**数值天气预报**

**——第二次上机作业**

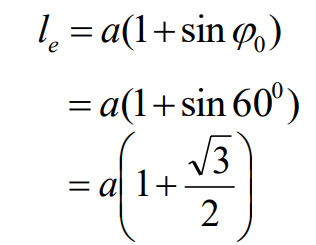
**吴佳根**

**201830196**

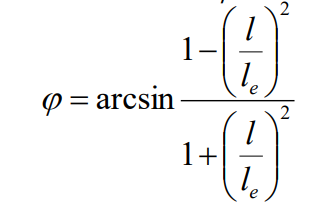
**大气科学学院**

**问题：**

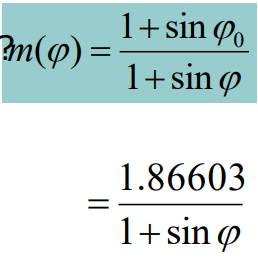
1. 与60°N相割的极射赤面投影映像平面上制作正方形网格，网格距d=500km，P(In=-4,Jn=8)，求出P点和网格中心点的极距、地图放大系数、科氏参数、纬度。
2. 与30°N和60°N相割的Lambert投影映像平面上制作正方形网格，网格距d=300km，P(In=5,Jn=15)，求出P点和网格中心点的极距、地图放大系数、科氏参数、纬度。
3. 与22.5°N和22.5°S相割的Mercator投影映像平面上制作正方形网格，网格距d=200km，P(Je=3)，求出P点和网格中心点的地图放大系数、科氏参数、纬度。
4. 作图展示不同投影类型下地图放大系数m和极距l随纬度的变化。
5. **极射赤面投影法**
6. 根据图片和网格距，求出P点和网格中心点的极距
7. 由地球半径和标准纬度，求出赤道处l的值



1. 由P点和网格中心点的极距和leq 求出P点和网格中心点的纬度



1. 由求出m



1. 由求出f
2. 将由弧度制转换为角度制

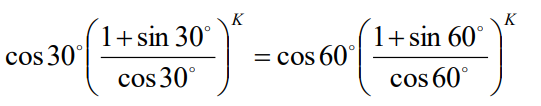
综上，得到结果如下：

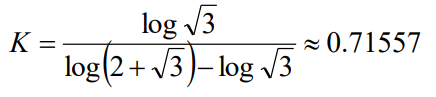
|  |  |  |
| --- | --- | --- |
|  | P点 | 网格中心点 |
| l极距（单位：km） | 4472.14 | 2500 |
| m地图放大系数 | 1.065 | 0.974 |
| f Coriolis参数 | 1.098\*10-4 | 1.336\*10-4 |
| 纬度（单位：°） | 48.77 | 66.25 |

1. **投影法**

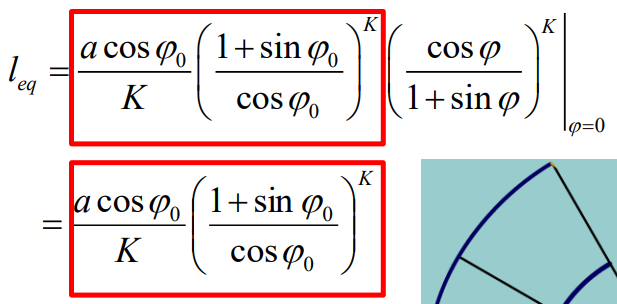
1）根据图片和网格距，求出P点和网格中心点的极距

2）由标准纬度取30°、60°，求得圆锥常数K



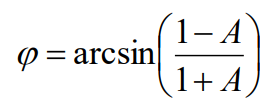


3）由圆锥常数K、地球半径和标准纬度，求出赤道处l的值

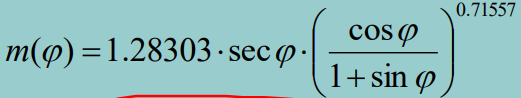


4）由圆锥常数K、P点和网格中心点的极距和leq 求出P点和网格中心点的纬度

令



5) 由求出m



6) 由求出f

1. 将由弧度制转换为角度制

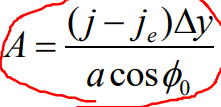
综上，得到结果如下：

|  |  |  |
| --- | --- | --- |
|  | P点 | 网格中心点 |
| l极距（单位：km） | 4743.42 | 3300 |
| m地图放大系数 | 0.9878 | 1.0836 |
| f Coriolis参数 | 1.229\*10-4 | 1.372\*10-4 |
| 纬度（单位：°） | 57.36 | 70.00 |

