PHYS2160 Introductory Computational Physics
Project report

Question 1

Theory

In part (a), the electric potential at a point due to different object is required to find. Therefore, it is necessary to determine the electric potential and compare the numerical solution with the analytical solution.

The general equation for finding electric potential:

$$V(\vec{r}) = \frac{1}{4\pi\epsilon}\int \frac{dq}{|\vec{r} - \vec{r}'|}$$

The analytical solution of electric potential at a point above the centre of a uniformly charged wire:

$$V(z) = \frac{Q}{4\pi\epsilon}\ln\left(\frac{\sqrt{z^2 + \left(\frac{L}{2}\right)^2} + \frac{L}{2}}{\sqrt{z^2 + \left(\frac{L}{2}\right)^2} - \frac{L}{2}}\right)$$

The numerical solution of electric potential at a point above the centre of a uniformly charged wire:

$$V(z) = \frac{1}{4\pi\epsilon}\int \frac{\lambda dx}{\sqrt{x^2 + z^2}}$$

$$\lambda = Q/L$$

$$V(z) = \frac{Q}{4\pi\epsilon L}\int_{-L/2}^{L/2} \frac{dx}{\sqrt{x^2 + z^2}}$$

The analytical solution of electric potential at a point above the centre of a uniformly charged ring:

$$V(z) = \frac{Q}{4\pi\epsilon\sqrt{z^2 + R^2}}$$

The numerical solution of electric potential at a point above the centre of a uniformly charged ring:

$$V(z) = \frac{1}{4\pi\epsilon}\int_0^{2\pi} \frac{\lambda R d\theta}{\sqrt{z^2 + R^2}}$$

$$\lambda = \frac{Q}{2\pi R}$$

$$V(z) = \frac{1}{8\pi\epsilon}\int_0^{2\pi} \frac{d\theta}{\sqrt{z^2 + R^2}}$$

The analytical solution of electric potential at a point above the centre of a uniformly charged disk:

$$V(z) = \frac{Q}{2\pi\epsilon R^2}(\sqrt{z^2 + R^2} - z)$$

The numerical solution of electric potential at a point above the centre of a uniformly charged disk:

$$dV = \frac{dq}{4\pi\epsilon\sqrt{x^2 + r^2}}$$

$$dV = \frac{\sigma 2\pi r dr}{4\pi\epsilon\sqrt{x^2 + r^2}}$$

$$\sigma = \frac{Q}{\pi R^2}$$

$$V(z) = \frac{Q}{2\pi\epsilon R^2}\int_0^{R} \frac{r dr}{\sqrt{x^2 + r^2}}$$

The analytical solution of electric potential at a point above the centre of a uniformly charged sphere:

$$V(z) = \frac{Q}{8\pi\epsilon R}(3 - \frac{z^2}{R^2})$$

$$V(z) = \frac{Q}{4\pi\epsilon z}$$

The numerical solution of electric potential at a point above the centre of a uniformly charged sphere:

$$E = \frac{kQ}{r^2} \ (r > R)$$

$$V = -\int_{\infty}^{r} \frac{kQ}{r^2} dr$$

$$E = \frac{kQ}{R^3} r \ (r < R)$$

$$V(z) = -\int_{\infty}^{r} \frac{kQ}{r^2} dr - \int_{R}^{r} \frac{kQ}{R^3} r dr$$

In part (b), the objective is to numerically evaluate the integral and find the electric potential at any point on xz plane.

For the uniformly charged wire:

$$V(x,z) = k \int_{-L/2}^{L/2} \frac{\lambda dx}{|\sqrt{(x-x')^2 + z^2}}$$

$$V(x,z) = \frac{kQ}{L} \int_{-L/2}^{L/2} \frac{dx}{|\sqrt{(x-x')^2 + z^2}|}$$

For the uniformly charged ring:

$$V(x,z) = \int_{0}^{2\pi} \frac{k\lambda R d\theta}{|\sqrt{(x-x')^2 + y^2 + z^2}|}$$

$$x = cos\theta, y = sin\theta$$

$$\lambda = \frac{Q}{2\pi R}$$

$$V(x,z) = \frac{kQ}{2\pi} \int_{0}^{2\pi} \frac{d\theta}{|\sqrt{(cos\theta - x')^2 + (sin\theta)^2 + z^2}|}$$

For the uniformly charged disk:

$$V(x,z) = k\int_0^{2\pi} d\theta \int_0^R \frac{\sigma r dr}{|\sqrt{(rcos\theta - x')^2 + (rsin\theta)^2 + z^2}|}$$

$$\sigma = \frac{Q}{\pi R^2}$$

$$V(x,z) = \frac{kQ}{\pi R^2}\int_0^{2\pi} d\theta \int_0^R \frac{\sigma r dr}{|\sqrt{(rcos\theta - x')^2 + (rsin\theta)^2 + z^2}|}$$

For the uniformly charged sphere:

$$V(x,z) = k(\int_0^d \frac{\rho(4\pi r^2)dr}{d} + \int_d^R \frac{\rho(4\pi r^2)dr}{r})$$

$$d = \sqrt{x^2 + z^2}$$

Algorithm

In part(a), using the SciPy function to numerically compute the value of electric potential and compare the difference with the analytical solution with matplotlib. Firstly, import NumPy, SciPy and matplotlib. Also, it is necessary to import the electric constant from SciPy to obtain the accurate result. Secondly, defining a function which is the analytical solution of the electric potential and use SciPy to integrate another function numerically. These equations are listed above. Moreover, in order to generate more data, using np.linspace function to create two arrays of the x and z value. To store the value of the electric potential of different points, creating a list then use for loop to compute the value one by one. After that use the np.array function to change it to an array. Finally, plotting can done with matplotlib and use the legend function to show two different line. In the plotting command, we should specify the colour of each line because the result of two methods is almost the same so that the two lines will coincide. By specifying the colour of the line, there are actually two lines. But because they are the same so the colour will change, for example, if the yellow line coincides with the green line, the resultant colour will be lighter.

