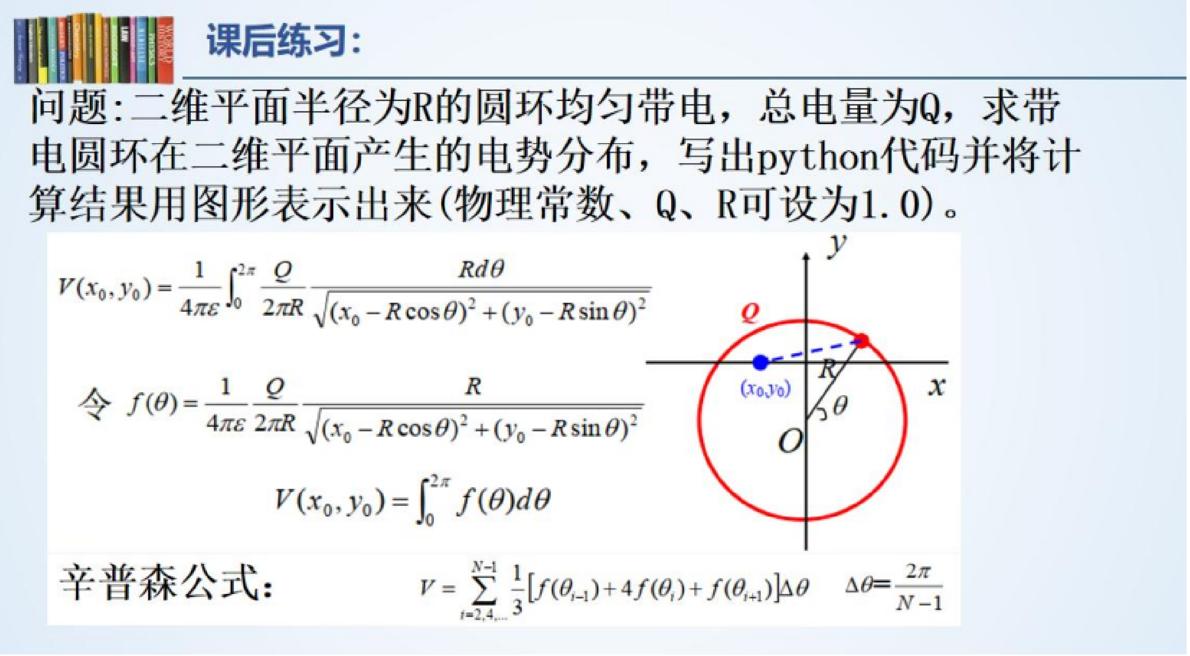
计算物理第二次作业

信息

- 韩懿杰
- 20213006413

题目

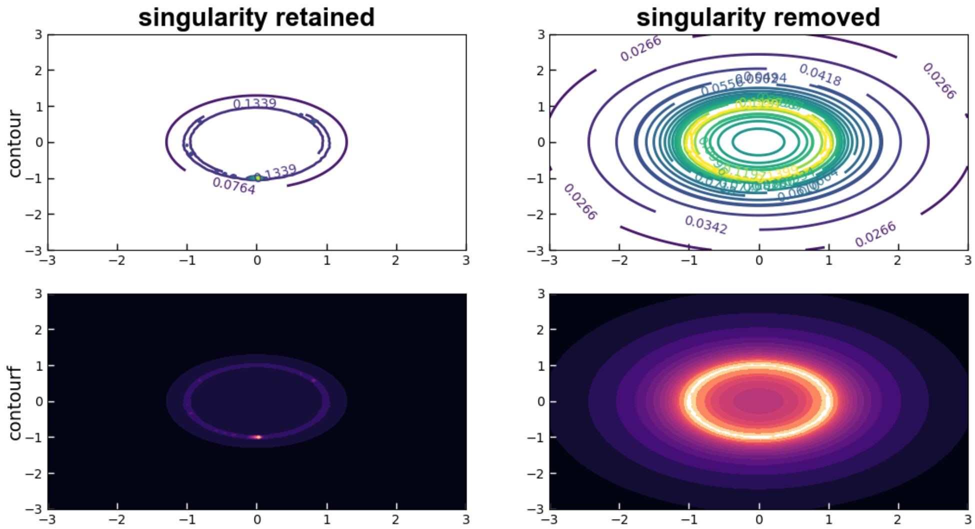


文字解释

- simps 法积分：环形带电体对空间任意一点电势；历遍6*6之范围，不予赘述。
- 问题：contour函数的*levels*参数受奇点影响较大；具体体现为下图第一列，即奇点点抢占大量的颜色区间。
    - contour给了一些收敛的方法，如对elevation进行log压缩。但此题数据范围极小且近0，效果不佳。
    - 我尝试以 $\frac{1}{1-\exp^{-a*x}}$ 或者 $\tanh(ax)$ 等强收敛函数进行压缩，效果尚可，但数据特征破坏较大。
    - 于是乎有个朴实无华的操作，即取 0.17*np.max(ring_distribution_simps）之上的范围颜色相同，且对梯度变化较大又远离奇点的范围更加细分，参考代码：

```
levels21=np.linspace(np.min(ring_distribution_simps),0.06*np.max(ring_distribution_simps),5)
levels22=np.linspace(0.061*np.max(ring_distribution_simps),0.12*np.max(ring_distribution_simps),10)
levels23=np.linspace(0.121*np.max(ring_distribution_simps),0.17*np.max(ring_distribution_simps),3)
levels2=np.concatenate((levels21,levels22,levels23),axis=0)
```

图片示范



代码示例

- Notice: 输出已注释
- 运行代码下载，复制会格式紊乱：https://github.com/hanyijie1/Homework_of_computate_physics

```
import numpy as np
import scipy.integrate as spi #自适应高斯-乌尔森积分法。
import matplotlib.pyplot as plt
def Point_potential(theta,x0,y0,R,Q,epsilon): #single_point_effect
    c=1/(4*np.pi*epsilon)*Q/(2*np.pi*R)*R
    rho=c/np.sqrt((x0-R*np.cos(theta))**2+(y0-R*np.sin(theta))**2)
    return rho
def Ring_potential_simps(x0,y0,R,Q,epsilon):
    theta=np.linspace(0,2*np.pi,100)
    point_potential=Point_potential(theta,x0,y0,R,Q,epsilon)
    ring_potential=spi.simps(point_potential, theta)
    return ring_potential
#Date
## Self_Deifition_variable
R=1
Q=1
epsilon=1
precision=100 #precison
extent1=np.array([-1,1,-1,1])*3 #extent
extent2=np.array([-1,1,-1,1])*3
## solve the distribution of potential of ring
x0=np.linspace(extent1[0],extent1[1],precision)
y0=np.linspace(extent1[2],extent1[3],precision)
ring_distribution_quan=np.zeros((precision,precision))
ring_distribution_simps=np.zeros((precision,precision))
for i in range(np.size(x0)):
    for j in range(np.size(y0)):
