地图,更小的值会产生更纤瘦的地图。也可以指定纵横比大于 1,但是这个并不推荐。

```
<,'dir<ection>',( 'horizontal' | 'vertical' )
```

这决定了所切大圆在页面上是水平(此时的地图短粗胖)还是垂直(此时的地图高又瘦)。

2.2.5 横向麦卡托投影 (Transverse Mercator)

横向麦卡托投影是斜麦卡托投影在所切大圆为经线时候的特殊情况,因此是保形的。它经常用于大比例尺地图和图表。该投影受以下属性影响:

```
<,'lon<gitude>',[min max]>
<,'lat<itude>',[min max]>
```

这些限制了地图的范围。

```
<,'clo<ngitude>',value>
```

尽管它的最大意义是指定中央经线为所切的经线(默认即此),特定的地图分类系统(尤其是通用横向麦卡托投影)仅仅使用一组固定的中央经度,这些有可能不在地图中央。

```
<,'rec<tbox>',( 'on' | 'off' )>
```

地图的限制既可以基于经纬度(默认),也可以根据一个确切的矩形确定地图边界。在大比例尺地图中区别不大。注:尽管本投影与通用横向麦卡托投影(UTM)类似,后者实质上是椭圆形。

2.2.6 通用横向麦卡托投影 (Universal Transverse Mercator, UTM)

UTM 地区仅仅用于地球上小区域(几个经度)的高质量地图,是椭圆形投影。选项和横向麦卡托投影类似,此外还有:

```
<,'zon<e>','value 1-60>
```

```
<,'hem<isphere>','value 0=N,1=S>
```

如果这些没有指定是自动计算的。椭球默认是'normal'，也就是一个半径为 1 单位的球面地。但是其它的选项也可以选择如下的属性：

```
<,'ell<ipsoid>','ellipsoid>
```

获得一系列可用的椭球体可以尝试 `m_proj('set','utm')`。

UTM 和其它投影最大的不同在于：对于不是'normal'的椭球而言，投影坐标在东面和北面以米为单位。为了充分利用这一点，通常调用 `m_proj` 将'rectbox' 设置成 'on'而非使用 `m_grid` 生成经纬度网格会更有用（因为正常的 matlab 网格以米为单位）。

2.2.7 正弦投影（Sinusoidal）

此投影也经常成为“伪圆锥”，因为平行的纬线显示为直线，与它们在圆锥投影切于赤道的样子类似。然而，在本投影中的经线是以正弦的方式连续（因此得名），这使得地图是等积的。

2.2.8 彼得斯投影（Gall-Peters）

经纬线都是直线，但是纵向比例尺被扭曲，因此得以等积。这常用于热带地区，但是在极地失真严重。

2.3 圆锥投影

将一个圆锥罩在球体上投影即可得到圆锥投影。圆锥顶点在球体的转轴上。圆锥既可以切于一个纬圈，也可以在两个不同的纬度上切割球体。在纬向范围比较大的中纬度地区该投影很有用。下列属性会影响本类投影：

```
<,'lon<gitude>',[min max]>
```

```
<,'lat<itude>',[min max]>
```

这些确定地图的范围。

```
<,'clo<ngitude>','value>
```

在页面上中央经度是垂直的。默认值是平均经度，然而它可以被设置为任意值（甚至可以是范围之外的值）。

```
<,'par<allels>',[lat1 lat2]>
```

指定标准纬圈。可以给出一个或者两个，默认的是平均纬度。

```
<,'rec<tbox>','( 'on' | 'off' )>
```

地图的限制既可以基于经纬度（默认），也可以根据一个由给出范围构成的确切的矩形确定

地图边界。如果区域在地图上显示的不是很小，最好置为'off'。

在地图映射转换中默认使用球形地球模型。然而，也可以指定椭球坐标系。然而这仅仅在做坐标转换的时候很有用（比如说在使用一种投影的指定的网格数据库中寻找经纬度数据），因为在图中不可能用肉眼看到差异。特定的椭球可以使用以下属性来选择：

```
<,'ell<ipsoid>', ellipsoid>
```

可以试着使用 `m_proj('set','albers')` 来查看可用的椭球列表，结果如下所示：

```
normal
    sphere
    grs80
    grs67
    wgs84
    wgs72
    wgs66
    wgs60
    clrk66
    clrk80
    intl24
    intl67
```

2.3.1 阿尔伯斯等积圆锥投影（Albers Equal-Area Conic）


此投影等积不保形

2.3.2 兰伯托投影（兰伯托正形圆锥投影）（Lambert Conformal Conic）

此投影保形不等积（译者注：我国最常用的就是此类投影）

2.4 多角度全球投影

有很多投影真的不适合任何上述投影归类。这些大部分都是全球投影（也就是它们展示的是整个世界），并且被设计的颜值很高。我不知道他们一般有什么用，但是他们做出了好看的

标志！

2.4.1 哈默-埃托夫投影（Hammer-Aitoff）

一种经纬线都是弯曲的等积投影。

2.4.2 摩尔魏特投影（Mollweide）

也叫做椭圆投影或者等比例等积投影。此投影中纬线都是直的。注意到[例 4](#)展示了一个被设计成减少失真而且相当复杂的应用，生成了一个更加标准的地图。其中使用了：

```
m_proj('mollweide');
m_coast('patch','r');
```

```
m_grid('xaxislocation','middle');
```

2.4.3 罗宾逊投影 (Robinson)

不等积不保形，但是可能“颜值比较高”。

2.5 然而我该使用何种投影？

好吧，这取决于你要画多大的区域。通常，世界地图采用麦卡托投影。尽管米勒圆柱投影看起来更好看，但是在两极地区强调不够。另一种选择是哈默-埃托夫投影或者摩尔魏特投影（在极地附近经线弯曲到一点）。他们都是等积的。但是地图中间附近没有赤道的投影可能会不太好。罗宾逊投影既不等积也不保形，但是国家地理 (National Geographic) 却选择了它，而且 IPCC 的报告中也会出现它的身影。

如果你要绘制一些南北方向范围很大但是不是很宽的图形（例如南北美洲或者南北大西洋），正弦投影或者摩尔魏特投影看起来很不错。另外一个选择就是横向麦卡托投影，尽管它一般只用于很大比例尺的地图。

对于在一个半球内或者其它小一点的区域（例如澳大利亚、美国、地中海或者北大西洋），你可以选择圆锥投影。不同的圆锥投影之间差距很小。如果你对于投影不甚了解，无论你选择什么变化都不会很大。

如果你想绘制的区域更小，无论使用什么投影都无所谓。在很多情况下我感觉都很管用的就是斜麦卡托投影，因为你可以沿着一列长的（但是狭窄）的沿海区域排列它。如果地图的经纬度范围合适，使用横向麦卡托投影或者圆锥投影，UTM 投影也是可以的。

极地区域通常使用赤平投影，因为纬线构成了“靶心”的图案，所以它看起来棒棒哒。

如果你想快速的看到投影的样子，所有函数的默认参数都设为"typical"用法。也就是说想快速查看投影看起来是什么样的，你可以这么做而无需计算一大堆数值：

```
m_proj('stereographic'); % 赤平投影的例子
m_coast;
m_grid;
```

2.6 地图比例尺

M_Map 通常会对地图进行比例缩放以适应当前坐标轴。如果你想得到一张颜值高的图片（大部分情况应该都是这样），这就正是你所需要的。然而另一方面，有时你需要将一些东西打印成精确的比例（也就是说如果你真的喜欢坐在你的办公桌前，用一把尺子和一张纸试着测量曼谷和东京之间有多远。）使用 m_scale 的原始比例 1:250000，可以在将所有东西都画好之后调用：

```
m_scale(250000);
```

需要注意的是一个 1:250000 的世界地图比一张 8.5"x11"的纸不知道大到哪里去了。

这个选项通常只用来绘制大比例尺地图，也就是很小区域的地图，

如果你想知道当前的比例尺，可以不添加任何参数调用 m_scale，随后将进行计算并返回该值。

返回默认比例尺可以调用 m_scale('auto')

(PS—如果你想测量曼谷到世界各地的距离，绘制一个以曼谷 (13°44'N, 100°30'E) 为中心的等距方位投影世界地图，选取一个很小的比例尺，比如说 1:200,000,000。) 另一个选项可以被用来绘制等距圆环，参考[例 11](#)。

2.7 地图坐标系——地理坐标系和地磁坐标系

经纬度是地图中使用最多的坐标系。在一些情况下也有使用 UTM 坐标的，但是这些仅仅是一些基于赤道和特定经线的位置的简单变换。另一方面，也存在基于其它一些坐标轴系的坐标系适用的情况。例如在空间物理学中，数据都是依据地磁坐标系进行组织的。M_Map 对于基于其它坐标系统的数据处理能力有限。m_coord 允许将坐标系统从地理坐标系更改到地磁坐标系。下列代码可能会给你一些思路：

```
lat=[25*ones(1,100) 50*ones(1,100) 25];
lon=[-99:0 0:-1:-99 -99];

clf
subplot(121);
m_coord('IGRF2000-geomagnetic'); %将所有的经纬度改成地磁坐标系
m_proj('stereographic');
m_coast;
m_grid;
m_line(lon,lat,'color','r'); %在地磁坐标系上假定地磁经纬度
m_coord('geographic'); % 转换成假定地理坐标系
m_line(lon,lat,'color','c'); % 现在变成了地理坐标系

subplot(122);
m_coord('geographic'); % 将所有都定义成地理坐标系
m_proj('stereographic');
m_coast;
m_grid;
m_line(lon,lat,'color','c');
m_coord('IGRF2000-geomagnetic'); % 现在假定值都是地磁坐标系中的
m_line(lon,lat,'color','r');
```

注：此选项不常用，因此不是支持的不是很完美。特别指出的是，海岸线可能无法适应。

3 海岸线和水深

M_Map 包含两个简单的海岸线和全球高程数据库。在本版本中没有引入高细节的数据库，因为他们特别大而且处理时间很长（内置循环）。如果需要更细节的地图，[第 9 章](#)和[第 10 章](#)给出了如何添加一些免费的高分辨率数据集。如果想要添加自己的海岸线或者水深，请阅读[第 7 章](#)和[第 8 章](#)。

3.1 海岸线选项

M_Map 内置一个 1/4 度分辨率的海岸线数据库。对于地球上大部分地区是合适的，但是对于很多大尺度应用来讲很明显是粗糙的。对于区域地图不满意的用户可以去[第 7 章](#)和/或[第 10 章](#)得到关于创建和使用高分辨率海岸线的更多信息。使用 `m_coast` 函数可以访问 M_Map 数据库。海岸线可以通过下述语句以简单线条形式绘制出：

```
m_coast('line', ...optional line arguments ); %可选的线条参数
```

或者

```
m_coast( optional line arguments );
```

可选的参数都是标准参数，比如特定线条样式、线宽、颜色等。海岸线也可以使用颜色填充的方式绘制，使用：

```
m_coast('patch', ...optional patch arguments );%可选的填充参数
```

其中尾部的可选参数都是标准的填充属性，例如：

```
m_coast('patch',[.7 .7 .7],'edgecolor','g');
```

绘制灰色的陆面和绿色的外廓。填充被绘制之后，湖泊和内陆区域使用坐标轴背景色填充。很多旧的（海洋）地图使用斑点式的陆地边界，黑白看起来美美哒。你也可以使用下述语句调用 `m_hatch` 得到斑点边界：

```
m_coast('speckle', ....optional m_hatch arguments);%可选的m_hatch参数
```

因为在图中没有太多很小的岛屿和/或湖泊，因此这个看起来美美哒（参考[例 13](#)）。

需要注意的是线条式的海岸线通常画起来很快。填充式的海岸线需要更多时间去生成（因为地图的范围不一定是矩形的，剪裁过程必须使用 `m` 文件完成）。

3.2 地形/水深选项

M_Map 可以访问 1 度分辨率的全球高程数据库（实际上，这个数据库是在 Matlab 中的，可以通过 `$MATLAB/toolbox/matlab/demos/earthmap.m` 使用）。采用默认层次的高程等高线图可以使用下面的语句进行绘制：

```
m_elev;
```

也可以指定不同的层次

```
m_elev('contour',LEVELS, optional contour arguments);
```

比如你想所有的等值线都是深蓝色，可以使用：

```
m_elev('contour',LEVELS,'edgecolor','b');
```

也可以绘制填充等高线图：

```
m_elev('contourf',LEVELS, optional contourf arguments);
```

最后，如果你只是想简单的提取高程数据自用：

```
[Z, LONG, LAT]=m_elev([LONG_MIN LONG_MAX LAT_MIN LAT_MAX]);
```

这将返回在指定经纬度范围内的深度 Z 矩阵。

4 自定义坐标轴

4.1 网格线和标签

为了得到完美的网格，你可能会想尝试不同的网格选项。这里有两个有用的函数，用来绘制网格的 `M_GRID` 和清除当前网格(但是会留下所有的海岸线和用户特定信息)的 `M_UNGRID`。尝试以下语句：

```
m_proj('Lambert');
m_coast;
m_grid;
```

就会得到一个北美的兰伯托投影。现在再尝试：

```
m_ungrid
```

海岸线还在，但是网格消失了，坐标轴显示原始的 X/Y 投影坐标。现在试试这个：

```
m_grid('xtick',10,'tickdir','out','yaxislocation','right','fontsize',
7);
```

可以被改变的选项是：

'box',('on' | 'off' | 'fancy')

这指定外框是否绘制。共有三种类型可以选择：默认的是 'on'，就是简单的线条。有两种精致 ('fancy') 的外框样式可以被选择。如果 'tickdir' 是 'in'，那么就是黑白块交替的效果（译者注：即铁路线效果）（参考[例 2](#)）。如果 'tickdir' 是 'out'，那么就是一个更复杂的图案效果（参考[例 6](#)）。精致的外框一般只适用于经纬度限制的地图（即不适用于方位投影），但是如果该选项选择不恰当，会出现警告消息。

'xtick',(num | [value1 value2 ...])

