绘制.py

***import*** pandas ***as*** pd

***from*** sklearn**.**linear\_model ***import*** LinearRegression

***import*** numpy ***as*** np

***import*** math

***from*** sympy ***import*** solve, Symbol

***import*** matplotlib**.***pyplot* ***as*** plt

# *定义颜色列表*

colors = ['*r*', '*g*', '*b*', '*c*', '*m*', '*y*']

# *海里到米的转换系数*

nautical\_mile\_to\_meter = *1852*

# *读取 CSV 文件*

data = pd**.*read\_csv***('*111111.csv*', *names*=['*x*','*y*','*z*'], *skiprows*=*1*)

# *将 x, y 列转换为米*

*data*['*x*'] = *data*['*x*'] \* nautical\_mile\_to\_meter

*data*['*y*'] = *data*['*y*'] \* nautical\_mile\_to\_meter

***def*** ***calculate\_and\_plot\_lines***(*alpha*, *length*, *width*, *former\_D*):

    re = []

    sei\_ta = *2* \* math**.***pi* / *3*

    dis = [*0* ***for*** \_ ***in*** ***range***(*300*)]

    x = ***Symbol***('*x*')

    # *计算第一个点的距离*

*dis*[*0*] = ***float***(***solve***(*length*/*2* + *x*/*math***.*cos***(*alpha*)-(*former\_D* - *x*\**math***.*tan***(*alpha*))\* *math***.*sin***(*sei\_ta*)/ *math***.*cos***(*sei\_ta*/*2* + *alpha*), *x*)[*0*])

    re**.*append***(*dis*[*0*])

***def*** ***cal\_W***(*d*, *index*):

        D = former\_D - d \* math**.*tan***(*alpha*)

***if*** index **==** *1*:

***return*** D \* math**.*sin***(*sei\_ta*/*2*) / math**.*cos***(*sei\_ta*/*2* + *alpha*)

***else***:

***return*** D \* math**.*sin***(*sei\_ta*/*2*) / math**.*cos***(*sei\_ta*/*2* - *alpha*)

    ind = *0*

***for*** i ***in*** ***range***(*1*, *300*):

        d = ***Symbol***('*d*')

*dis*[*i*] = ***float***(***solve***(***cal\_W***(*d*, *1*)-(*d* - *dis*[*i*-*1*])/ *math***.*cos***(*alpha*)+***cal\_W***(*dis*[*i*-*1*], *2*)-(*1*/*10*)\*(***cal\_W***(*d*, *1*)+***cal\_W***(*d*, *2*)), *d*)[*0*])

***if*** *dis*[*i*] **>=** length/*2*:

***break***

        re**.*append***(*dis*[*i*])

        ind += *1*

***for*** i ***in*** ***range***(*ind*+*1*):

***print***(*dis*[*i*]/ *math***.*cos***(*alpha*), *end*='')

***return*** ind+*1*, (ind+*1*)\*width, re

***def*** ***fit\_region***(*data*, *x\_min*, *x\_max*, *y\_min*, *y\_max*, *x\_or\_y*, *id*):

    # *筛选出符合条件的数据*

    filtered\_data = *data*[

(*data*['*x*']**>=** *x\_min*)&(*data*['*x*']**<=** *x\_max*)&

(*data*['*y*']**>=** *y\_min*)&(*data*['*y*']**<=** *y\_max*)

]

***if*** x\_or\_y **==** '*x*':

        # *如果 x 是自变量*

        x\_filtered = *filtered\_data*['*x*']**.***values***.*reshape***(-*1*, *1*)

        y\_filtered = *filtered\_data*['*z*']**.***values*

***elif*** x\_or\_y **==** '*y*':

        # *如果 y 是自变量*

        x\_filtered = *filtered\_data*['*y*']**.***values***.*reshape***(-*1*, *1*)

        y\_filtered = *filtered\_data*['*z*']**.***values*

***else***:

***raise*** *ValueError*("*x\_or\_y must be either 'x' or 'y'*")

    y\_filtered = -*filtered\_data*['*z*']**.***values*

    # *使用线性回归模型进行拟合*

    model = ***LinearRegression***()

    model**.*fit***(*x\_filtered*, *y\_filtered*)

    # *计算斜率*

    slope = model**.***coef\_*[*0*]

    # *计算中心点*

    center\_x = np**.*mean***(*x\_filtered*)

    center\_y = model**.*predict***([[*center\_x*]])[*0*]

    # *计算 R^2*

    r\_squared = model**.*score***(*x\_filtered*, *y\_filtered*)

***return*** math**.*atan***(***abs***(*slope*)), -center\_y, model, r\_squared, (***abs***(*x\_max*-*x\_min*), ***abs***(*y\_max*-*y\_min*))

regions = [

    {'*x\_min*': *0* \* nautical\_mile\_to\_meter, '*x\_max*': *4* \* nautical\_mile\_to\_meter, '*y\_min*': *4.5* \* nautical\_mile\_to\_meter, '*y\_max*': *5* \* nautical\_mile\_to\_meter, '*x\_or\_y*': '*y*', '*id*': *1*},

