

地图投影免考作业——等差分纬线多圆锥投影

我们之前用 matlab 做的地图投影都是用 matlab 的 mapping toolbox 中自带的投影函数。等差分纬线多圆锥投影在 mapping toolbox 中没有，所以要建立自定义投影的话，需要对 matlab 的 mapping toolbox 代码进行修改。通过我自己的探索和参考前一届同学的博客，我了解了自定义投影的方法。只需要在 mapproj 中添加投影函数，再在 maplist 中添加投影。

(有可能出现调用不了自己定的投影的情况，这时需要把工作文件夹设在 mapproj) 下一步，便是学习等差分纬线多圆锥投影的坐标变换方法了。

等差分纬线多圆锥投影是一个难度比较大的投影。它的纬线表现为同心圆圆弧，投影基本公式与我们之前认识的圆锥投影公式有一定区别。而且坐标转换时需要选择中央经线和边缘经线上一系列的参考点进行坐标变换参数的计算。这些都给坐标转换带来了一定的麻烦。下面，我将分步对坐标转换方法进行说明。

第一步，是选定一系列的经线参考点。利用书本上 P160 和 P162 提供的表格，可以进行参考点选定。为了表示世界地图范围，我选择了经差为 180 度上经线的坐标数据。如下图所示。

表12-1是该投影的直角坐标值。

表 12-1

$\Delta\lambda$	x	y	0°	30°	60°	90°	120°	150°	180°	210°	240°
0°	x	y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
			0.006	29.792	58.602	86.625	113.667	139.792	165.000	189.292	212.587
30°	x	y	33.339	33.812	35.171	37.332	40.211	43.725	47.796	52.348	57.388
			0.000	27.626	54.373	80.214	105.130	129.104	152.128	174.194	195.289
60°	x	y	67.488	68.081	69.786	72.492	76.092	80.479	85.551	91.308	97.853
			0.000	22.592	44.432	65.466	85.707	104.993	123.432	140.971	157.565
90°	x	y	103.257	103.477	104.111	105.119	106.461	108.100	110.000	112.125	114.480
			0.000	14.960	29.448	43.452	56.966	69.983	82.500	94.513	106.020

比例尺 1:10 000 000

表12-2是该投影的纬圆半径 ρ 和极角 ψ 。

(单位: 厘米)

对于中央经线，书本上直接给出了坐标变换公式，如下图所示：

该投影的基本公式：

中央经线上的 x_0 的函数式

$$x_0 = q = \frac{(0.9953537 \varphi + 0.01476138 \varphi^3) R}{\mu_0}$$

设 $\mu_0 = 10\,000\,000$, $R = 6\,371\,116$ 米, φ 以弧度计

代入上式 $x_0 = q = 63.41514\varphi + 0.9404646\varphi^3$

$$y_0 = 0$$

式中 x_0 、 y_0 以厘米为单位。

下一步则是把 180 度经线的坐标参考点带入多项式求多项式系数, 求出 x_n , y_n 关于纬度的变化方程式, 这里采用三次拟合:

$$x_n = K_{x_1} + K_{x_2}\varphi + K_{x_3}\varphi^2 \quad (5)$$

$$y_n = K_{y_1} + K_{y_2}\varphi + K_{y_3}\varphi^2 \quad (6)$$

这里, 可以使用 matlab 的矩阵运算, 即 `a\b` 的方式, 快速地求出多项式系数:

```
X [100.8721;-18.3126]
Y [-9.4729;-28.8595]
```

把方程写进函数里

```
%定义基本方程和常数
xn=100.8721.*lat-18.3126.*lat.^2;
yn=165-9.4729.*lat-28.8595.*lat.^2;
```

基本参数和中央经线的方程也可以写进去了

```
lonm=pi;
b=1.1;
c=0.0005050505;
```

```
%中央经线x0函数式
x0=63.41514.*lat+0.9404646.*lat.^3;
```

有了以上基本公式和参数, 就可以开始坐标变换了。

首先是求各纬线的动半径;

然后是右边经线与各纬度线交点的动径角 (极角)

3) 求解各纬线的动半径 ρ 的公式为

$$\rho = \frac{(y_n - y_0)^2 + (x_n - x_0)^2}{2(x_n - x_0)} \quad (7)$$

4) 求解右边纬线与各纬度线交点的动径角 δ_n

$$\delta_n = \arcsin \frac{y_n - y_0}{|\rho|} \quad (8)$$

再求解投影后的动径角:

5)求解投影之后的动径角 δ 公式

$$\delta = \frac{\delta_n}{180}(b - c|\lambda - 150|)(\lambda - 150) \quad (9)$$

matlab 中代码如下:

```
%计算各纬线的动半径p 中央经线,右边经线与各纬度线交点的动径角an 投影之后的动径角ai  
p=(yn.^2+(xn-x0).^2).^2./(xn-x0);  
an=asin(yn./abs(p));  
ai=an./lonn.*b.*(1-c.*lon).*lon;
```

之后,就可以将极(轴)坐标形式转换为平面直角坐标,参考书上的多圆锥投影基本公式,输入代码如下:

```
q=p+x0;  
y=q-p.*cos(ai);  
x=abs(p).*sin(ai);
```

现在,投影的基本换算公式都求得了,开始考虑投影中的特殊情况。由于在赤道区域,投影的动径角为无穷大,上面的公式无法使用。书本上提供了赤道区域的坐标转换方程:

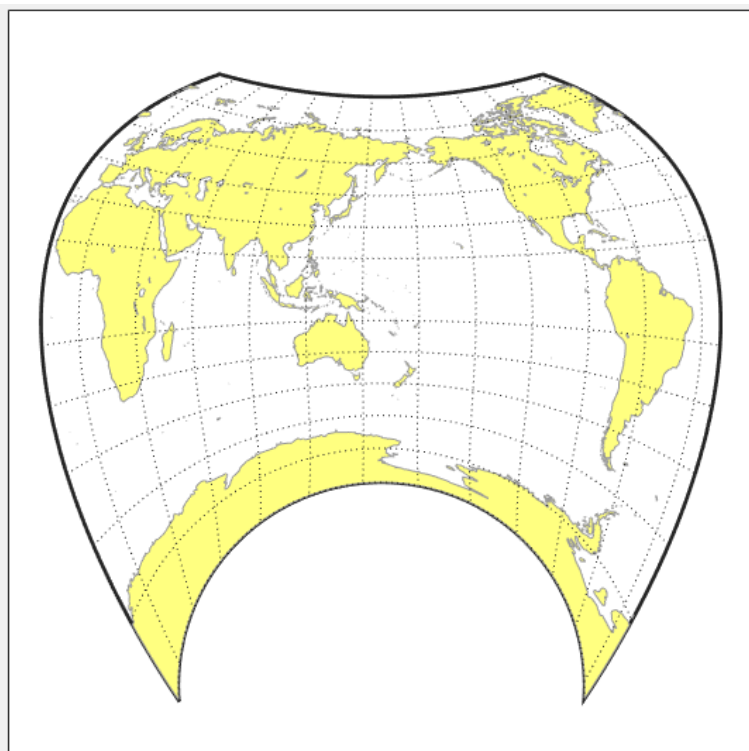
赤道上的直角坐标值

$$y_i = \frac{y_n}{\lambda_n} b(1 - C\lambda_i)\lambda_i$$
$$x_i = 0$$

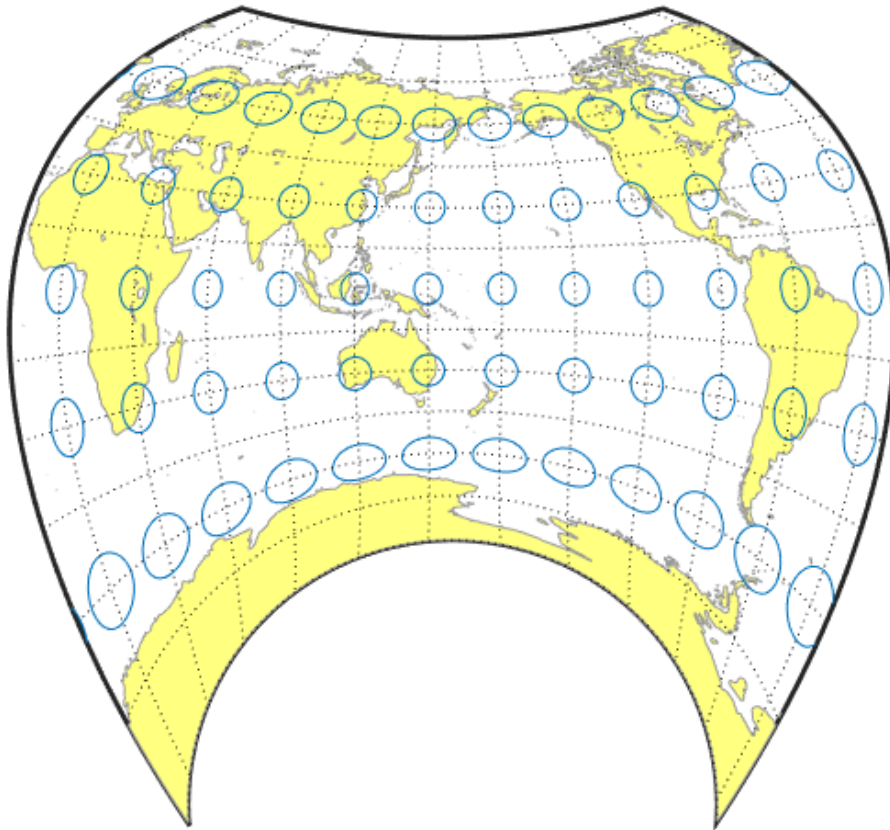
%计算赤道的特殊情况

然后写入 matlab: `y1=yn./lonn.*b.*(1-c.*lon).*lon;`

现在,投影坐标变换公式就完成了,以 160 度为中央经线,查看效果如下。



现在考虑投影的变形分析,我使用变形椭圆查看投影的变形情况,效果如下。可以看出,投影的性质接近等面积的任意投影。我国处在的区域面积变形、角度变形相对较小。由于我选定参考点 (180, 0) (180, 30) (180, 60) 均属赤道以北,在赤道以北地区投影变形较赤道以南小。而且,这也可以解释南极地区投影变形较大的原因。即拟合的坐标变换曲线 $x_n=f(lat), y_n=f(lat)$ 在南极区域误差较大。要减少这种误差,或者为了着重显示赤道以南的地区,可以进行四次拟合,或者多次拟合;也可以在赤道以南地区选定参考点。



备注:

在 maplist 中添加的代码:

```
i=i+1;  
list(i).IdString      = 'dengchafenweixian';  
list(i).Classification = 'PolyConic';  
list(i).Name          = 'dengchafenweixian*';  
list(i).ClassCode     = 'Poly';
```

显示地图投影的代码 (中央经线为 160 度):

```
landareas=shaperead('landareas.shp','UseGeoCoords',true);  
axesm('dengchafenweixian','Frame','on','Grid','on','origin',[0 160 0]);  
geoshow(landareas,'FaceColor',[1 1 .5],'EdgeColor',[.6 .6 .6]);  
tissot
```