In part(b), plotting the equipotential lines of different object on xz plane and evaluating the electric potential of an arbitrary point on that plane. These can be done by plotting contour line and 3D plot. First of all, import those relevant packages such as NumPy, SciPy and matplotlib. And then define a function that can calculate the potential of a point on xz plane, which are listed above. For the uniformly charged wire, firstly, define a function and then create two arrays of x and z. After that, using np.meshgrid function to make them become a 2d array so that they can be used to make a contour plot. Moreover, using np.zeros function to create an array that can store the value of electric potential and using for loop to compute the magnitude of the electric potential at different point and then change the value of that array. Finally, begin to make a contour plot and 3D plot with matplotlib. In addition, for the uniformly charged disk and sphere, use dblquad function to numerically evaluate the value of the electric potential. The method is the same as the uniformly charged wire and ring, but the major difference is the double integral calculation.

Listing of your program

(a)

Uniformly charged wire:

```
#This program can plot the analytical and numerical value of the electric potential of a wire
#This program is wriiten by XU Shiyang on 17/4/2022
#first import the relevant module
import numpy as np
import math
import scipy.constants as pc
import matplotlib.pyplot as plt
from scipy.integrate import quad
e = pc.value('electric constant')
def V1(e, Q, L, z):
    return (Q/(4*np.pi*e*L))*np.log((math.sqrt(z**2+(L/2)**2)+L/2)/(math.sqrt(z**2+(L/2)**2)-L/2))
z_1 = np.linspace(0.1,10,1000)
#create a list that can store the value of V
V1_lst1 = []
for i in z_1:
    V1_lst1.append(V1(e,1,1,i))
V1_ar1 = np.array(V1_lst1)
def func1(x):
    return (1/(4*math.pi*e))/math.sqrt(x**2+z**2)
lst1 = []
for i in z_1:
    z = i
    y = quad(func1, -1/2, 1/2)
    lst1.append(y[0])
#using the reshape to change the array
la1 = np.array(lst1).reshape(1000,)
fig = plt.figure(figsize=(12, 6))
line1 = plt.plot(z_1,V1_ar1,label = 'Analytical',color='y')
line2 = plt.plot(z_1, la1,label='Numerical',color='g')
plt.xlabel('Z Value')
plt.ylabel('Electric Potential')
plt.title('Uniformly Charged Wire')
plt.legend()
```

Uniformly charged ring:

```
#This program can plot the analytical and numerical value of the electric potential of a ring
#This program is wriiten by XU Shiyang on 17/4/2022
#first import the relevant module
import numpy as np
import math
import scipy.constants as pc
import matplotlib.pyplot as plt
from scipy.integrate import quad
e = pc.value('electric constant')
def V2(e, Q, R, z):
    return Q/(4*math.pi*e*math.sqrt(z**2+R**2))
z_1 = np.linspace(0,10,1000)
V2_lst1 = []
#create a list that can store the value of V
for i in z_1:
    V2_lst1.append(V2(e,1,1,i))
V2_ar1 = np.array(V2_lst1)
def func2(theta):
    return theta**0/(8*math.pi**2*e*math.sqrt(z**2+1))
V2_lst1 = []
for i in z_1:
    z = i
    y1 = quad(func2, 0, 2*math.pi)
    V2_lst1.append(y1[0])
#using the reshape to change the array
V2_la1 = np.array(V2_lst1).reshape(1000,)
fig = plt.figure(figsize=(12, 6))
line1 = plt.plot(z_1,V2_ar1,label = 'Analytical',color='y')
line2 = plt.plot(z_1, V2_la1,label='Numerical',color='g')
plt.xlabel('Z Value')
plt.ylabel('Electric Potential')
plt.title('Uniformly Charged ring')
plt.legend()
```

Uniformly charged disk:

```
#This program can plot the analytical and numerical value of the electric potential of a disk
#This program is wriiten by XU Shiyang on 17/4/2022
```

```
#first import the relevant module
import numpy as np
import math
import scipy.constants as pc
import matplotlib.pyplot as plt
from scipy.integrate import quad
e = pc.value('electric constant')
def V3(e, Q, R, z):
    return Q*(math.sqrt(abs(z)**2+R**2)-abs(z))/(2*math.pi*e*R**2)
z_1 = np.linspace(0,10,1000)
V3_lst1 = []
#create a list that can store the value of V
for i in z_1:
    V3_lst1.append(V3(e,1,1,i))
V3_ar1 = np.array(V3_lst1)
def func3(r):
    return (1/(2*math.pi*1*e))*r/math.sqrt(z**2+r**2)
V3_lst1 = []
for i in z_1:
    z = i
    y = quad(func3, 0, 1)
    V3_lst1.append(y[0])
#using the reshape to change the array
V3_la1 = np.array(V3_lst1).reshape(1000,)
fig = plt.figure(figsize=(12, 6))
line1 = plt.plot(z_1,V3_ar1,label = 'Analytical',color='y')
line2 = plt.plot(z_1, V3_la1,label='Numerical',color='g')
plt.xlabel('Z Value')
plt.ylabel('Electric Potential')
plt.title('Uniformly Charged disk')
plt.legend()
```