    {'*x\_min*': *0* \* nautical\_mile\_to\_meter, '*x\_max*': *2.5* \* nautical\_mile\_to\_meter, '*y\_min*': *2.5* \* nautical\_mile\_to\_meter, '*y\_max*': *4.5* \* nautical\_mile\_to\_meter, '*x\_or\_y*': '*y*', '*id*': *2*},

    {'*x\_min*': *2.5* \* nautical\_mile\_to\_meter, '*x\_max*': *4* \* nautical\_mile\_to\_meter, '*y\_min*': *2* \* nautical\_mile\_to\_meter, '*y\_max*': *4.5* \* nautical\_mile\_to\_meter, '*x\_or\_y*': '*x*', '*id*': *3*},

    {'*x\_min*': *0* \* nautical\_mile\_to\_meter, '*x\_max*': *1* \* nautical\_mile\_to\_meter, '*y\_min*': *0* \* nautical\_mile\_to\_meter, '*y\_max*': *2.5* \* nautical\_mile\_to\_meter, '*x\_or\_y*': '*y*', '*id*': *4*},

    {'*x\_min*': *1* \* nautical\_mile\_to\_meter, '*x\_max*': *2.5* \* nautical\_mile\_to\_meter, '*y\_min*': *0* \* nautical\_mile\_to\_meter, '*y\_max*': *2.5* \* nautical\_mile\_to\_meter, '*x\_or\_y*': '*x*', '*id*': *5*},

    {'*x\_min*': *2.5* \* nautical\_mile\_to\_meter, '*x\_max*': *4* \* nautical\_mile\_to\_meter, '*y\_min*': *0* \* nautical\_mile\_to\_meter, '*y\_max*': *2* \* nautical\_mile\_to\_meter, '*x\_or\_y*': '*x*', '*id*': *6*},

    # *添加更多区域...*

]

sum\_lines = *0*

sum\_length = *0*

# *循环拟合每个区域*

plt**.*figure***()

plt**.*title***('*测线区域图*', *fontproperties*="*SimHei*")

plt**.*xlabel***('*x(n mile)*', *fontproperties*="*SimHei*")

plt**.*ylabel***('*y(n mile)*', *fontproperties*="*SimHei*")

***for*** i, region ***in*** ***enumerate***(*regions*):

    alpha, center, model, r\_squared, ran = ***fit\_region***(*data*,\*\**region*)

***print***(*f"Region: x in [*{*region*['*x\_min*']/ *nautical\_mile\_to\_meter:.2f*}*,* {*region*['*x\_max*']/ *nautical\_mile\_to\_meter:.2f*}*] nautical miles, y in [*{*region*['*y\_min*']/ *nautical\_mile\_to\_meter:.2f*}*,* {*region*['*y\_max*']/ *nautical\_mile\_to\_meter:.2f*}*] nautical miles"*)

***print***(*f"alpha:* {*alpha*}*"*)

***print***(*f"Center:* {*center*}*"*)

***print***(*f"R^2:* {*r\_squared*}\n*"*)

    color = *colors*[*i* %***len***(*colors*)]  # *获取颜色*

***if*** *region*['*x\_or\_y*'] **==** '*x*':

        line, length, res = ***calculate\_and\_plot\_lines***(*alpha*, *ran*[*0*], *ran*[*1*], *center*)

        sum\_lines += line

        sum\_length += length

***print***(*f'测线数为*{*line*}*'*)

***if*** *region*['*id*'] **==** *5*:

            # *res=res[:-5]*

            res = res

***elif*** *region*['*id*'] **==** *6*:

            # *res=res[:-3]*

            res = res

***for*** i, da ***in*** ***enumerate***(*res*):

*res*[*i*] = *res*[*i*] / nautical\_mile\_to\_meter + (*region*['*x\_min*'] / *1852* + *region*['*x\_max*'] / *1852*) / *2*

*res*[*i*] = (*region*['*x\_min*'] + *region*['*x\_max*']) / *1852* - *res*[*i*]

        plt**.*vlines***(*res*, *region*['*y\_min*']/ *1852*, *region*['*y\_max*']/ *1852*, *colors*=*color*, *linestyles*='*solid*', *label*='')

***else***:

        line, length, res = ***calculate\_and\_plot\_lines***(*alpha*, *ran*[*1*], *ran*[*0*], *center*)

        sum\_length += length

        sum\_lines += line

***print***(*f'测线数为*{*line*}*'*)

***for*** i, da ***in*** ***enumerate***(*res*):

*res*[*i*] = *res*[*i*] / nautical\_mile\_to\_meter + (*region*['*y\_min*'] / *1852* + *region*['*y\_max*'] / *1852*) / *2*