指定经度网格的数量或者位置。如果只指定了一个数字，那么网格或者值就会大约在这个数值等距的位置（这个数值只能是大约，因为 `M_GRID` 会试图去寻找“好看的”间距，即以一个增长率不断的循环）。确定的位置可以通过一个由位置数值组成的数组来指定。也有一个类似的 'ytick' 属性。

'xticklabels',[label1;label2 ...]

特定的标签可以被指定。标签可以是数值（会被 `M_GRID` 格式化），也可以是使用时不会产生变化的字符串。也有一个类似的 'yticklabels' 属性

'xlabledir',('middle' | 'end')

经度的标签要么是对齐在网格线的一端(与这些线正交),要么在这些线的末端延伸出来。也有一个类似的'ylabeldir'属性。

'ticklen',value

给定标记的长度(作为绘图宽度的一部分)。

'tickdir',('in' | 'out')

指定刻度线向内还是向外。如果'box'被设置为'fancy',本属性将指定精致外框的类型。

'tickstyle',('dd' | 'dm')

指定坐标轴的标签单位是十进制的度还是度-分(默认即此)。

'color',colourspec

'linewidth',value

'linestyle',(linespec | 'none')

'fontsize',value

'fontname',name

指定网格和其标签的不同线条/文本属性。

'XaxisLocation',('bottom' | 'middle' | 'top')

指定 X 轴的位置,要不然就是在底部(最南端),要不然就是在顶部(最北端),或者是在中间。

'YaxisLocation',('left' | 'middle' | 'right')

指定 Y 轴的位置,要不然就是在左部(最西端),要不然就是在右部(最东端),或者是在中间。

4.2 标题和 x/y 轴标签

标题和 x/y 轴标签可以通过使用标题和 x/y 轴标签函数来添加(在 V1.0 版本中'visible'属性必须明确的设置为'on',现在已经改为在 m_grid 函数内执行)。

4.3 图例

可以使用 m_legend 在地图中添加图例。目前只包括一些图例的功能。图例盒可以使用鼠标按钮进行拖放。

4.4 比例尺

可以使用 m_ruler 在地图上面添加比例尺。比例尺可以垂直绘制也可以水平绘制,而且其标签数会使用一个让其美观的数值(尽管这个可以通过调用其它函数进行改变)。其位置依据标准化坐标进行指定(即 0-1 之间),因此你可以在地图上调整其位置。建议最好在调用 m_grid 之后调用本函数,因为 m_grid 会将标准化重置。

注意:比例尺对于任何全球或者甚至是地球上的一块有效部分都可能没有用,但是我是不会阻止你使用它的。警告用户!

5 添加自己的数据

这个工具箱的目的就是允许你在地图上绘制你自己的数据!一旦选择了合适的网格和(可能有)海岸线,你就可以通过内置的 M_Map 绘图函数添加你自己的线条、文本或者等值线图。这些函数会将经纬度坐标处理变换成投影坐标系。这些绘制函数与标准的 Matlab 绘图函数很是类似,在下一部分会介绍。

有时你需要进行经纬度和投影坐标之间的转换但是并不立即绘图。这可能在你想使用 `ginput` 交互选择点或者你想绘制和屏幕上指定点而非指定的经纬度有关的标签时发生。投影坐标的转换在 5.2 和 5.3 中会被阐述。一旦原始的经纬度坐标被转换成投影坐标，标准的 Matlab 绘图函数就可以使用了。

地图的大小被绘制成符合绘图坐标系的边界。因此他们的比例尺是有些随意的，如果你对绘制给定比例尺的地图，比如说 1:200000 或者类似的，你可以使用基本的 `m_scale` 函数，参见 2.6。数据的单位是距离都被表示成地球半径的分数的投影坐标系。想得到地图上两点之间的“距离”，使用投影坐标系下面的笛卡尔距离并且将其乘以你喜欢的地球半径数值，一般是 6370km 左右（有一个例外——UTM 投影的东向和北向以米为单位，因此不需要转换）。

注意：有时发生的一个问题是因为经度数据模糊造成数据不会出现。例如，如果绘图的经度范围是 $[-180, 180]$ ，一个点的经度是 200，这就不会在圆锥投影和圆柱投影中出现。这不是错误。处理包围曲线的剪裁需要加点（而不是移动他们），因此与一系列其它要求不匹配（比如说在下面阐述的坐标转换中保持输入和输出矩阵大小移植）。

5.1 绘制线条、文本、箭头、填充、阴影、斑点状和轮廓

大多数情况下，你不需要知道投影坐标系到底是什么——你只是想在指定的经纬度上绘制一些东西。大多数时间当你在地图上面绘制一些东西，你想做的其实是在指定的经纬度坐标下完成，而非通常的 X/Y 位置。在 M_Map 实现这一点，需要调用能够识别经纬度坐标的 `plot`, `line`, `text`, `quiver`, `patch`, `contour` 和 `contourf` 在 M_Map 中的等价版本：在函数名称之前加上了“m_”。例如：

```
m_plot(LONG,LAT,...line properties) %在地图上面画一个线条（清除现有线条）
m_line(LONG,LAT,...line properties) % 在地图上面画一个线条
m_quiver(LONG,LAT,U,V,S) %箭头线条
m_text(LONG,LAT,'string') % 文本
m_patch(LONG,LAT,..patch properties) % 填充
```

这些函数的每一个都会自备坐标转换功能，并且会在有需求的情况下返回一个包含图形对象句柄的数组。这些函数与标准 Matlab 函数唯一的区别就是前两个参数必须是经纬度。

关于 `m_patch` 有一点非常需要注意。由于兼容性的原因，这个和填充海岸线使用的是相同的代码。海岸线必须是“岛屿”或者“湖泊”，并且 M_Map 会持续通过假设曲线都在其右侧来追踪其区别。因此被填充的区域（“陆地”）都是在环绕曲线右侧的。这和 `patch` 中总是填充内部有一些轻微的不同。在直角坐标系中，保持追踪差异还是比较简单的，但是在球坐标系中就没那么简单了。`m_patch` 在缺少其他信息的情况下会试着做正确的事情，但是（特别是当填充区域和地图边界相交的时候）它可能会搞混。如果填充的不正确，可以试着使用 `flipud` 或 `fliplr` 将点的顺序倒过来。

格点的经纬度的数据也可以用等值线图画出来。

```
m_contour(LONG,LAT,VALUES)
m_contourf(LONG,LAT,VALUES)
```

同样，这些函数也会返回图形对象的句柄，允许（例如）绘制标记的等值线图。

```
[cs,h]=m_contour(LONG,LAT,VALUES)
```



```
clabel(cs,h,'fontsize',6);
```

精致的箭头（即指定宽度、箭头形状和颜色）可以使用 `m_vec.m` 创建。其有关信息可以参阅在线帮助。

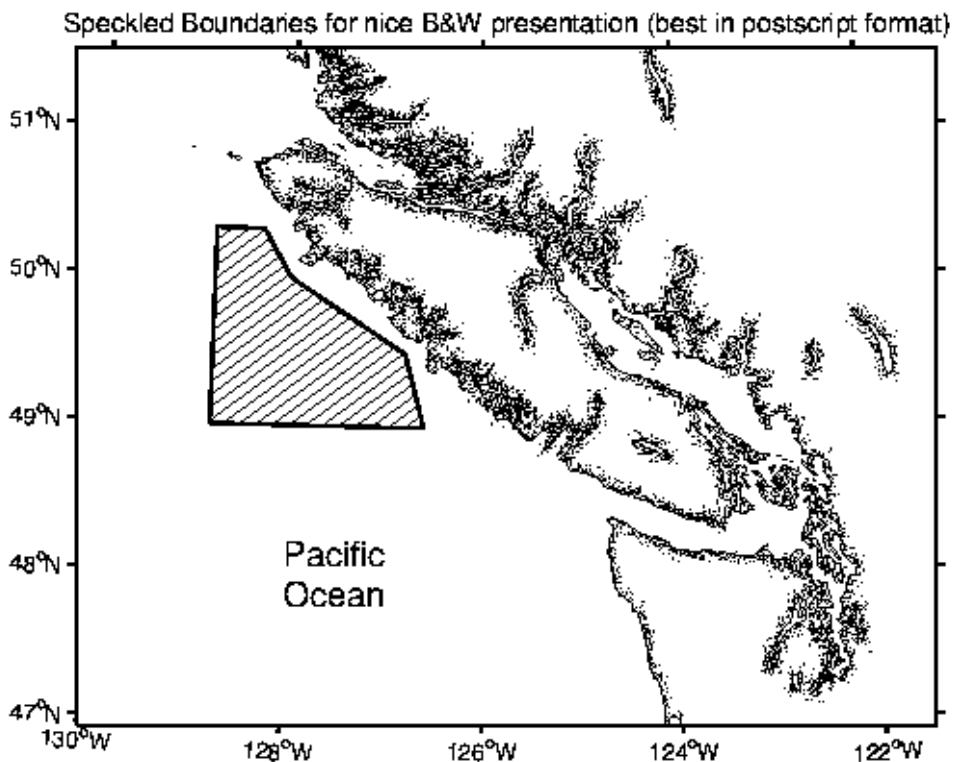
你还可以通过调用 `m_hatch` 来获得阴影区：

```
m_hatch('single',LONG,LAT,...hatch properties) % 单线阴影
m_hatch('cross',LONG,LAT,...hatch properties) % 交叉线阴影
```

需要注意的是这种调用不会产生边线（如果想添加边线，需要额外调用 `m_line`。此外，我们可以对填充的边缘通过如下语句实现斑点效果：

```
m_hatch('speckle',LONG,LAT,...speckle properties) % 斑点状边缘
)
```

使用 `m_hatch` 的更多细节和有关信息可以参阅在线帮助和/或[例 13](#)。



5.2 绘图与 `p_color`

`m_pcolor` 是 `p_color` 的替代，但是在地图边界附近一定要谨慎使用。理想状态下，数据会延伸至地图边界（而不会与之交叉）（即被剪裁成多边形）。然而由于 Matlab 平面处理的方法实现并不简单。与之相对的是，除非你使用简单的圆锥投影或者圆柱投影，否则你就会得到一个带有锯齿状边缘的彩色平面。

`m_image` 函数是不存在的。`image()` 函数只使用矩形像素展示数据图，并且一般投影数据都不会以矩形像素的形式出现。如果你想在地图上展示一个大的位图，有以下几种选择：

如果地理参考图像使用经纬度坐标（即每行每列的经纬度都是一致的），可以使用带有 `shading flat` 参数的 `m_pcolor`。尽管对于大图可能会慢一些，但是效果还是相当令人满意的，

但是你首先必须要记得将坐标偏移半个像素的距离。这是因为在给出相同数据的情况下，`p_color` 和 `image` 会给出不同的结果。

`image` 会在 X/Y 矩阵的第(i,j)个元素的中心绘制像素(i,j)。

使用 `shading flat` 参数的 `p_color` 会在 X/Y 矩阵的(i,j),(i+1,j),(i+1,j+1),(i,j+1)坐标中间绘制一个色块用以代表(i,j)的数值。因此，所有都要移动半个像素的距离。

中纬度的卫星数据有时对这个解决方案很好用，参考卫星数据处理的例子。

如果图像已经被置于某种投影之下，并且如果你知道投影的具体参数，你可以直接调用图像然后重复绘制 `M_Map` 地图。例如，极轨卫星的图像一般是极射赤平投影。在这种情况下，你应该使用 `m_ll2xy` 获取图像四角的屏幕坐标，然后在重新绘制数据之前调用 `image()`，并在该函数中使用这些点。参考特定的该例。