Uniformly charged sphere:

```
#This program can plot the analytical and numerical value of the electric potential of a sphere
#This program is wriiten by XU Shiyang on 17/4/2022
#first import the relevant module
import numpy as np
import math
import scipy.constants as pc
```

```
import matplotlib.pyplot as plt
from scipy.integrate import quad
e = pc.value('electric constant')
def V4(e, Q, R, z):
   if z < R:
      return (Q/(8*math.pi*e*R))*(3-z**2/R**2)
   else:
      return Q/(4*math.pi*e*z)
z_1 = np.linspace(0,10,1000)
V4_lst1 = []
for i in z_1:
   V4_lst1.append(V4(e,1,1,i))
V4_ar1 = np.array(V4_lst1)
def func4(r):
   return -1/(4*np.pi*e*r**2)
r2 = np.linspace(1, 10, 900)
lst2 = []
#create a list that can store the value of V
for i in r2:
   y = quad(func4, np.inf, i)
   lst2.append(y[0])
def func4_1(r):
   return -r/(4*np.pi*e*1**3)
r1 = np.linspace(0, 1, 100)
y1 = quad(func4, np.inf, 1)[0]
lst1 = []
for i in r1:
   y2 = quad(func4_1, 1, i)[0]+y1
   lst1.append(y2)
lst = lst1+lst2
#using the reshape to change the array
r = np.linspace(0, 10, 1000)
fig = plt.figure(figsize=(12, 6))
line1 = plt.plot(z_1,V4_ar1,label = 'Analytical',color='y')
line2 = plt.plot(r, lst, label='Numerical', color='g')
plt.xlabel('Z Value')
plt.ylabel('Electric Potential')
plt.title('Uniformly Charged solid sphere')
plt.legend()
plt.show()
```

(b)

Uniformly charged wire:

```
#this program can plot the equipotential line of the uniformly charged wire
#this program is written by XU Shiyang on 18/4/2022
#first import the relevant modules
from scipy import integrate
import math
import numpy as np
import scipy.constants as pc
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
e = pc.value('electric constant')
k = 1/(4*np.pi*e)
L = 1
Q = 1
#define a funtion
def wire(x, x0, z0):
    d = (x - x0)**2 + z0**2
    return k*Q/L/abs(np.sqrt(d))
x = np.linspace(-1, 1, 100)
z = np.linspace(-1, 1, 100)
X, Z = np.meshgrid(x, z)
V = np.zeros((100,100))
#create a zero array that can store the value of electric potential
for i in range(100) :
    for j in range(100) :
        V[i][j] = integrate.quad(wire, -L/2, L/2, args=(X[i][j], Z[i][j]))[0]
fig1 = plt.figure(dpi=200)
ax1 = fig1.add_subplot(111)
ax1.set_xlabel('X')
ax1.set_ylabel('Z')
ax1.set_title('Equipotential line of a wire')
cpf = ax1.contourf(X, Z, V)
fig1.colorbar(cpf)
plt.show(fig1)
fig2 = plt.figure(dpi=200)
```

```
ax2 = fig2.add_subplot(111, projection='3d')
ax2.set_xlabel('X')
ax2.set_ylabel('Y')
ax2.set_zlabel('V')
ax2.set_title('Electric potential of a wire')
sur = ax2.plot_surface(X, Z, V, cmap='rainbow')
plt.show(fig2)
```

Uniformly charged ring:

```
#this program can plot the equipotential line of the uniformly charged ring
#this program is written by XU Shiyang on 18/4/2022
#first import the relevant modules
from scipy.integrate import quad
import math
import numpy as np
import scipy.constants as pc
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
e = pc.value('electric constant')
R = 1
Q = 1
k = 1/(4*np.pi*e)
#define a funtion
def ring(theta, x0, z0):
    x = np.cos(theta)*R
    y = np.sin(theta)*R
    d2 = (x - x0)**2 + z0**2 + y**2
    return k*Q/(2*math.pi*R)/abs(np.sqrt(d2))
x = np.linspace(-2, 2, 100)
z = x.copy()
X, Z = np.meshgrid(x, z)
#create a zero array that can store the value of electric potential
V = np.zeros((100, 100))
for i in range(100) :
    for j in range(100) :
        V[i][j] = integrate.quad(ring, 0, 2*math.pi, args=(X[i][j], Z[i][j]))[0]
fig1 = plt.figure(dpi=200)
```

```
ax1 = fig1.add_subplot(111)
ax1.set_xlabel('X')
ax1.set_ylabel('Z')
ax1.set_title('Equipotential line of a ring')
cpf = ax1.contourf(X, Z, V)
fig1.colorbar(cpf)
plt.show(fig1)
fig2 = plt.figure(dpi=200)
ax2 = fig2.add_subplot(111, projection='3d')
ax2.set_xlabel('X')
ax2.set_ylabel('Y')
ax2.set_zlabel('V')
ax2.set_title('Electric potential of a ring')
sur = ax2.plot_surface(X, Z, V, cmap='rainbow')
plt.show(fig2)
```