*res*[*i*] = (*region*['*y\_min*'] + *region*['*y\_max*']) / *1852* - *res*[*i*]

        plt**.*hlines***(*res*, *region*['*x\_min*']/ *1852*, *region*['*x\_max*']/ *1852*, *colors*=*color*, *linestyles*='*solid*', *label*='')

plt**.*show***()

***print***(*f'总数为：*{*sum\_lines*}*,总长度为：*{*sum\_length*/*1852*}*'*)

划分图——线条.py

***import*** pandas ***as*** pd

***import*** numpy ***as*** np

***import*** matplotlib**.***pyplot* ***as*** plt

***from*** scipy**.**interpolate ***import*** griddata

# *读取CSV文件*

df = pd**.*read\_csv***('*111111.csv*')

# *选取数据的一个子集（例如，每10个点取一个）*

df\_subset = df**.***iloc*[::*10*,:]

# *提取x, y, z值*

x = *df\_subset*['*x*']**.***values*

y = *df\_subset*['*y*']**.***values*

z = *df\_subset*['*z*']**.***values*

z = -z

# *创建规则网格*

grid\_x, grid\_y = np**.***mgrid*[***min***(*x*):***max***(*x*):*100j*,***min***(*y*):***max***(*y*):*100j*]

# *使用griddata进行插值*

grid\_z = ***griddata***((*x*, *y*), *z*,(*grid\_x*, *grid\_y*), *method*='*linear*')

# *使用matplotlib绘制等高线图*

fig, ax = plt**.*subplots***()

cs = ax**.*contourf***(*grid\_x*, *grid\_y*, *grid\_z*, *levels*=*100*, *cmap*='*viridis*')

c = ax**.*contour***(*grid\_x*, *grid\_y*, *grid\_z*, *colors*='*k*', *levels*=*cs***.***levels*[::*2*])

# *plt.clabel(c, inline=True, fontsize=8)*

# *设置坐标轴标签和标题*

ax**.*set\_xlabel***('*由西向东(海里)*', *fontproperties*="*SimHei*")

ax**.*set\_ylabel***('*由南向北(海里)*', *fontproperties*="*SimHei*")

ax**.*set\_title***('*区域海底深度图*', *fontproperties*="*SimHei*")

# *添加颜色条*

plt**.*colorbar***(*cs*)

# *在地形图上绘制分割线*

plt**.*hlines***(*4.5*, *0*, *4*, *colors*='*r*', *linestyles*='*solid*')

plt**.*hlines***(*2.5*, *0*, *2.5*, *colors*='*r*', *linestyles*='*solid*')

plt**.*hlines***(*2*, *2.5*, *4*, *colors*='*r*', *linestyles*='*solid*')

plt**.*vlines***(*1*, *0*, *2.5*, *colors*='*r*', *linestyles*='*solid*')

plt**.*vlines***(*2.5*, *0*, *4.5*, *colors*='*r*', *linestyles*='*solid*')

# *设置坐标轴范围*

plt**.*xlim***(*0*,***max***(*x*))

plt**.*ylim***(*0*,***max***(*y*))

# *在左上角矩形内添加文本*

plt**.*text***(*2*, *4.75*,'*区域①*', *fontsize*=*12*, *ha*='*center*', *va*='*center*', *fontproperties*="*SimHei*", *bbox*=***dict***(*facecolor*='*white*', *alpha*=*1*))

plt**.*text***(*1.25*, *3.5*,'*区域②*', *fontsize*=*12*, *ha*='*center*', *va*='*center*', *fontproperties*="*SimHei*", *bbox*=***dict***(*facecolor*='*white*', *alpha*=*1*))

plt**.*text***(*3.25*, *3.25*,'*区域③*', *fontsize*=*12*, *ha*='*center*', *va*='*center*', *fontproperties*="*SimHei*", *bbox*=***dict***(*facecolor*='*white*', *alpha*=*1*))

plt**.*text***(*0.5*, *1.25*,'*区域④*', *fontsize*=*12*, *ha*='*center*', *va*='*center*', *fontproperties*="*SimHei*", *bbox*=***dict***(*facecolor*='*white*', *alpha*=*1*))

plt**.*text***(*1.75*, *1.25*,'*区域⑤*', *fontsize*=*12*, *ha*='*center*', *va*='*center*', *fontproperties*="*SimHei*", *bbox*=***dict***(*facecolor*='*white*', *alpha*=*1*))

plt**.*text***(*3.25*, *1*,'*区域⑥*', *fontsize*=*12*, *ha*='*center*', *va*='*center*', *fontproperties*="*SimHei*", *bbox*=***dict***(*facecolor*='*white*', *alpha*=*1*))

# *显示图形*

plt**.*show***()