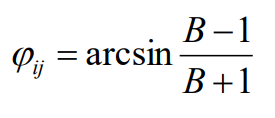
1. **墨卡托投影法**

1）根据图片和网格距，求出P点和网格中心点的l（与前面的极距不同，此处指离赤道面的距离）

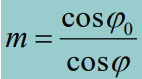
2）由P点和网格中心点的离赤道面的距离求出P点和网格中心点的纬度

记

记



3）由求出m

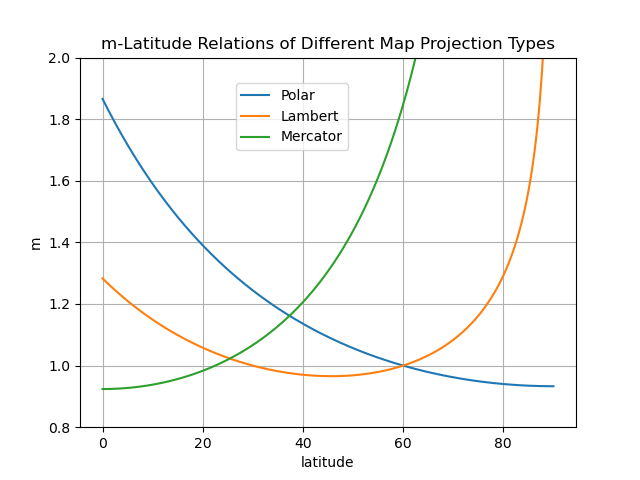


4）由求出f

1. 将由弧度制转换为角度制
2. 综上，得到结果如下：

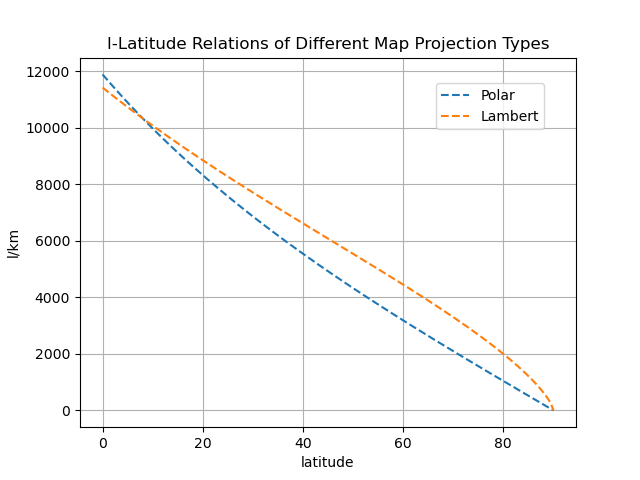
|  |  |  |
| --- | --- | --- |
|  | P点 | 网格中心点 |
| l极距（单位：km） | 无 | 无 |
| m地图放大系数 | 0.9287 | 0.9239 |
| f Coriolis参数 | 1.4831\*10-5 | 0 |
| 纬度（单位：°） | 5.83 | 0 |

1. **不同投影类型下地图放大系数m随纬度的变化**



可见：

1. 不同投影坐标系下对应于标准纬度的地图投影放大系数为1
2. 离标准纬度越远，地图投影放大系数与1的差距越大
3. Mercator投影在低纬地区放大系数与1差距较小，适用于低纬地区，但在中高纬地区放大系数较大，不适用于中高纬地区。
4. Lambert投影在中纬地区放大系数与1差距较小，适用于中纬地区，但在低纬、高纬地区放大系数较大，不适用于低纬、高纬地区。
5. 极射赤面投影在高纬地区放大系数与1差距较小，适用于高纬地区，但在中低纬地区放大系数较大，不适用于中低纬地区。
6. **不同投影类型下极距l随纬度的变化**



可见：

1. 极射赤面投影、Lambert投影的极距都随纬度的增加而下降，在90°的位置降为0。
2. 在低纬某一纬度处，极射赤面投影的极距与Lambert投影的极距相等。在纬度小于该纬度时，极射赤面投影的极距大于Lambert投影的极距；在纬度大于该纬度时，Lambert投影的极距大于极射赤面投影。

**总结：**

经过这次上机作业，我练习了在极射赤面投影、Lambert投影、Mercator投影当中计算极距、地图放大系数、科氏参数、纬度，让我对三种投影方式以及其中各种物理量之间的关系理解更加深刻。并且通过自己动手画图，获得了不同投影类型下地图放大系数m和极距l随纬度的变化关系，对我在之后选取投影方式提供了经验。

计算过程代码：

1. Polar

import math  
#%%  
a = 6371 #km  
d = 500 #km  
omg = 7.3e-5  
l\_eq = a\*(1+math.sqrt(3)/2)  
l\_p = math.sqrt((-4)\*(-4)+8\*8)\*d  
l\_m = 5\*d  
fi\_p = math.asin((1-(l\_p/l\_eq)\*(l\_p/l\_eq))/(1+(l\_p/l\_eq)\*(l\_p/l\_eq)))  
fi\_m = math.asin((1-(l\_m/l\_eq)\*(l\_m/l\_eq))/(1+(l\_m/l\_eq)\*(l\_m/l\_eq)))  
m\_p = 1.86603/(1+math.sin(fi\_p))  
m\_m = 1.86603/(1+math.sin(fi\_m))  
f\_p = 2\*omg\*math.sin(fi\_p)  
f\_m = 2\*omg\*math.sin(fi\_m)  
#%%  
print(l\_p,l\_m)  
print(m\_p,m\_m)  
print(f\_p,f\_m)  
print(math.degrees(fi\_p),math.degrees(fi\_m))