Uniformly charged disk:

```
#this program can plot the equipotential line of the uniformly charged disk
#this program is written by XU Shiyang on 18/4/2022
#first import the relevant modules
from scipy.integrate import dblquad
import math
import numpy as np
import scipy.constants as pc
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
e = pc.value('electric constant')
R = 1
Q = 1
k = (1/(4*np.pi*e))*(Q/(np.pi*R**2))
x = np.linspace(-2, 2, 100)
z = np.linspace(-2, 2, 100)
#define a funtion
def func1(r, theta, x, z):
   d = math.sqrt(((r*np.cos(theta)-x)**2)+(r*np.sin(theta))**2+z**2)
   d1 = abs(d)
   return k*r/d1
```

```
a, b = 0, 2*np.pi
def gfunc(r):
    return 0
def hfunc(r):
    return R
X, Z = np.meshgrid(x, z)
#create a zero array that can store the value of electric potential
V_a = np.zeros((100,100))
for i in range(100):
    for h in range(100):
        V_a[i][h] = dblquad(func1, a, b, gfunc, hfunc, args=(X[i][h], Z[i][h]))[0]
fig1 = plt.figure(dpi=200)
ax1 = fig1.add_subplot(111)
ax1.set_xlabel('X')
ax1.set_ylabel('Z')
ax1.set_title('Equipotential line of a disk')
cpf = ax1.contourf(X, Z, V_a)
fig1.colorbar(cpf)
plt.show(fig1)
fig2 = plt.figure(dpi=200)
ax2 = fig2.add_subplot(111, projection='3d')
ax2.set_xlabel('X')
ax2.set_ylabel('Y')
ax2.set_zlabel('V')
ax2.set_title('Electric potential of a disk')
sur = ax2.plot_surface(X, Z, V_a, cmap='rainbow')
plt.show(fig2)
```

Uniformly charged sphere:

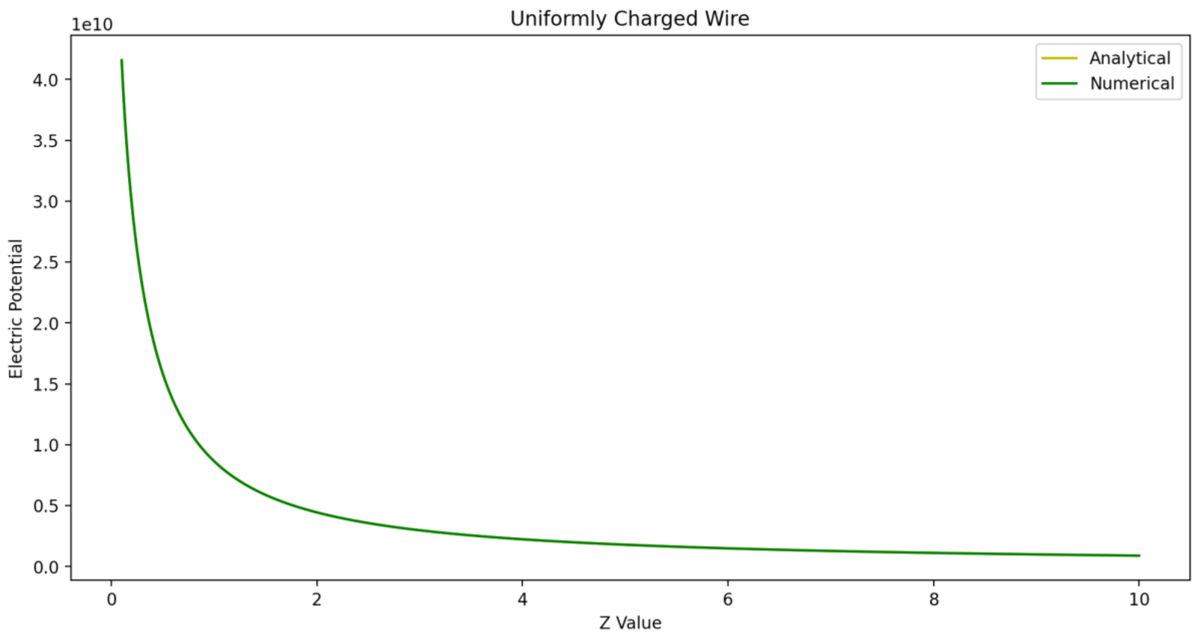
```
#this program can plot the equipotential line of the uniformly charged sphere
#this program is written by XU Shiyang on 18/4/2022
#first import the relevant modules
from scipy.integrate import quad
import math
import numpy as np
import scipy.constants as pc
import matplotlib.pyplot as plt
```

```python
from mpl_toolkits.mplot3d import Axes3D
e = pc.value('electric constant')
R = 1
Q = 1
k = 1/(4*np.pi*e)
rho = Q/(4*np.pi*R**3)
x = np.linspace(-1,1,100)
z = np.linspace(-1,1,100)
X, Z = np.meshgrid(x,z)
def V1(r, x, z):
    d = np.sqrt(x**2+z**2)
    return k*rho*4*np.pi*r**2/d
def V2(r):
    return k*rho*4*np.pi*r**2/r
V1_a = np.zeros((100,100))
for i in range(len(X)):
    for h in range(len(Z)):
        x = X[i][h]
        z = Z[i][h]
        d = np.sqrt(x**2+z**2)
        V1_a[i][h] = quad(V1, 0, d, args=(x, z))[0] + quad(V2, d, R)[0]
fig1 = plt.figure(dpi=200)
ax1 = fig1.add_subplot(111)
ax1.set_xlabel('X')
ax1.set_ylabel('Z')
ax1.set_title('Equipotential line of a sphere')
cpf = ax1.contourf(X, Z, V1_a)
fig1.colorbar(cpf)
plt.show(fig1)
fig2 = plt.figure(dpi=200)
ax2 = fig2.add_subplot(111, projection='3d')
ax2.set_xlabel('X')
ax2.set_ylabel('Y')
ax2.set_zlabel('V')
ax2.set_title('Electric potential of a sphere')
sur = ax2.plot_surface(X, Z, V1_a, cmap='rainbow')
plt.show(fig2)
```