提示:为保证执行正确，应检查海岸线是否重新绘制。

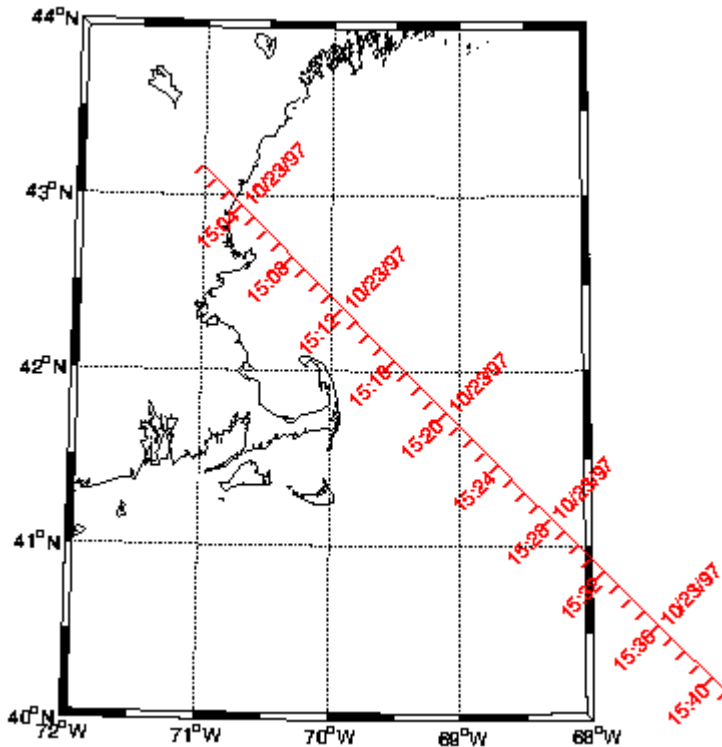
5.3 绘制轨迹

有时标注表示船舶、飞行器或者卫星随时间变化的位置曲线，并且添加时间标注是很有用的。这个可以使用 `m_track` 实现。

```
m_proj('UTM','long',[-72 -68],'lat',[40 44]);
m_gshhs_i('color','k');
m_grid('box','fancy','tickdir','out');

% 伪造一条轨迹
lons=[-71:.1:-67];
lats=60*cos((lons+115)*pi/180);
dates=datetime(1997,10,23,15,1:41,zeros(1,41));

m_track(lons,lats,dates,'ticks',0,'times',4,'dates',8,...
        'clip','off','color','r','orient','upright');
```



关于 `m_track` 和其设置字体、标注间隔、日期格式等选项的用法的更多详细信息请参考在线帮助。

摆弄着不同参数，清除绘制的轨迹但是保留海岸线和网格却经常很方便。这个可以通过如下语句实现：

```
m_ungrid track
```

或

```
m_ungrid('track')
```

5.4 绘制等距圆环和测地线

一件很漂亮的事情是：有时绘制一个到指定的地点距离给定的圆环很有用。这个可以通过调用 `m_range_ring` 实现。其有三个必要参数：LONG, LAT, RANGE（经度、纬度、距离），其次都是一些（可选的）线条属性。[例 11](#) 讲解了如何使用 `m_range_ring`。

如果要绘制圆形的测地线（即在所有距离上都垂直于等距圆环的曲线），`m_lldist` 可以同时在这点中间找到距离和沿着测地线的点。[例 13](#) 讲解了如何使用 `m_lldist`。

如果关心大圆和椭圆测地线之间的差异（我敢打赌只有很少很少比例的用户），`m_fdist`（计算从一个给定的距离/方位到另外一个的距离）、`m_idist`（点之间的距离和方位）、`m_geodesic`（沿着测地线的点）可以配合一系列（用户指定的）椭圆使用。由于历史原因，这些调用的顺序和 `m_lldist` 是不同的。

5.5 从经纬度转换到投影坐标

如果你想使用投影坐标（可能你想计算地图面积或者距离，或者你想在左上角放一个图例），

下面的命令可以将经纬度坐标转换成投影坐标：

```
[X,Y]=m_ll2xy(LONG,LAT, ...optional clipping arguments )
```

这里 LONG, LAT, X, 和 Y 是相同大小的矩阵。投影坐标在地图中心附近和真实距离相同，并且表示成地球半径的一部分。想得到距离，需要乘以地球半径（6370KM）。UTM 投影在北方和东方提供以米为单位的坐标，这是例外。

可能的裁剪参数有：

'clip','on'

默认。LONG 和 LAT 的列被假设成表格线，并且这些都依据地图限制进行剪裁。因此地图以外的第一个点被移动到地图边缘，其他点被转换成 NaN。

'clip','off'

不进行剪裁，有时对于调试目的很有用。

'clip','point'

测试不在地图范围内的点。范围外的点被转换成 NaN，范围内的点转换成投影坐标。没有被移动的点。该选项对于点状数据（比如站点位置）很有用。

'clip','patch'

测试不在地图范围内的点。那些在限制以外的点被变成恰好在范围上的点。那些在范围内的被转换成投影坐标。这个选项对于视图填充可能有效，然而可能并不是很好用

5.6 投影坐标转换成经纬度坐标

可以直接从投影坐标转换成经纬度坐标：

```
[LONG,LAT]=m_xy2ll(X,Y)
```

没有其它的选项。

5.7 计算点之间的距离

球形地球上的测地线（大圆）距离可以通过两点之间的地理（经纬度）或者地图坐标（X/Y）使用 m_lldist 和 m_xydist 函数进行计算。例如：

```
DIST=m_lldist([20 30],[44 45])
```

该句计算 20E, 44N 到 30E, 45N 的距离。另外如果你想计算鼠标选择的两点之间的距离：

```
[X,Y]=ginput(2);
DIST=m_xydist(X,Y)
```

这样就会返回距离。因为不准确的球形地球近似，计算出来的距离与真实地理距离会有 1% 左右的误差。

如果你想更精确，必须在椭圆形地球上计算测地线。关于这个有一个非常精确的数值算法（[Vincenty's algorithm](#)，其在 m_idist, m_fdist, 和 m_geodesic 函数中有运用。例如：

```
[distance,a12,a21] = m_idist(lon1,lat1,lon2,lat2,spheroid)
```

将计算(lon1,lat1) 和 (lon2,lat2)在特定的球体（默认是'wgs84'，其它选项请参考代码或者使用 `m_proj('get','utm')`所示的选项）上的以米为单位的距离。正向和反向方位角 `a12` 和 `a21` 也会以角度为单位被计算。

`m_fdist` 被用作获取从一个指定点的给出距离和方位角之后的位置。

最后，如果你想在地图上绘制测地线，`m_geodesic` 可以被用来生成两个指定点之间沿着椭圆测地线的一系列点的矢量。如果你发现自己需要这个，我会对了解这个感兴趣！

6 更多复杂图形

对于如何绘制更多复杂图形的想法，请参考例子。这些图形也包含着在 `m_demo` 函数中。

7 从图中移除数据

一旦给定的地图包含几个元素，就很有必要进行一系列操作来满足人类对于图像感官质量。如果图像包含复杂的海岸线（比如说下面讨论的那些），其需要绘制时间比清除时间还长，并且重绘会很无聊。上面介绍的 `m_ungrid` 命令可以用来选择图形中移除的部分。例如：

```
m_proj('lambert','long',[-160 -40],'lat',[30 80]);
m_coast;
m_range_ring(-123,49,[1e3:1e3:10e3],'color','r');
```

该命令绘制了距离我办公室 1000km 的等距圆环。但是我对其并不满意，并且希望重新绘制的间隔仅为 200km。我可以将 `m_range_ring` 的效果移除，并且使用以下语句重新绘制：

```
m_ungrid range_ring
m_range_ring(-123,49,[200:200:2000],'color','r');
```

一般一切以 `m_`开头的函数结果都可以通过调用 `m_ungrid` 加上函数名称来删除。

`m_ungrid` 可以通过寻找所有由不同 M_Map 语句设置的绘图元素的“标签”属性来识别并删除特定的元素。

8 添加自己的海岸线

如果你对于一个特定的区域很感兴趣，并且想要一个比使用 `m_coast` 更高分辨率的海岸线，最好的选择是：

从一些高分辨率的数据库获取点集，并且

使用 `m_ll2xy` 将其转换为屏幕坐标系并绘图

一个可以获取高分辨率海岸线数据的地方是 [The Coastline Extractor](#)。在那里按照提示可以将海岸线数据保存到一个 Matlab 可以读取的文件中并且下载到电脑里。如果文件保存名称为 "coast.dat"，你就可以按照如下方法绘图（绘制线条）：

```
load coast.dat
m_line(coast(:,1),coast(:,2));
```

填充海岸线需要更多的工作。首先需要阅读将海岸线数据拼接成连续段的说明。如果你很幸运（也就是没有湖和其它东东），你可能会通过以下语句取得成功：

```
load coast.dat
[X,Y]=m_ll2xy(coast(:,1),coast(:,2),'clip','patch');
k=[find(isnan(X(:,1)))];
for i=1:length(k)-1,
    x=coast([k(i)+1:(k(i+1)-1) k(i)+1],1);
    y=coast([k(i)+1:(k(i+1)-1) k(i)+1],2);
    patch(x,y,'r');
end;
```

如果这无效，阅读 `private/mu_coast` 中的说明，让曲线适应需要的样式并且使用 `m_usercoast` 来载入自己的数据。

8.1 DCW 政治边界

由于 2011 年 DCW 网站已经停用。下面的信息只因为历史原因而保留。新的用户请参考下一节关于形状文件的内容。

可以从 <http://www.maproom.psu.edu/dcw/> 下载许多国家和美国各州的政治边界文件。选择一个地区然后选择“下载点”（“download points”）选项（而不是“下载数据”（“download data”））下载完成之后，可以使用 `m_plotbndry` 来进入并且绘制需要的边界。比如说，如果你下载了美国各州的边界，并且存入了子目录“states”：

```
m_plotbndry('states/arizona','color','r')
```

这样就会在当前地图上绘制亚利桑那州了。

8.2 ESRI 形状文件

矢量数据交换的试试标准是 ESRI 形状文件。一个数据集至少包括三个文件，每一个都有相同的根名称，但是有三种扩展名：`.dbf`、`.shp` 和 `.shx`。文件可以包含点、线或者多边形信息，以及其它自描述方式的域。更多信息请参阅[此说明](#)。

很多（所有）形状文件都可以使用 `m_shaperead` 进行读取，并且返回一个包含文件信息的数据结构。然而，搞清楚这些内容能做什么需要检查数据结构的内容。在 `.prj` 文本文件中检查投影信息也是很有用的，特别是当你需要在其他来源的数据上重叠的时候更应如此。因为这些文件包含地图投影的有关信息。

一般你可以至少在文件 `datafile.shp`、`datafile.shx` 和 `datafile.dbf` 储存的数据中创建一个简单的绘图通过使用：

```
M=m_shaperead('datafile');
clf;
for k=1:length(M.ncst),
```

```
line(M.ncst{k}(:,1),M.ncst{k}(:,2));  
end;
```

9 添加自己的地形/水深

在 [NCAR](#) 可以获得很多全球或区域的地形数据库。可以从他们的 [FTP 网站](#) 中免费获得一些。只要数据以经纬度矩阵的形式被存储在 mat 文件中,就可以使用 `m_contour` 或者 `m_contourf` 进行绘制。

9.1 Sandwell 和 Smith 水深

最近,这个分辨率大约 1 公里的在低纬地区的水深数据正在被很多人使用。该数据集的描述在 http://topex.ucsd.edu/marine_topo/, 可以在 ftp://topex.ucsd.edu/pub/global_topo_2min/ 获取一个约 134Mb 的二进制文件(文件名: `topo_X.Y.img`, 其中 X 和 Y 是版本号)。作者已经包含一个可以提取部分数据的 m 文件 ([mygrid_sand.m](#)) 在其中(需要自行在代码中改动路径名)。一旦这个数据库(和这个 m 文件)安装到电脑中,你就可以在 M_Map 中很轻易的使用它。以下是一个典型应用:

% 提取数据

```
[elevations,lat,lon]=mygrid_sand([lat_south lat_north long_west  
long_east]);
```

%在 M_Map 命令中使用

```
m_contour(lon,lat,elevations);
```

对于一些投影,你必须确保 `mygrid_sand.m` 返回的 'lon' (经度) 值在投影的范围之内(即你可能需要加/减 360)。在西半球貌似这个都是这样的(即负经度),如果你忘了这一点,你就会经常得到关于空矩阵的错误信息。

10 使用 TerrainBase 5 分钟或者 ETOPO2 2 分钟的全球水深/地形

1. M_Map 提供的高程数据库在很多用途下都会提供足够的分辨率。然而,也有很多需要更多细节的情况。我没有引入更高分辨率的数据库,因为这会大大的增加软件包的大小。然而, v1.2 版本包含可以在几分钟之内访问并且绘制全球 5 分钟水深/地形数据库的 m 文件。本章提供如何下载 [TerrainBase](#) 并使用 `m_tba2b.m` 将其从一个 56Mb 的 ASCII 文件转换成一个 18Mb 的二进制文件的说明。然后就可以简单的使用 `m_tbase.m` 从这个文件中读取并且绘制水深,这和 `m_elev` 的使用基本上是相同的方式(参见 3.2)。

TerrainBase 也可以通过 CDrom 访问,并且也通常在很多学术网络中使用 `netcdf`(或者其它)二进制格式储存。如果你可以修改 `m_tbase.m` 使得其可以从其中一种来源读取文件,请告诉

我。

如何安装 TerrainBase:

从 <http://dss.ucar.edu/datasets/ds759.2/> 上获取并解压缩 tbase.Z 文件, 存于 m_map 目录中。

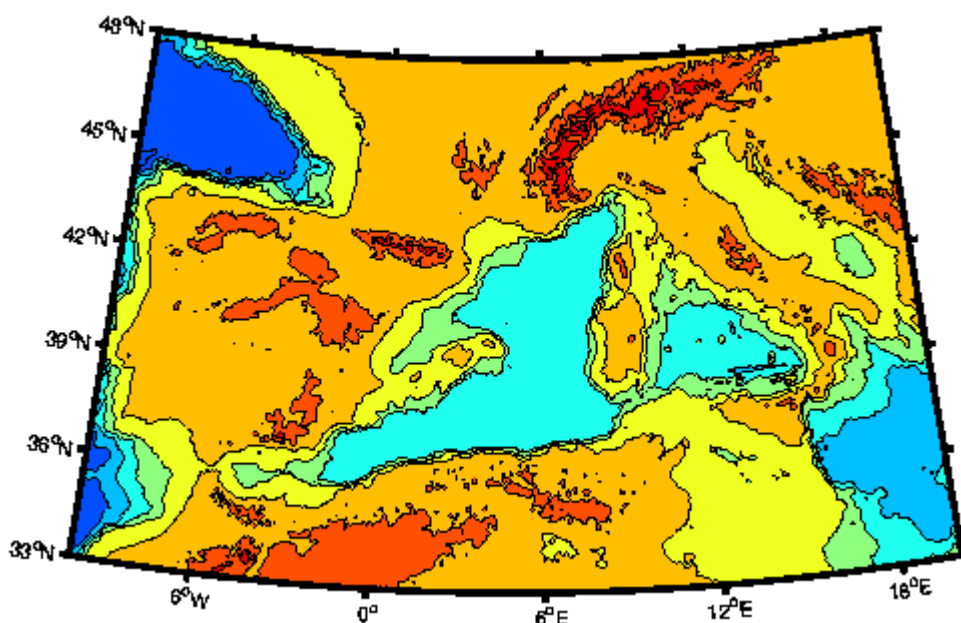
运行 m_tba2b(PATHNAME) 将输出的 18Mb 二进制文件存储为 PATHNAME/tbase.int。

删除原始的 ASCII 文件 tbase。

编辑在 m_tbase 中设置的 PATHNAME 以指向文件的位置。

搞定了! 使用西地中海的地图来测试:

```
m_proj('lambert','lon',[-10 20],'lat',[33 48]);  
m_tbase('contourf');  
m_grid('linestyle','none','tickdir','out','linewidth',3);
```



2. 2006 年 4 月, 有一个修正的高分辨率数据库 (2 分钟) [ETOPO2](http://dss.ucar.edu/dsszone/ds759.3/etopo2_2006apr/etopo2_2006apr.raw.gz)。下载地址: http://dss.ucar.edu/dsszone/ds759.3/etopo2_2006apr/etopo2_2006apr.raw.gz (一个压缩的二进制文件)。解压成一个 116Mb 的文件, 在 m_etopo2 中修改 PATHNAME 的设置以指向文件位置, 然后你就可以像 m_tbase 和 m_elev 一样用它了。UCAR 建议用户注册。如果你不注册, 第二个链接就不好使 (进入第一个链接并且按照说明操作)。

11 使用 GSHHS 高分辨率海岸线数据库

11.1 安装 GSHHS

绘制地图时, 总是有一个在程序运行时间和生成地图分辨率直接的权衡。M_Map 中内含一个 1/4 度的海岸线数据库来快速生成地图, 在这很多用途下都是足够的分辨率。

然而, 通常在有限的地理区域中制作详尽的地图才是令人满意的。为了这个目的, 必须要有一个高分辨率的海岸线数据库是。我没有在 m_map 中引入这样的数据库, 因为这会大大的

增加软件包的大小。然而，我内置了可以读取并使用一个叫做 [GSHHS](#) 的高分辨率数据库的 `m` 文件。

如同发布的一样，GSHHS 包含了一个不同分辨率下的分层数据库集。尽管其包含很多内陆湖泊，但是其最低或“最粗”的分辨率没有 M_Map 好。“高”分辨率的所包含的点距大约为 200m。还有一个更好的“全”分辨率。你可以安装这个数据库的部分或全部（取决于你的磁盘空间）。“全”分辨率大约占用 90Mb 的硬盘空间，更粗糙的分辨率大约小 1/4。“高”分辨率的大约 21Mb，“中”分辨率大约 6Mb，“低”分辨率大约 11Mb（一个并不总是使用“高”分辨率的原因是其 90Mb 的数据库每次调用必须全部读取和处理，这会需要一些时间的）。

如何安装 GSHHS:

1. 访问 <http://www.ngdc.noaa.gov/mgg/shorelines/data/gshhs/>
2. 下载并解压以下全部或者任何文件于一个方便的目录中：`gshhs_c.b.gz`, `gshhs_l.b.gz`, `gshhs_i.b.gz` 和/或 `gshhs_h.b.gz`, 推荐。GSHHS 数据格式在 V1.2 与 1.3 版本中发生了改变，并且在 V2.0 中再次改变，然而 `m_map` 应该能够分辨出这一点。
3. 如果数据库文件不在 `m_map/private` 的子目录中，你必须将编辑在 `m_gshhs_c.m`, `m_gshhs_l.m`, `m_gshhs_i.m`, `m_gshhs_h.m` 和/或 `m_gshhs_f.m` 中的 `FILENAME` 的设置指向相应的文件。

11.2 有效的使用 GSHHS（译者注：原文为 4，但是根据上下文逻辑应为 11.2）

最简单的调用机制就是与 `m_coast`（[第 3 章](#)）的一致。例如，绘制一个灰色填充的高分辨率海岸线：

```
m_gshhs_h('patch',[.5 .5 .5]);
```

然而，运行的时间会非常非常长，因为整个数据库必须被搜索和处理。我不建议使用中高分辨率的海岸线绘制世界地图！有两种方式加速。第一种是使用低分辨率，点少一些的数据库。第二种在你需要重新绘制地图的时候更使用一些（例如这是你工作的底图）。在这种情况下，我建议保存一个中间处理文件（通常更小）如下：

```
m_proj ... % 设置投影参数
```

```
%这个命令什么也不画—它只是使用
```

```
%投影参数处理高分辨率数据库
```

```
% 来生成一个叫做"gumby"的较小的海岸线文件
```

```
m_gshhs_h('save','gumby');
```

```
% 现在可以更快的绘制一些相同区域的地图
```

```
figure(1);
```

```
m_usercoast('gumby','patch','r');
```

```
m_grid;
```

```
figure(2);
```



```
m_usercoast('gumby','linewidth',2,'color','b');  
m_grid('tickdir','out','yaxisloc','left');
```

等等

12 M_Map 工具箱内容和功能

1. Contents.m - 工具箱的内容

2. m_demo.m - 演示了几个地图

用户可调用的函数:

1. m_proj.m - 初始化投影

2. m_coord - 地磁坐标转换为地理坐标

3. m_grid.m - 绘制网格

4. m_scale.m - 将地图强制调整为给定的比例尺

5. m_ruler - 绘制比例尺

6. m_ungrid.m - 删除地图元素 (如果你想改变参数)

7. m_coast.m - 绘制海岸线

8. m_elev.m - 绘制高程数据

9. m_tbase.m - 从高分辨率数据库中绘制高程数据

10. m_etopo2.m - 从 (另一个) 高分辨率数据库中绘制高程数据

11. m_gshhs_c.m - 从 GSHHS 粗分辨率数据库中绘制海岸线

12. m_gshhs_l.m - 从 GSHHS 低分辨率数据库中绘制海岸线

13. m_gshhs_i.m - 从 GSHHS 中分辨率数据库中绘制海岸线

14. m_gshhs_h.m - 从 GSHHS 高分辨率数据库中绘制海岸线

15. m_gshhs_f.m - 从 GSHHS 全分辨率数据库中绘制海岸线

16. m_plotbndry.m - 从 DCW 数据中绘制政治边界

17. m_usercoast.m - 使用用户指定的数据库绘制海岸线

18. m_shaperead.m - 读取 ESRI 形状文件

19. m_plot.m - 在地图坐标上绘制线条数据

20. m_line.m - 在地图坐标上绘制线条数据

21. m_text.m - 在地图坐标上添加文本数据

22. m_legend.m - 绘制图例

23. m_patch.m - 在地图坐标上添加填充数据

24. m_pcolor - 绘制 pcolor 平面

25. m_quiver - 为矢量数据绘制箭头

26. m_contour - 为网格数据绘制等值线

27. m_contourf - 绘制填充等值线

28. m_track - 绘制注释轨迹

29. m_range_ring - 绘制等距圆环

30. m_ll2xy.m - 从经纬度坐标转换成地图坐标

31. m_xy2ll.m - 从地图坐标转换成经纬度坐标

32. m_geo2mag - 从磁坐标转换成地理坐标

- 33. `m_mag2geo` – 从地理坐标转换成磁坐标
 - 34. `m_lldist` – 经纬度坐标下点之间的距离
 - 35. `m_xydist` – 地图坐标下点之间的距离
 - 36. `m_fdist` – 在椭球地球上给定距离/方位角的位置
 - 37. `m_idist` – 在椭球地球上点之间的距离/方位角
 - 38. `m_geodesic` – 在椭球地球上两点之间的测地线点
 - 39. `m_tba2b.m` – 用于安装高分辨率的高程数据库
 - 40. `m_vec.m` – 精致的箭头
- 内部函数（并不意味着用户可以调用）
- 1. `private/mp_azim.m` – 方位投影
 - 2. `private/mp_cyl.m` – 圆柱投影（赤道）
 - 3. `private/mp_conic.m` – 圆锥投影
 - 4. `private/mp_tmerc.m` – 横向圆柱投影
 - 5. `private/mp_utm.m` – 椭球通用横向圆柱投影
 - 6. `private/mp_omerc.m` – 斜圆柱投影
 - 7. `private/mu_util.m` – 一些实用程序
 - 8. `private/mu_coast.m` – 处理海岸线的程序
 - 9. `private/mc_coords` – 坐标转换
 - 10. `private/clabel.m` – `clabel` 的布丁版本（Matlab v5.1 版本不包括不同文本属性的功能）
`private/m_coasts.mat` – 海岸线数据
- HTML 文档
- 1. `map.html` – 文档介绍
 - 2. `private/mapug.html` – 用户指南
 - 3. `various.gif` – 例子

13 已知的问题和错误

- 1. 在 Matlab 5.0（学生版？）运行 `M_Map` 有时会出错，因为 5.0 有着后续版本不会出现的一些列错误和“功能”。一个会经常会突然出现的就是在使用 `m_coast.m` 的时候找不到 `m_coast.mat`。最简单的解决方案就是将 `../m_map/private` 放到和 `../m_map` 一样的路径中（在之后的版本中，Matlab 会找到 `mat` 文件）。
- 2. 在 PC 上使用 Matlab 5.1 运行 `M_Map` 有时会产生很多这个信息：

```
> Warning: Divide by zero.
```

这个归咎于 Matlab 的一个错误（实际上归咎于编译器 TMW 的使用），这个会在一些数字除以 NaN 的时候引起错误的警告标记设定。你可以放心的忽略这些错误并使用 V5.2 版本。

- 3. 如果你绘制的数据是白色的，在 `postscript` 输出的时候会变成黑色。这是因为打印命令工作的问题。为了避免这种情况，将图片的背景设为白色，即：

```
set(gcf, 'color', 'white')
```

4. 当你使用填充等值线的时候，一般会出现奇葩的填充。出于某种原因，这个在 Matlab 中已经是各种奇葩错误的大坑了。大多数多多少少都和地图背景与 `contourf` 的填充有交叉或者“填充器”（Matlab 的内部代码，决定绘制什么）的工作与否有关。不幸的是，我不能考虑到在所有情况下解决问题的方案，但是如果你看到奇葩的东东，试试在 `m_grid` 之后调用：

```
set(findobj('tag','m_grid_color'),'facecolor','none')
```

或者：

```
set(gcf,'renderer','opengl');
```

（在 Unix 中你可能在启动 Matlab 之后就要这么做）

5. 在 `tiff` 输出中没有正确显示。Matlab 使用 `ghostscript` 将 `ps` 转化为其它格式。但是他们的版本有一些问题。更好的方法是打印输出到 `postscript` 并且自己执行转换（比如说到 `tiff`）。

6. 译者注：在新版 Matlab 中已经取消 `bitmax`，使用 `flintmax` 取而代之。因此在调用 `m_grid` 的时候会报错：

```
>BITMAX has been removed. Use FLINTMAX instead.
```

解决方案是在 `m_grid` 的第 263 行中，将 `bitmax` 改为 `flintmax`。

14 OCTAVE 兼容性问题

在他们的网站中可以看到：“[GNU Octave](#) 是一个高级的解释语言，主要为数字计算准备 [...] Octave 语言和 Matlab 很像，因此大部分程序都可以简单移植。”