（2）Lambert

import math  
#%%  
a = 6371 #km  
d = 300 #km  
omg = 7.3e-5  
k = 0.71557  
l\_p = math.sqrt(5\*5+15\*15)\*d  
l\_m = 11\*d  
l\_eq = a\*1.28303/k  
A\_p = pow(l\_p/l\_eq,2/k)  
A\_m = pow(l\_m/l\_eq,2/k)  
fi\_p = math.asin((1-A\_p)/(1+A\_p))  
fi\_m = math.asin((1-A\_m)/(1+A\_m))  
m\_p = 1.28303/math.cos(fi\_p)\*pow(math.cos(fi\_p)/(1+math.sin(fi\_p)),k)  
m\_m = 1.28303/math.cos(fi\_m)\*pow(math.cos(fi\_m)/(1+math.sin(fi\_m)),k)  
f\_p = 2\*omg\*math.sin(fi\_p)  
f\_m = 2\*omg\*math.sin(fi\_m)  
#%%  
print(l\_p,l\_m)  
print(m\_p,m\_m)  
print(f\_p,f\_m)  
print(math.degrees(fi\_p),math.degrees(fi\_m))

（3）Mercator

import math  
#%%  
a = 6371 #km  
d = 200 #km  
fi0 = math.radians(22.5) #radians  
omg = 7.3e-5  
A\_p = 3\*d/(a\*math.cos(fi0))  
A\_m = 0/(a\*math.cos(fi0))  
B\_p = math.exp(2\*A\_p)  
B\_m = math.exp(2\*A\_m)  
fi\_p = math.asin((B\_p-1)/(B\_p+1)) #radians  
fi\_m = math.asin((B\_m-1)/(B\_m+1)) #radians  
m\_p = math.cos(fi0)/math.cos(fi\_p)  
m\_m = math.cos(fi0)/math.cos(fi\_m)  
f\_p = 2\*omg\*math.sin(fi\_p)  
f\_m = 2\*omg\*math.sin(fi\_m)  
#%%  
print(m\_p,m\_m)  
print(f\_p,f\_m)  
print(math.degrees(fi\_p),math.degrees(fi\_m))

（4）m-latitude

import math  
import numpy as np  
import matplotlib.pyplot as plt  
#%%  
fi = np.linspace(0,90,200)  
m\_polar = np.zeros(200)  
m\_lambert = np.zeros(200)  
m\_mercator = np.zeros(200)  
for i in range(200):  
 m\_polar[i] = 1.86603/(1+math.sin(fi[i]\*math.pi/180))  
 m\_lambert[i] = 1.28303/math.cos(fi[i]\*math.pi/180)\*pow((math.cos(fi[i]\*math.pi/180)/(1+math.sin(fi[i]\*math.pi/180))),0.71557)  
 m\_mercator[i] = 0.92388/math.cos(fi[i]\*math.pi/180)  
#%%  
plt.ylim(0.8,2)  
plt.plot(fi,m\_polar,label='Polar')  
plt.plot(fi,m\_lambert,label='Lambert')  
plt.plot(fi,m\_mercator,label='Mercator')  
plt.xlabel('latitude')  
plt.ylabel('m')  
plt.title('m-Latitude Relations of Different Map Projection Types')  
plt.grid()  
plt.legend(loc='upper left', bbox\_to\_anchor=(0.3, 0.95))  
plt.savefig('m-latitude')

（5）l-latitude

import math  
import numpy as np  
import matplotlib.pyplot as plt  
#%%  
fi = np.linspace(0,90,200)  
fi0 = math.radians(60)  
a = 6371  
k\_lambert = 0.71557  
k\_polar = 1  
l\_eq\_lambert = a\*math.cos(fi0)/k\_lambert\*math.pow((1+math.sin(fi0))/math.cos(fi0),k\_lambert)  
l\_eq\_polar = a\*math.cos(fi0)/k\_polar\*math.pow((1+math.sin(fi0))/math.cos(fi0),k\_polar)  
fi\_radians = np.zeros(200)  
l\_polar = np.zeros(200)  
l\_lambert = np.zeros(200)  
for i in range(200):  
 fi\_radians[i] = fi[i]\*math.pi/180  
 l\_lambert[i] = l\_eq\_lambert\*pow(math.cos(fi\_radians[i])/(1+math.sin(fi\_radians[i])),k\_lambert)  
 l\_polar[i] = l\_eq\_polar\*math.cos(fi\_radians[i])/(1+math.sin(fi\_radians[i]))  
#%%  
plt.plot(fi,l\_polar,label='Polar',linestyle='--')  
plt.plot(fi,l\_lambert,label='Lambert',linestyle='--')  
plt.xlabel('latitude')  
plt.ylabel('l/km')  
plt.title('l-Latitude Relations of Different Map Projection Types')  
plt.grid()  
plt.legend(bbox\_to\_anchor=(0.95, 0.95))  
plt.savefig('l-latitude')