        ring_distribution_simps[i,j]=Ring_potential_simps(x0[i],y0[j],R,Q,epsilon)
## Graph
### Data
```

```
X,Y = np.meshgrid(x0,y0)
fig = plt.figure()
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,3)
ax3 = fig.add_subplot(2,2,2)
ax4 = fig.add_subplot(2,2,4)
levels1=np.linspace(np.min(ring_distribution_simps),np.max(ring_distribution_simps),15)
levels21=np.linspace(np.min(ring_distribution_simps),0.06*np.max(ring_distribution_simps),5)
levels22=np.linspace(0.061*np.max(ring_distribution_simps),0.12*np.max(ring_distribution_simps),10)
levels23=np.linspace(0.121*np.max(ring_distribution_simps),0.17*np.max(ring_distribution_simps),3)
levels2=np.concatenate((levels21,levels22,levels23),axis=0)


### plot contour画的是等高线中的线，而contourf画的是登高线之间的区域
cs1=ax1.contour(X,Y,ring_distribution_simps,levels=levels1,origin='lower',linewidths=2,extent=extent1)
ax1.clabel(cs1)
cs2=ax2.contourf(X,Y,ring_distribution_simps,levels=levels1,origin='lower',extent=extent1,cmap='magma')
cs3=ax3.contour(X,Y,ring_distribution_simps,levels=levels2,origin='lower',linewidths=2,extent=extent2)
ax3.clabel(cs3)
cs4=ax4.contourf(X,Y,ring_distribution_simps,levels=levels2,origin='lower',extent=extent2,cmap='magma')

### title
ax1.set_title("singularity retained",fontname='Arial',fontsize=20,weight='bold',x=0.5,y=1)
ax3.set_title("singularity removed",fontname='Arial',fontsize=20,weight='bold',x=0.5,y=1)

## axis
### label
ax1.set_ylabel("contour",fontsize=14,labelpad=0)
ax2.set_ylabel("contourf",fontsize=14,labelpad=0)
### tick
ax1.tick_params(axis='both',direction='in',color='black',length=5,width=1) #axis='x'or'y'or'both
ax2.tick_params(axis='both',direction='in',color='white',length=5,width=1)
ax3.tick_params(axis='both',direction='in',color='black',length=5,width=1)
ax4.tick_params(axis='both',direction='in',color='white',length=5,width=1)

# output
# plt.savefig('Homework2.png',dpi=300) #png picture
plt.show()
```