M_Map 目前可以部分在 Octave 下运行。在 M_Map 1.4f 中，一些小的不兼容性已经解决了（大部分是拼写出图形对象的全名而不是只在 Matlab 中允许的简写），因此不会一下子出大错。然而，还是有一些功能不工作或者工作不正常。

注：因为“private”子目录没有执行（例如 Octave 3.2.3），因此你必须在路径中明确添加 `m_map/private` 子目录：

```
addpath [whatever]/m_map
addpath [whatever]/m_map/private
```

不能工作会或者工作不正常的功能包括：

1. Octave `line` 命令不接收 `MATRIX x/y` 参数。这就意味着在调用 `m_line` 的时候你的代码必须要改变。
2. 3D 填充功能在 Octave 中还没有完全实现。这就意味着精致的外框不会直线，所以 M_Map 一旦发现在 Octave 下使用，这些功能就会被关闭。
3. 2D 功能还不完善。填充不总是正确的填充。这就意味着填充海岸线或者填充等值线有时看起来是错误的。还是坚持线条图吧。
4. 在 Octave 下线条对象的剪切功能还不完善。甚至有时图形边界外的线条当 `'clipping'` 设置为 `'off'` 的时候还是不会画出来。忍着吧。

5. 简单的使用 `m_elev` 调用水深是来源于一个内置的 Matlab demo .mat 文件，这在 Octave 中不可行。
6. 图形中在其它物体“前面”或者“后面”的物体可能正确的好像在前面或者后面一样展示出来。
7. 有时文本标签的对齐方式可能或有点古怪（特别是在经纬度网格下）。忍着吧。
8. 阴影和斑点状的外廓需要坐标轴“点”的单位，并且尽管你可以改变“点”的单位属性，但是返回的位置数值并没有改变。所以尽量避免阴影或者斑点状。

15 自上一版本以来的变化

1. 终于包含了 `m_pcolor`。事实上我很久之前就能搞定了，但是有一些哲学上的原因我没有——它并没有在做“正确”的事情。但是另一方面他还是做了*一些事情*（并且我给出了例子）。
2. 地磁/地理坐标系统。如果你使用这个请让我知道。增加这个是一件有趣的事情，但也许只是浪费时间...
3. 在椭球地球上超级准确的测地线！
4. 老式手工绘制的斑点状海岸线外观
5. 一个形状文件读取器。

最后一次修改于 4/Dec/2011. 问题和意见请联系: rich@eos.ubc.ca

中文版翻译 (2016/03/05): 史纬恒 happytfotshit@hotmail.com

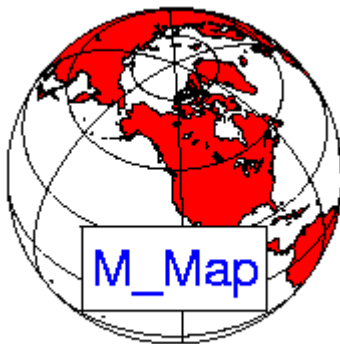
校对: 张钰 yaohuazhangyu@hotmail.com

孔峰 kongfeng0824@qq.com

Examples

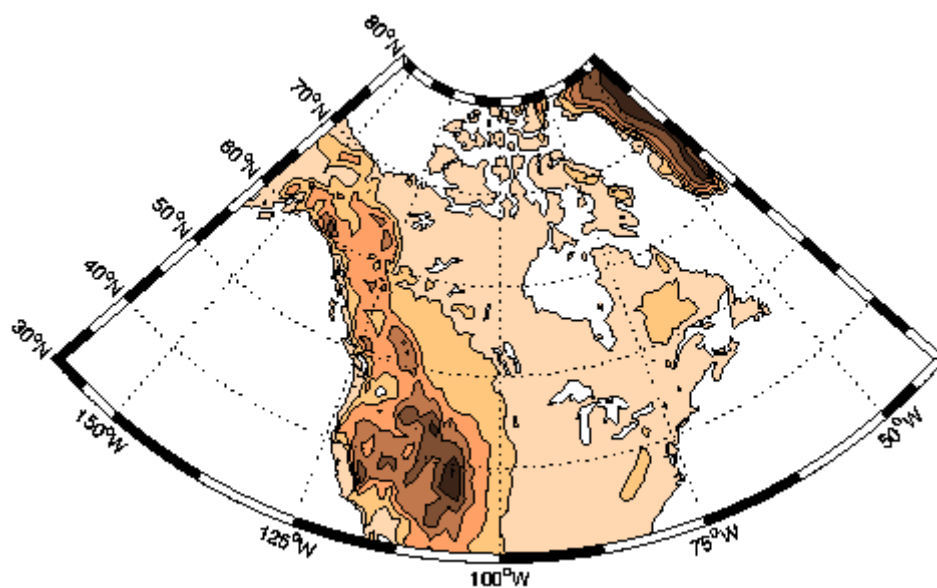
1. M_Map 标志

```
m_proj('ortho','lat',48,'long',-123);
m_coast('patch','r');
m_grid('linest','-','xticklabels',[],'yticklabels',[]);
patch(.55*[-1 1 1 -1],.25*[-1 -1 1 1]-.55,'w');
text(0,-.55,'M_Map','fontsize',25,'color','b',...
     'vertical','middle','horizontal','center');
set(gcf,'units','inches','position',[2 2 3 3]);
set(gcf,'paperposition',[3 3 3 3]);
```



2. 兰伯托投影下的北美地形图

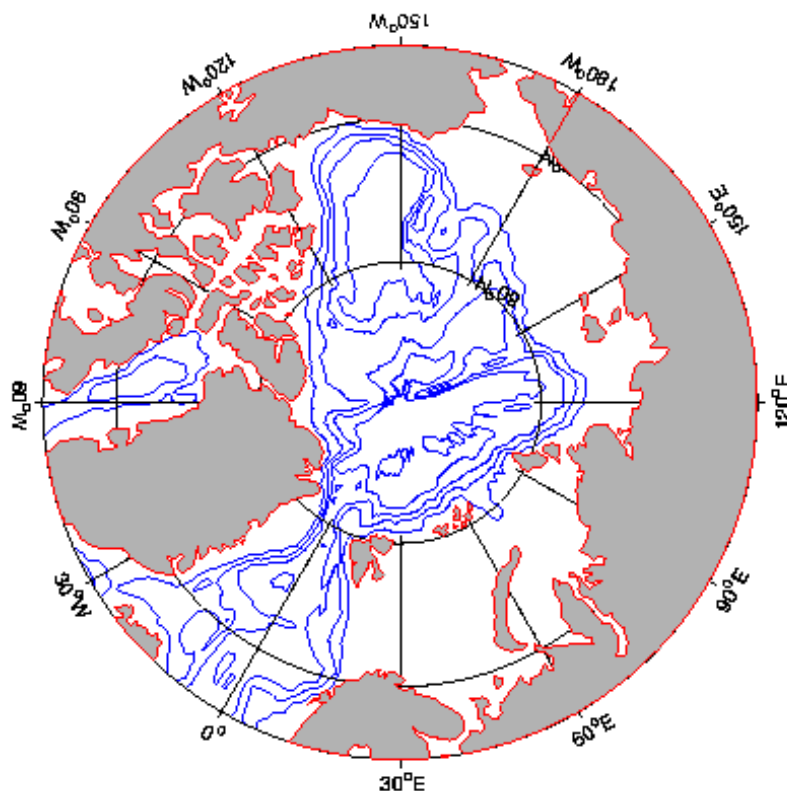
```
m_proj('lambert','long',[-160 -40],'lat',[30 80]);
m_coast('patch',[1 .85 .7]);
m_elev('contourf',[500:500:6000]);
m_grid('box','fancy','tickdir','in');
colormap(flipud(copper));
```



3 北极地区赤平投影.

% 注意到海岸线是在网格线之上
 % 因为是根据两个路径调用的顺序

```
m_proj('stereographic','lat',90,'long',30,'radius',25);
m_elev('contour',[-3500:1000:-500],'edgecolor','b');
m_grid('xtick',12,'tickdir','out','ytick',[70 80],'linest','-');
m_coast('patch',[.7 .7 .7],'edgecolor','r');
```



4. 全球海洋两种独立投影展示

```
subplot(211);
Slongs=[-100 0;-75 25;-5 45; 25 145;45 100;145 295;100 290];
Slats= [ 8 80;-80 8; 8 80;-80 8; 8 80;-80 0; 0 80];
for l=1:7,
    m_proj('sinusoidal','long',Slongs(l,:), 'lat',Slats(l,:));
    m_grid('fontsize',6,'xticklabels',[], 'xtick',[-180:30:360],...
          'ytick',[-80:20:80], 'yticklabels',[], 'linest', '-
', 'color',[.9 .9 .9]);
    m_coast('patch','g');
end;
xlabel('Interrupted Sinusoidal Projection of World Oceans');;
% 为了看见全部地图, 我们必须撤销m_grid调用的坐标轴限制:
set(gca, 'xlimmode', 'auto', 'ylimmode', 'auto');

subplot(212);
Slongs=[-100 43;-75 20; 20 145;43 100;145 295;100 295];
Slats= [ 0 90;-90 0;-90 0; 0 90;-90 0; 0 90];
for l=1:6,
    m_proj('mollweide','long',Slongs(l,:), 'lat',Slats(l,:));
    m_grid('fontsize',6,'xticklabels',[], 'xtick',[-180:30:360],...
```

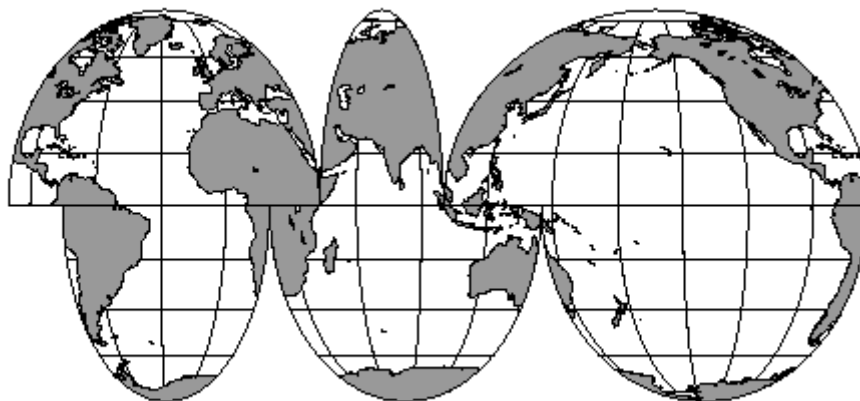
```

        'ytick', [-80:20:80], 'yticklabels', [], 'linest', '-
        ', 'color', 'k');
    m_coast('patch', [.6 .6 .6]);
end;
xlabel('Interrupted Mollweide Projection of World Oceans');
set(gca, 'xlimmode', 'auto', 'ylimmode', 'auto');

```



Interrupted Sinusoidal Projection of World Oceans



Interrupted Mollweide Projection of World Oceans

5. 斜麦卡托投影下的矢量箭头图和等值线图

%% 好看的数据

```

[lon,lat]=meshgrid([-136:2:-114],[36:2:54]);
u=sin(lat/6);
v=sin(lon/6);

```

```

m_proj('oblique','lat',[56 30],'lon',[-132 -120],'aspect',.8);

```

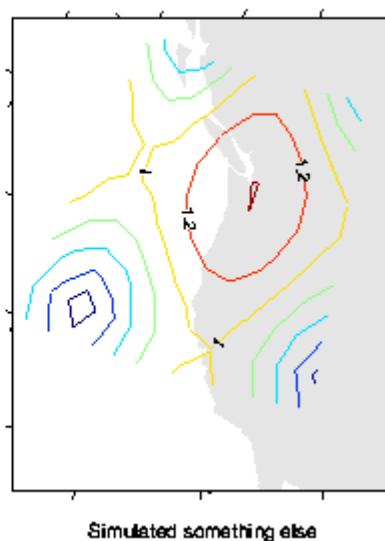
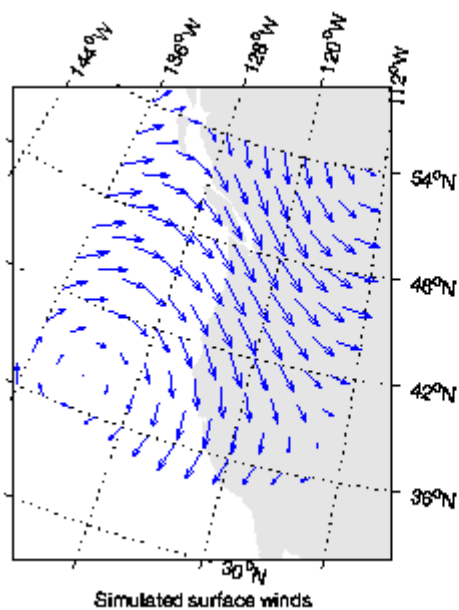


```

subplot(121);
m_coast('patch',[.9 .9 .9],'edgecolor','none');
m_grid('tickdir','out','yaxislocation','right',...
       'xaxislocation','top','xlabelldir','end','ticklen',.02);
hold on;
m_quiver(lon,lat,u,v);
xlabel('Simulated surface winds');

subplot(122);
m_coast('patch',[.9 .9 .9],'edgecolor','none');
m_grid('tickdir','out','yticklabels',[],...
       'xticklabels',[],'linestyle','none','ticklen',.02);
hold on;
[cs,h]=m_contour(lon,lat,sqrt(u.*u+v.*v));
clabel(cs,h,'fontsize',8);
xlabel('Simulated something else');

```



6. 米勒投影下的大圆

%绘制一个圆形轨迹

```

lon=[-180:180];
lat=atan(tan(60*pi/180)*cos((lon-30)*pi/180))*180/pi;

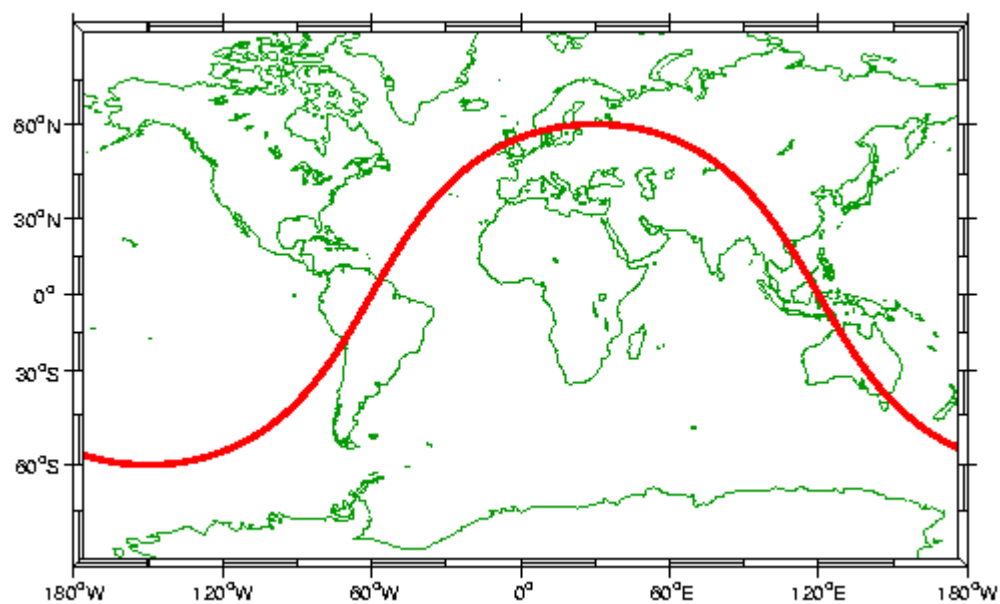
```

```

m_proj('miller','lat',82);
m_coast('color',[0 .6 0]);
m_line(lon,lat,'linewi',3,'color','r');

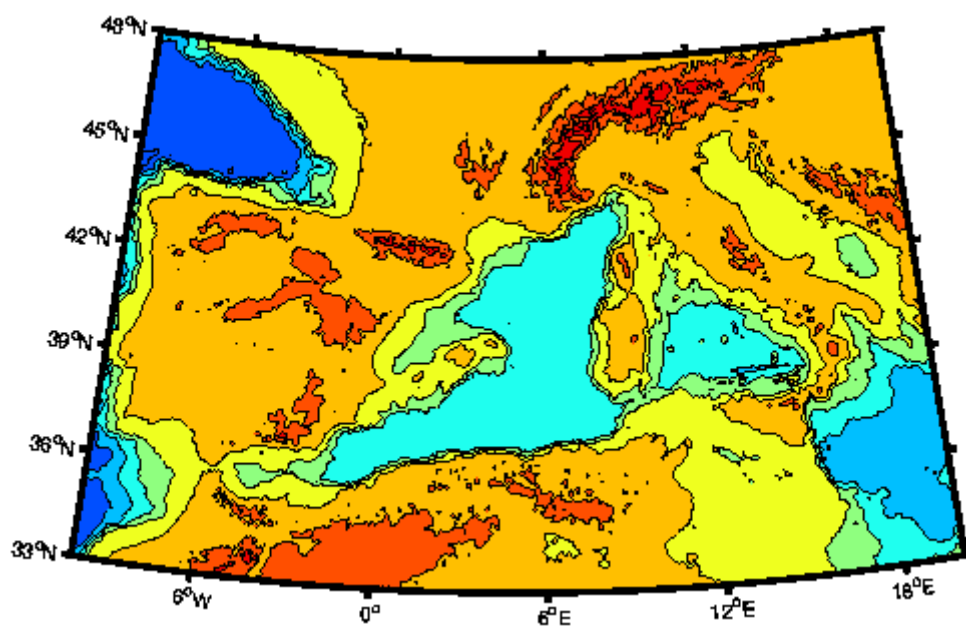
```

```
m_grid('linestyle','none','box','fancy','tickdir','out');
```



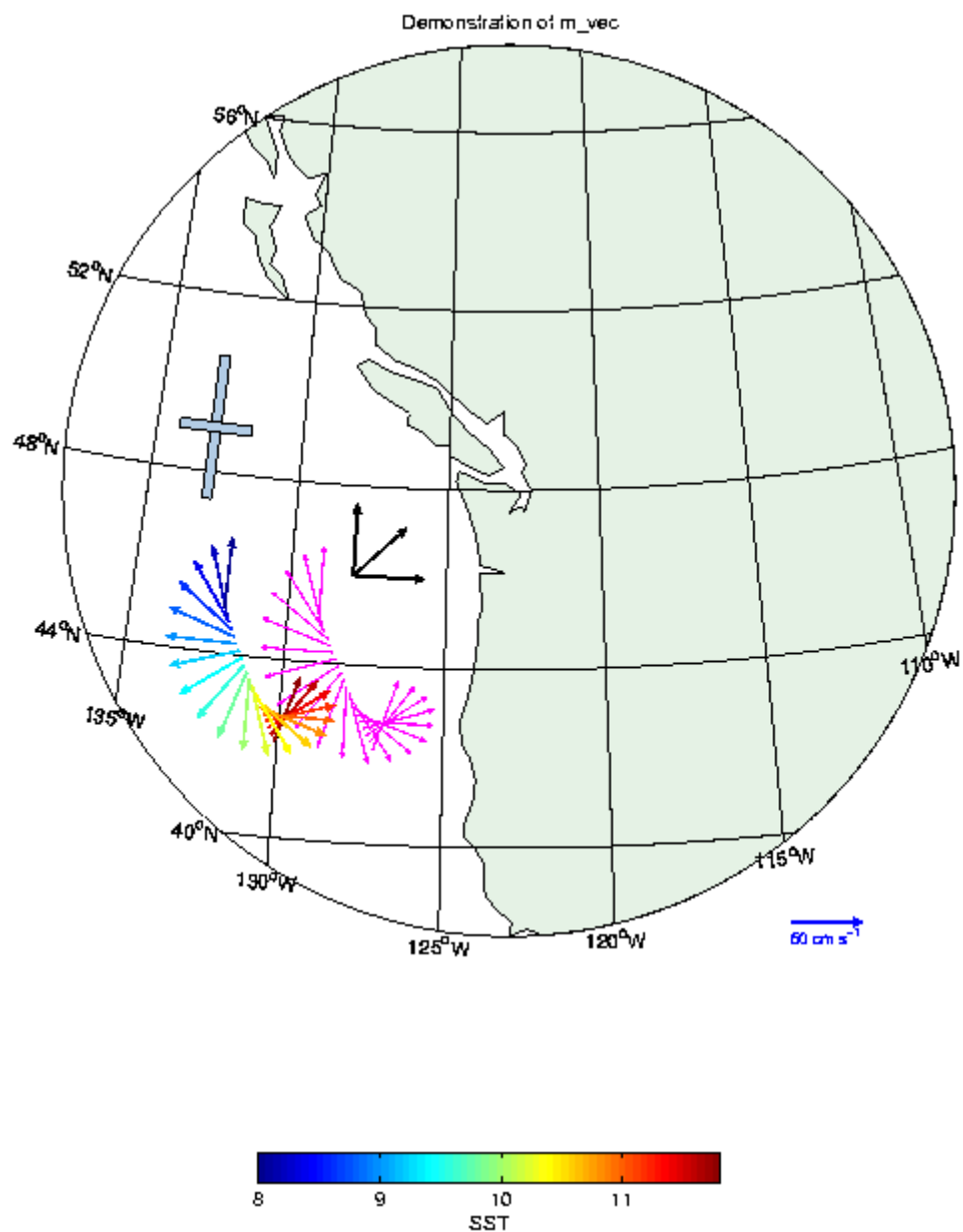
7. 兰伯托投影下的西地中海高分辨率水深

```
m_proj('lambert','lon',[-10 20],'lat',[33 48]);
m_tbase('contourf');
m_grid('linestyle','none','tickdir','out','linewidth',3);
```



8. 矢量展示方式

`m_vec` % 详见 `m_vec.m` 的代码



9. 在爱德华王子岛上展示不同分辨率的海岸线

%例子展示了默认的海岸线和爱德华王子岛部分放大
% 的所有不同分辨率GSHHS海岸线。

```
clf
axes('position',[.35 .6 .37 .37]);
```

```

m_proj('albers equal-area', 'lat', [40 60], 'long', [-90 -
50], 'rect', 'on');
m_coast('patch', [0 1 0]);
m_grid('linest', 'none', 'linewidth', 2, 'tickdir', 'out', 'xaxisloc', 'top'
, 'yaxisloc', 'right');
m_text(-69, 41, 'Standard coastline', 'color', 'r', 'fontweight', 'bold');

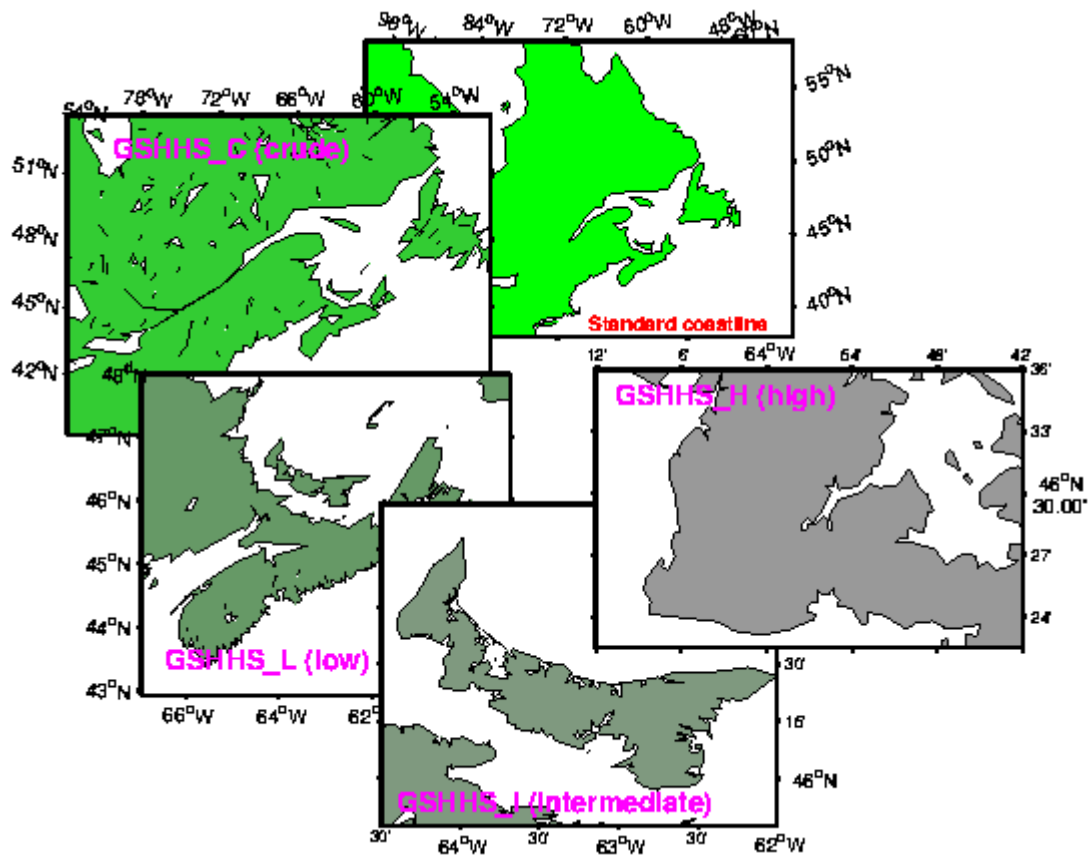
axes('position', [.09 .5 .37 .37]);
m_proj('albers equal-area', 'lat', [40 54], 'long', [-80 -
55], 'rect', 'on');
m_gshhs_c('patch', [.2 .8 .2]);
m_grid('linest', 'none', 'linewidth', 2, 'tickdir', 'out', 'xaxisloc', 'top'
);
m_text(-80, 52.5, 'GSHHS\_C
(crude)', 'color', 'm', 'fontweight', 'bold', 'fontsize', 14);

axes('position', [.13 .2 .37 .37]);
m_proj('albers equal-area', 'lat', [43 48], 'long', [-67 -
59], 'rect', 'on');
m_gshhs_l('patch', [.4 .6 .4]);
m_grid('linest', 'none', 'linewidth', 2, 'tickdir', 'out');
m_text(-66.5, 43.5, 'GSHHS\_L
(low)', 'color', 'm', 'fontweight', 'bold', 'fontsize', 14);

axes('position', [.35 .05 .37 .37]);
m_proj('albers equal-area', 'lat', [45.8 47.2], 'long', [-64.5 -
62], 'rect', 'on');
m_gshhs_i('patch', [.5 .6 .5]);
m_grid('linest', 'none', 'linewidth', 2, 'tickdir', 'out', 'yaxisloc', 'righ
t');
m_text(-64.4, 45.9, 'GSHHS\_I
(intermediate)', 'color', 'm', 'fontweight', 'bold', 'fontsize', 14);

axes('position', [.55 .23 .37 .37]);
m_proj('albers equal-area', 'lat', [46.375 46.6], 'long', [-64.2 -
63.7], 'rect', 'on');
m_gshhs_h('patch', [.6 .6 .6]);
m_grid('linest', 'none', 'linewidth', 2, 'tickdir', 'out', 'xaxisloc', 'top'
, 'yaxisloc', 'right');
m_text(-64.18, 46.58, 'GSHHS\_H
(high)', 'color', 'm', 'fontweight', 'bold', 'fontsize', 14);

```



10. UTM 投影下的轨迹图

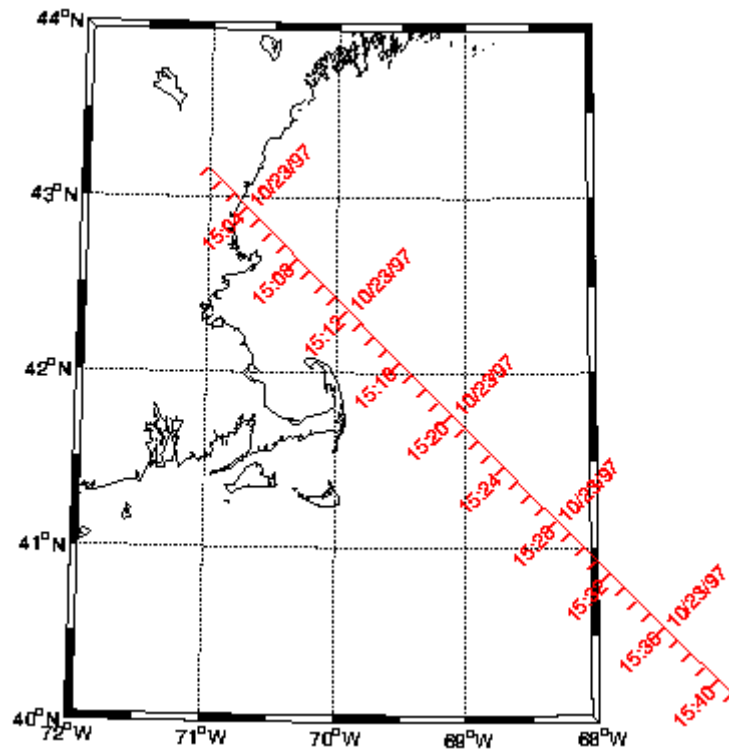
```

m_proj('UTM','long',[-72 -68],'lat',[40 44]);
m_gshhs_i('color','k');
m_grid('box','fancy','tickdir','in');

% 做一个假的轨迹
lons=[-71:.1:-67];
lats=60*cos((lons+115)*pi/180);
dates=datenum(1997,10,23,15,1:41,zeros(1,41));

m_track(lons,lats,dates,'ticks',0,'times',4,'dates',8,...
        'clip','off','color','r','orient','upright');

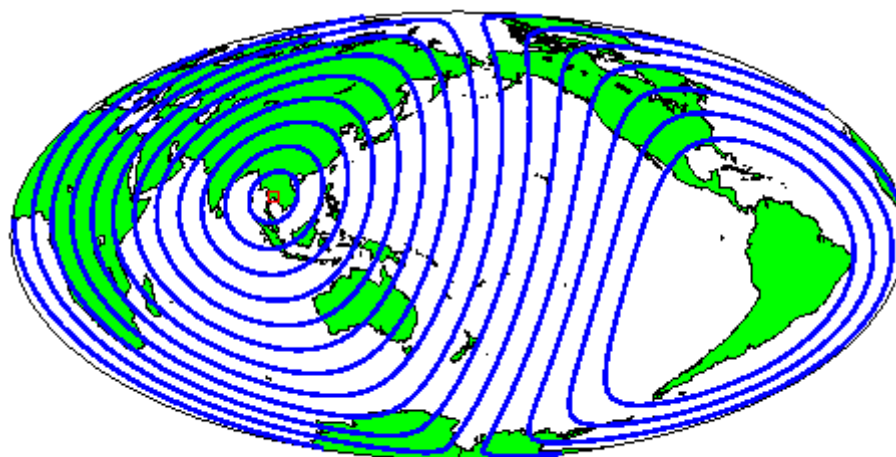
```



11. 等距圆环

```
m_proj('hammer','clong',170);
m_grid('xtick',[],'ytick',[],'linestyle','-');
m_coast('patch','g');
m_line(100.5,13.5,'marker','square','color','r');

m_range_ring(100.5,13.5,[1000:1000:15000],'color','b','linewi',2);
xlabel('1000km range rings from Bangkok');
```



1000km range rings from Bangkok

12. 斑点状边界

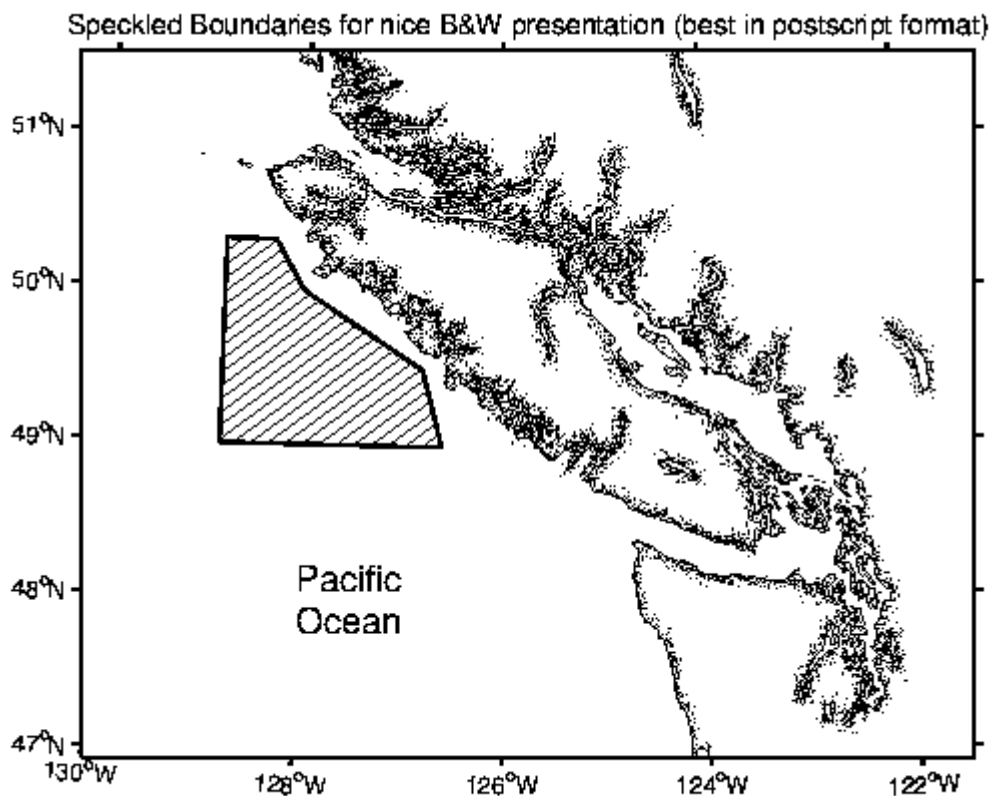
```
bndry_lon=[-128.8 -128.8 -128.3 -128 -126.8 -126.6 -128.8];
bndry_lat=[49      50.33 50.33 50   49.5  49      49];

clf;
m_proj('lambert','long',[-130 -121.5],'lat',[47
51.5],'rectbox','on');

m_gshhs_i('color','k');           % 海岸线...
m_gshhs_i('speckle','color','k'); % 斑点状

m_line(bndry_lon,bndry_lat,'linewi',2,'color','k'); % 区域外
廓 ...
m_hatch(bndry_lon,bndry_lat,'single',30,5,'color','k'); % ..添加阴影.

m_grid('linewi',2,'linest','none','tickdir','out','fontsize',12);
title('Speckled Boundaries for nice B&W presentation (best in
postscript format)','fontsize',14);
m_text(-128,48,5,{'Pacific','Ocean'},'fontsize',18);
```



13. 蓝色海洋

```

m_proj('miller','lat',[-75 75]);

set(gca,'color',[.9 .99 1]);      % 在调用填充之前设置.

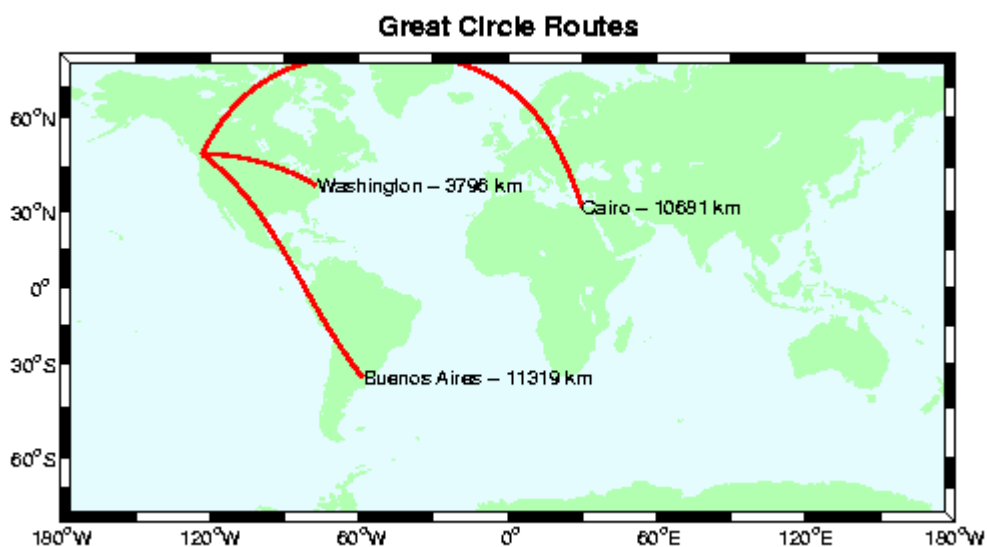
m_coast('patch',[.7 1 .7],'edgecolor','none');
m_grid('box','fancy','linestyle','none');

cities={'Cairo','Washington','Buenos Aires'};
lons=[ 30+2/60 -77-2/60 -58-22/60];
lats=[ 31+21/60 38+53/60 -34-45/60];

for k=1:3,
    [range,ln,lt]=m_lldist([-123-6/60 lons(k)],[49+13/60
lats(k)],40);
    m_line(ln,lt,'color','r','linewidth',2);
    m_text(ln(end),lt(end),sprintf('%s - %d
km',cities{k},round(range)));
end;

title('Great Circle Routes','fontsize',14,'fontweight','bold');

```

卫星资料处理示例

1. 全球 SST（或者任何一种全球经纬度格点资料）

```
% NOAA/NASA的AVHRR SST产品路径
% http://podaac.jpl.nasa.gov/sst/

[P,map]=imread('../m_mapWK/199911h54ma-gdm.hdf');

%54km的数据集文档给出了
% 以下温度计算公式
P=0.15*double(P)-3; % deg C

%...and defines this Lat/Long grid for the data
Plat=90-.25-[0:359]*.5;Plon=-180+.25+[0:719]*.5;

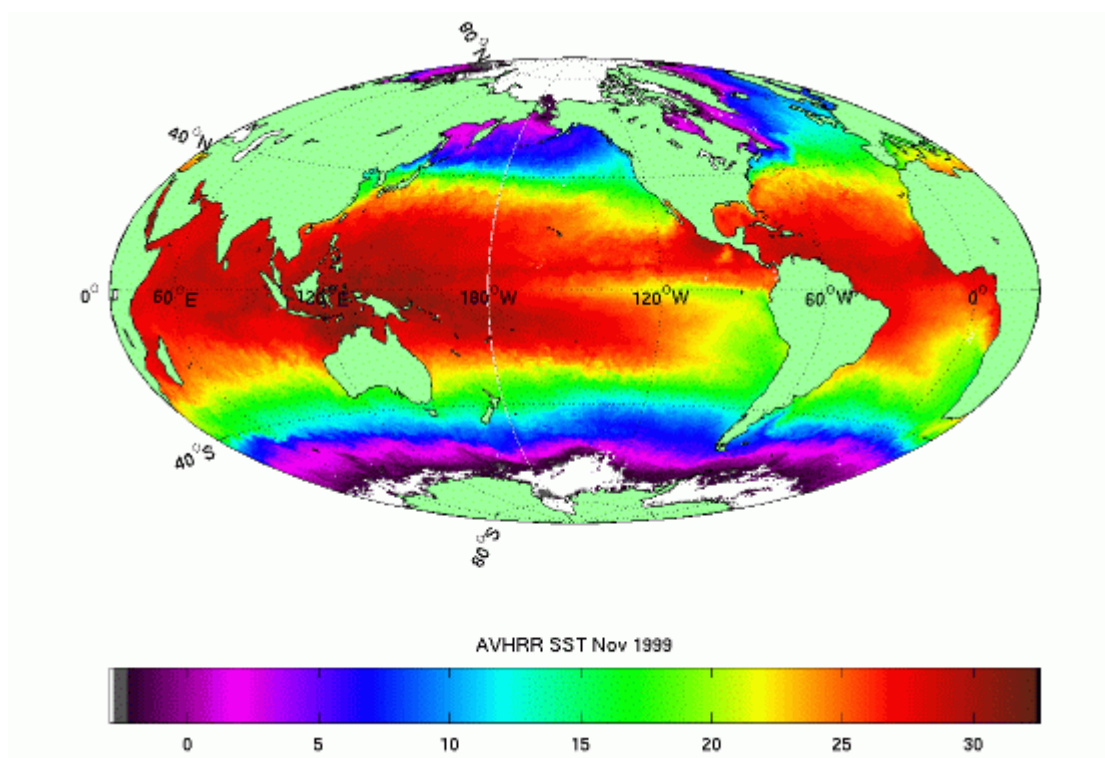
% 由于网格是经纬度矩形坐标
% （即不是真的投影，尽管在m_map
% “等距圆柱投影” 名下）
% 因此不会使用“图像”技术。与此相反，
% 建立网格，为了pcolor平面移动半个网格间距
[Plg,Plt]=meshgrid(Plon-0.25,Plat+0.25);

m_proj('hammer-aitoff','clongitude',-150);
```

```
% 比起将数据按照图形限制重新排列
% 我只是画了两次（你可以在180W看到拼接
% 因为pcolor平面的奇葩之处）（注意到
% 所有全球投影360度都有不对之处）
m_pcolor(Plg,Plt,P);shading flat;colormap(map);
hold on;
m_pcolor(Plg-360,Plt,P);shading flat;colormap(map);

m_coast('patch',[.6 1 .6]);
m_grid('xaxis','middle');

% 添加标准色标.
h=colorbar('h');
set(get(h,'title'),'string','AVHRR SST Nov 1999');
```



2. SSM/I 冰覆盖（修正网格资料）

```
% SSM/I Ice 覆盖网格
% （美国冰雪数据中心）
```

```
P=hdfread('/mnt/cdrom/nasateam/northern/1991/feb/910222.tot','8-bit
Raster Image #2');
P(P==168)=101; % 没有覆盖的陆地.
P(P==157)=102; % 错误数据

% SSM/I冰产品使用赤平投影
% 并且给出了网格的边缘。这里只是
% 使用了给出的边缘并且“假设”一切安好。
% 尽管实际上投影使用的是一个椭球地球
% (m_map使用球形地球)，但是这也不坏。

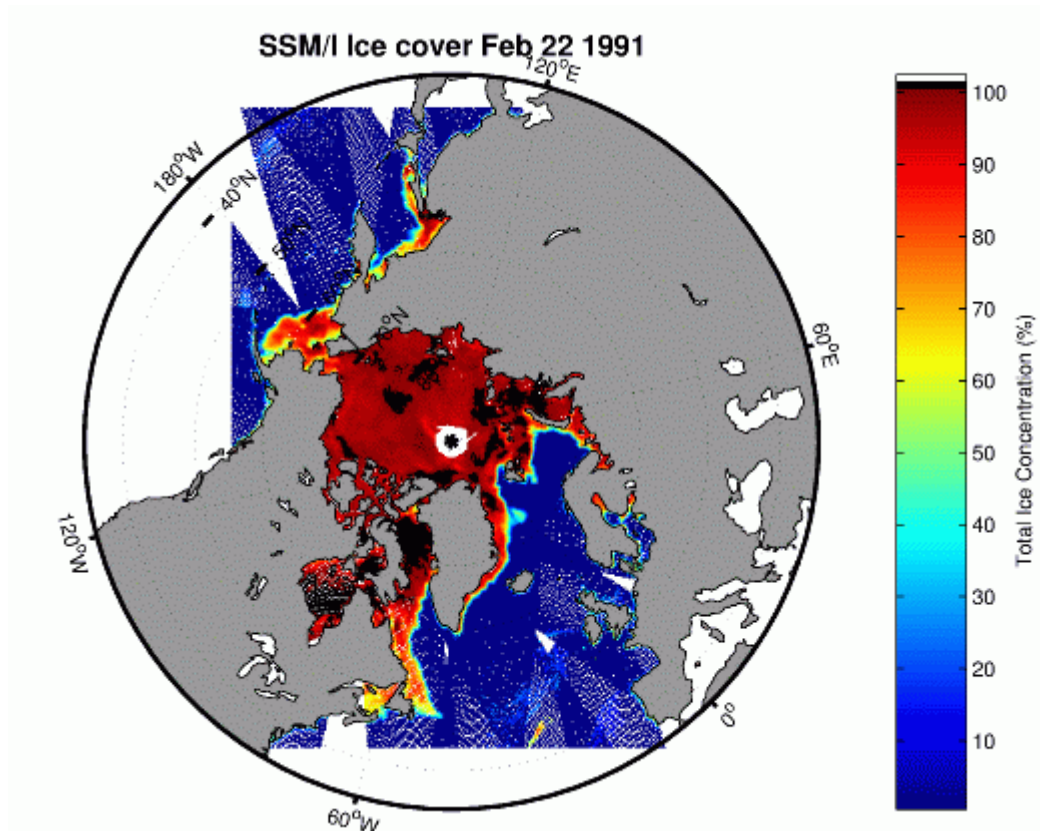
m_proj('stereographic','latitude',90,'radius',55,'rotangle',45);

% 将底部和左侧的边缘变换成屏幕坐标。
% 这当然是一个组装货。
[MAPX,dm]=m_ll2xy([279.26 350.03],[33.92 34.35],'clip','off');
[dm,MAPY]=m_ll2xy([168.35 279.26],[30.98 33.92],'clip','off');

clf
%将数据绘制成图
image(MAPX,MAPY,P);set(gca,'ydir','normal');
colormap([jet(100);0 0 0;1 1 1]);

m_coast('patch',[.6 .6 .6]);
m_grid('linewi',2,'tickdir','out');
title('SSM/I Ice cover Feb 22
1991','fontsize',14,'fontweight','bold');

h=colorbar('v');
set(get(h,'ylabel'),'string','Total Ice Concentration (%)');
```

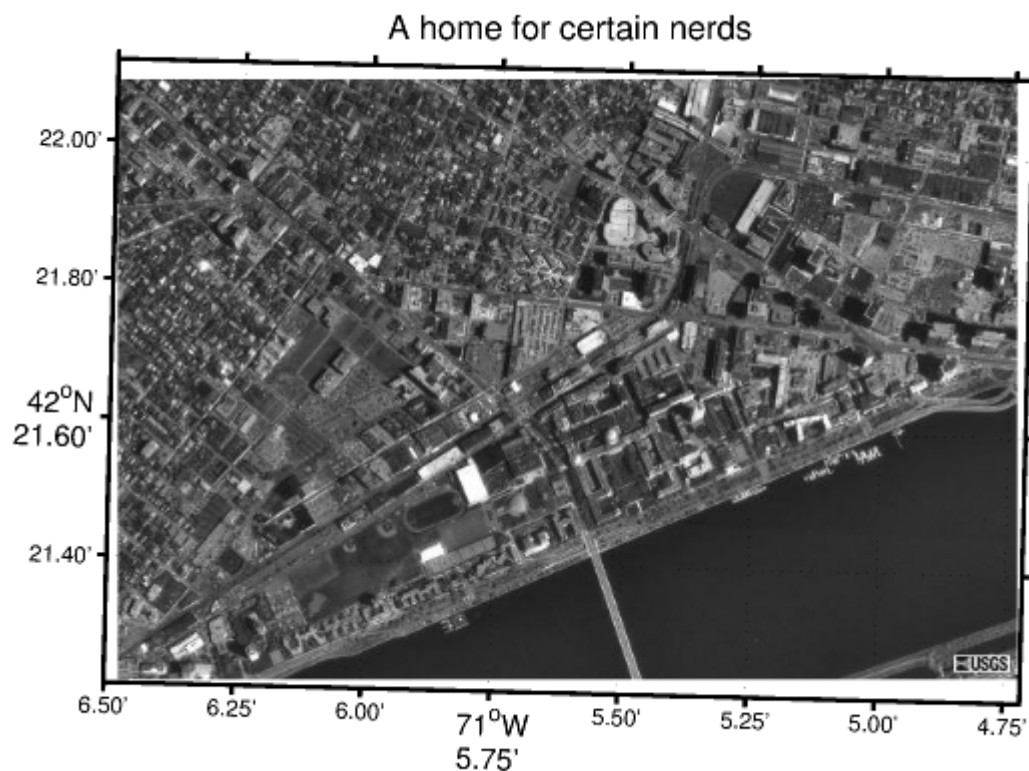


3. UTM 网格下的航拍图

```
% 这是一张来自TerraServer的图像
% (http://terraserver.microsoft.com/)
% 并且已经被转换成UTM坐标。UTM投影椭圆变量
% 在屏幕上使用UTM坐标（也就是
% 被设置成不再是默认值）。
[P,map]=imread('../m_mapWK/oncehome.jpeg');

% 将投影范围设置成图像边缘的经纬度。
m_proj('UTM','long',[-71-6/60-30/3600 -71-4/60-43/3600],...
        'lat',[42+21/60+13/3600
        42+22/60+7/3600],'ellipse','wgs84');

clf;
image([326400 328800],[4692800 4691200],P);set(gca,'ydir','normal');
m_grid('tickdir','out','linewi',2,'fontsize',14);
title('A home for certain nerds','fontsize',16);
```



4. 一个全球数据集的子集（HDF 格式）

% 来自 <http://seawifs.gsfc.nasa.gov/SEAWIFS.html> 的水色信息
 % 获取 4km 周平均数据集并且绘制乔治亚海峡和外海岸线的地图
 % 注意到大多数代码都用来读取和设置数据。

```
LATLIMS=[47 51];
LONLIMS=[-130 -121];
```

% 注-这大概不是最方便的读取和处理 HDF 数据

% 的方法，但是我这件事情不经常做。.

%

% 首先，得到属性数据

```
PI=hdfinfo('A20040972004104.L3m_8D_CHLO_4KM');
```

% 并且将其写成数据结构

```
pin=[];
```

```
for k=1:59,
```

```
    nm=PI.Attributes(k).Name; nm(nm==' ')='_';
```

```
    if isstr(PI.Attributes(k).Value),
```

```
        pin=setfield(pin,nm,PI.Attributes(k).Value);
```

```

else
    pin=setfield(pin,nm,double(PI.Attributes(k).Value));
end
end;

%网格边角的经纬度
lon=[pin.Westernmost_Longitude:pin.Longitude_Step:pin.Easternmost_Longitude];
lat=[pin.Northernmost_Latitude:-
pin.Latitude_Step:pin.Southernmost_Latitude];

% 获取感兴趣地区的经纬度
[mn,ilt]=min(abs(lat-max(LATLIMS)));
[mn,ilg]=min(abs(lon-min(LONLIMS)));
ltlm=fix(diff(LATLIMS)/pin.Latitude_Step);
lglm=fix(diff(LONLIMS)/pin.Longitude_Step);

% 载入所需地图范围的数据子集
P=hdfread('A20040972004104.L3m_8D_CHLO_4KM','l3m_data','Index',{[ilt
ilg],[],[ltlm lglm]});

% 使用给出的公式将数据转换成log(Chla)。没数据留空。
P=double(P);
P(P==255)=NaN;
P=(pin.Slope*P+pin.Intercept); % chla的以10为底的对数

LT=lat(ilt+[0:ltlm-1]);LG=lon(ilg+[0:lglm-1]);
[Plg,Plt]=meshgrid(LG,LT);

% 画地图...

clf;
m_proj('lambert','lon',LONLIMS,'lat',LATLIMS);
m_pcolor(Plg,Plt,P);shading flat;
m_gshhs_i('color','k');;
m_grid('linewi',2,'tickdir','out');;
h=colorbar;
set(get(h,'ylabel'),'String','Chla (\mug/l)');
set(h,'ytick',log10([.5 1 2 3 5 10 20 30]),'yticklabel',[.5 1 2 3 5
10 20 30],'tickdir','out');
title(['MODIS Chla '
datestr(datum(pin.Period_Start_Year,1,0)+pin.Period_Start_Day) ' ->
' ...

```

```
datestr(datetime(pin.Period_Start_Year,1,0)+pin.Period_End_Day)],...  
    'fontsize',14,'fontweight','bold');
```