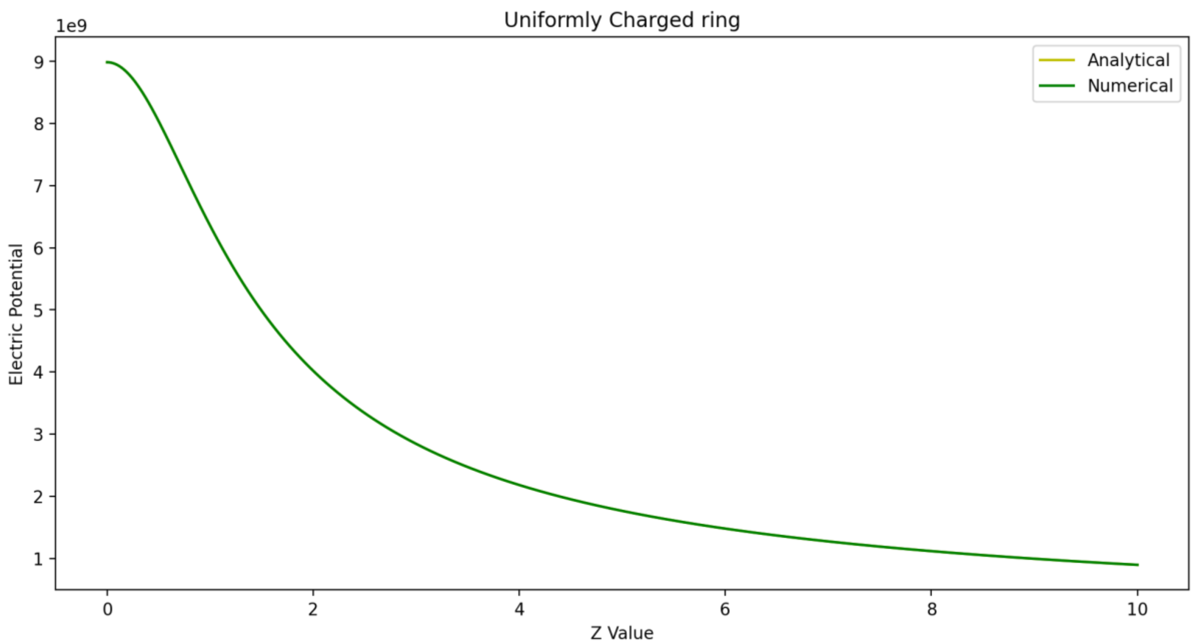
Results

(a)

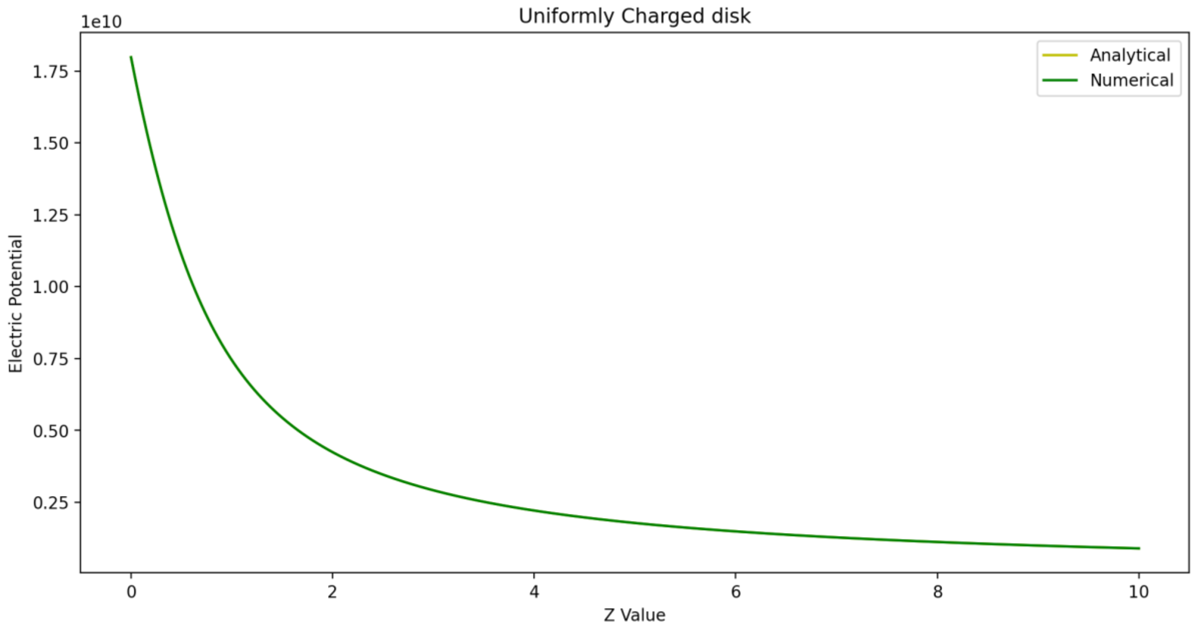
Uniformly charged wire:



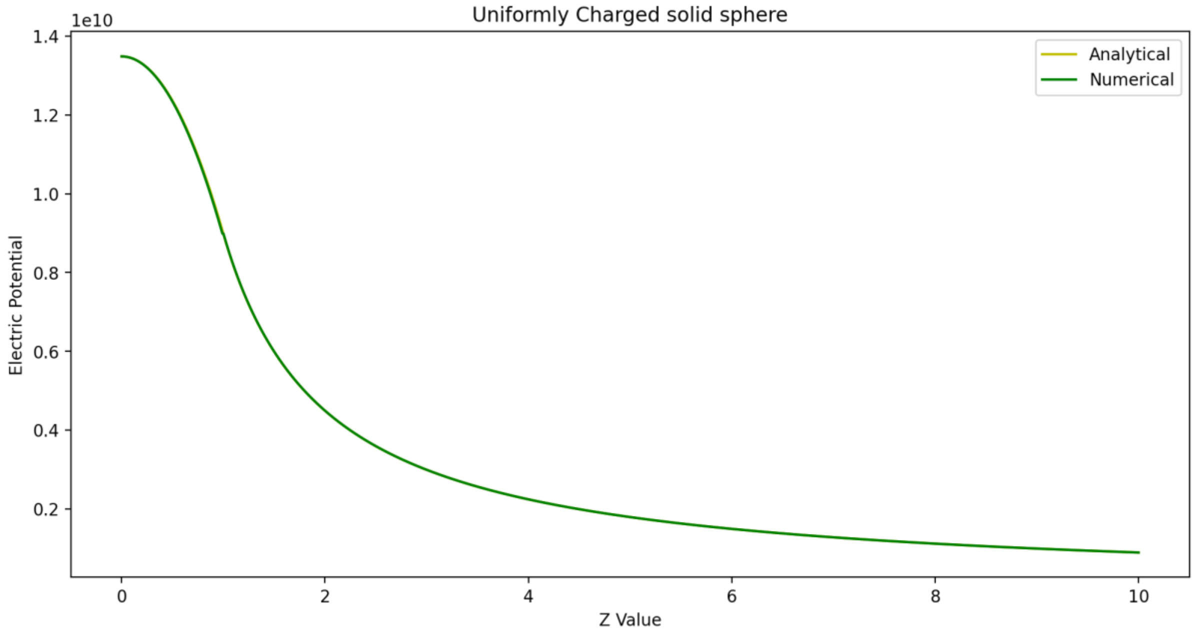
Uniformly charged ring:



Uniformly charged disk:

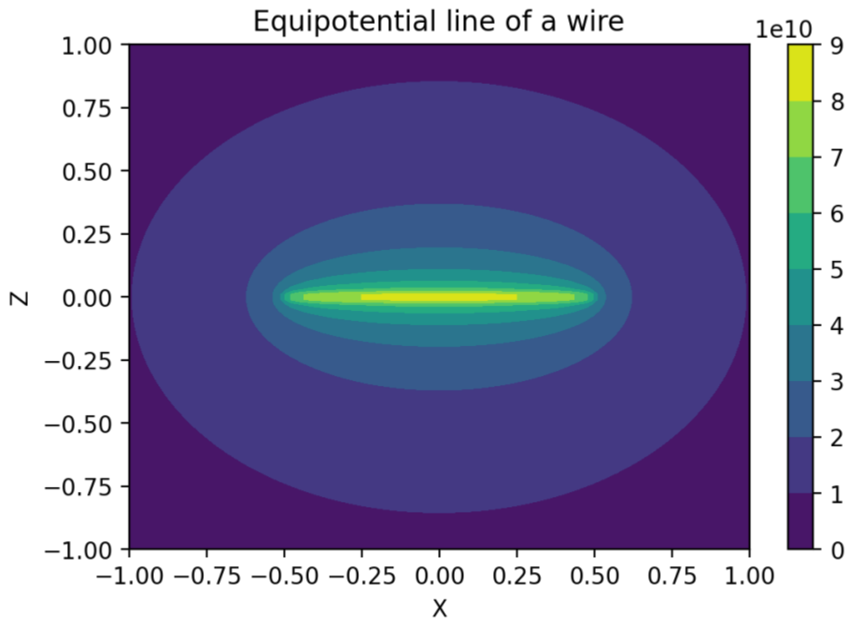


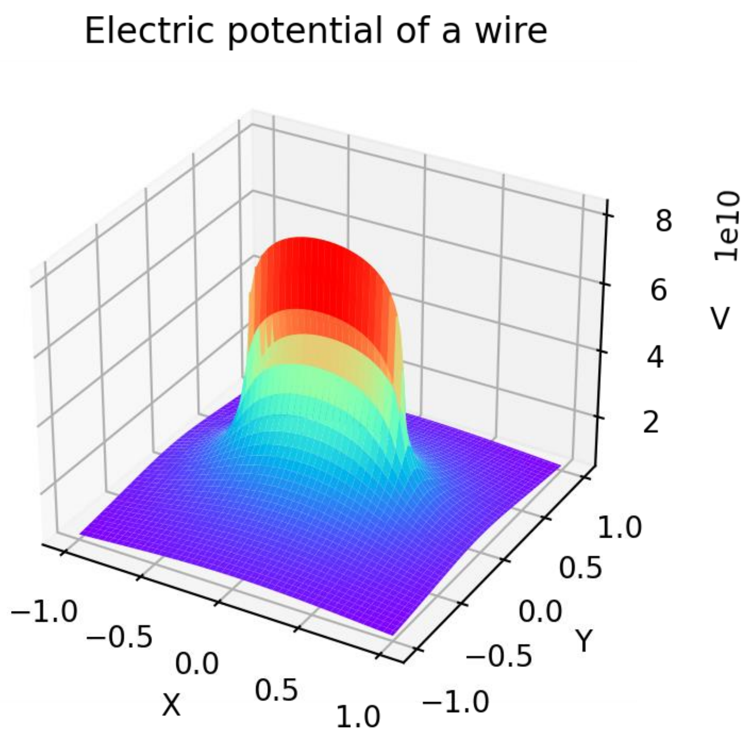
Uniformly charged sphere:



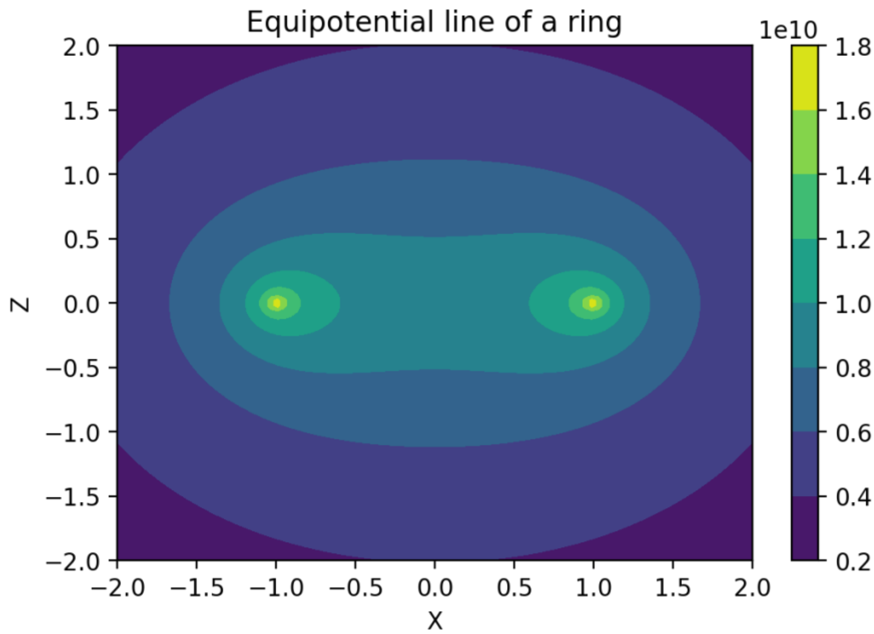
(b)

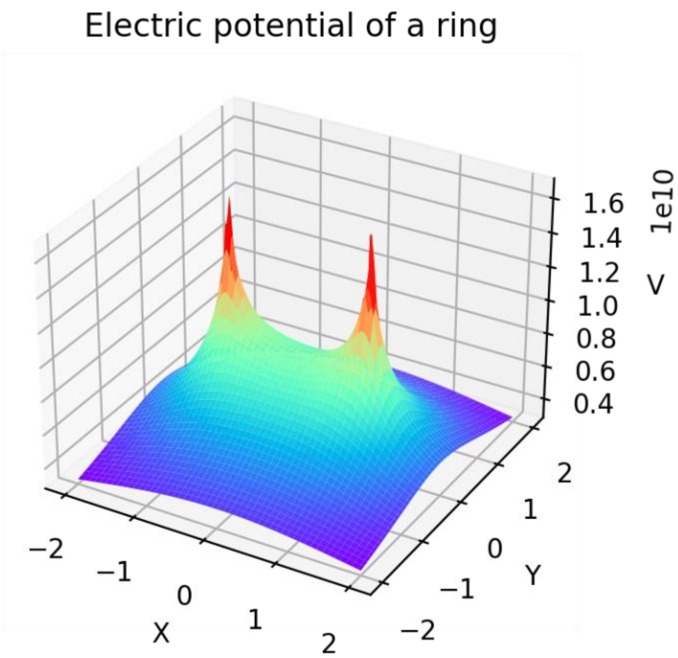
Uniformly charged wire:



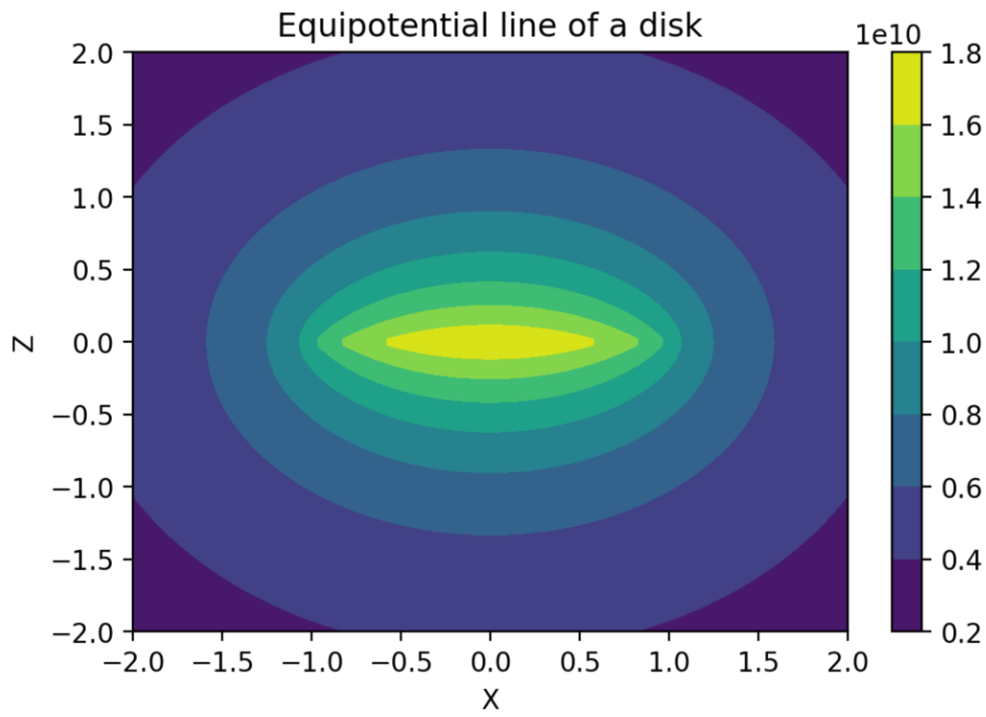


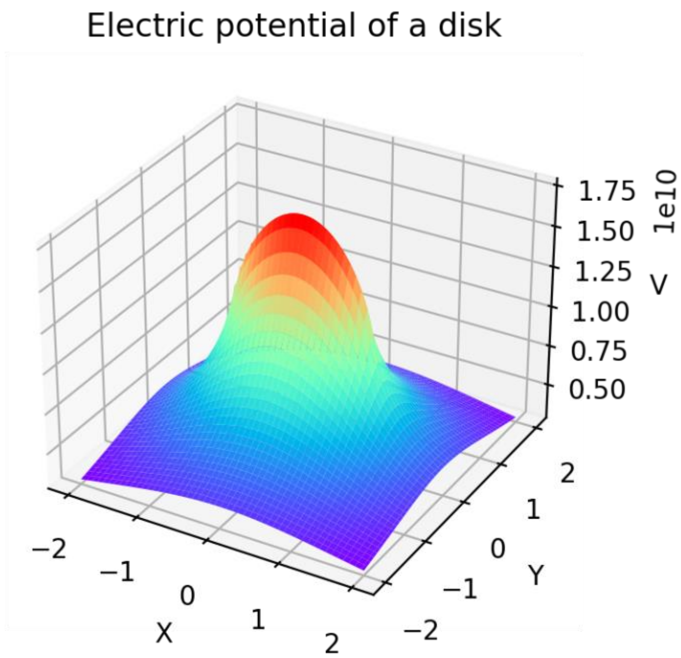
Uniformly charged ring:



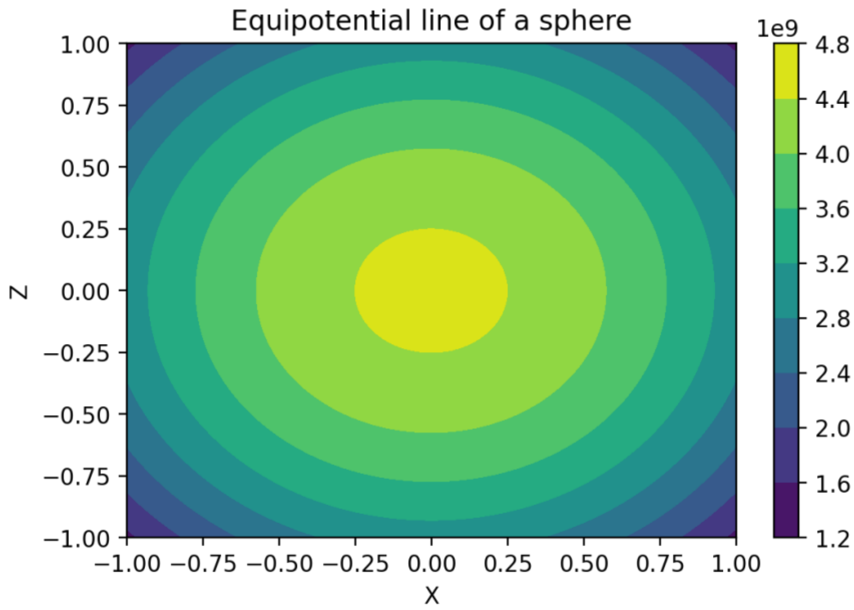


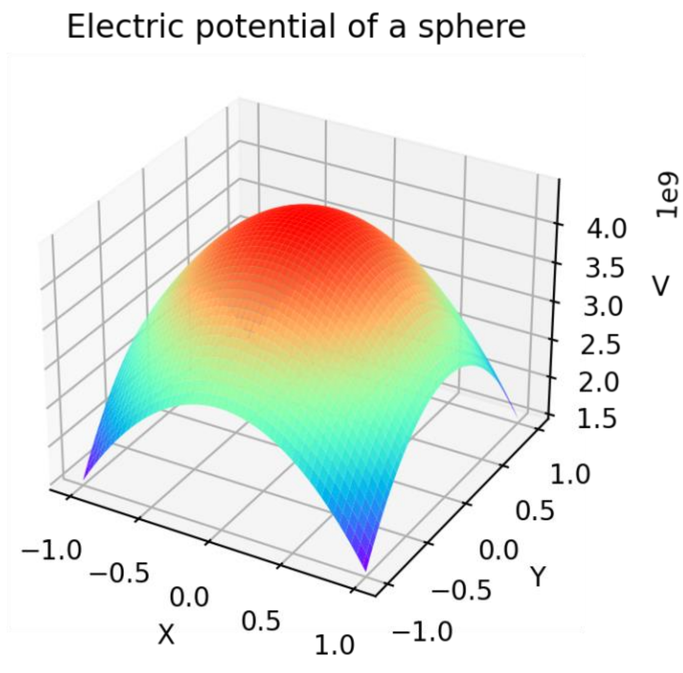
Uniformly charged disk:





Uniformly charged sphere:





Discussion:

In part (a), the numerical result obtained from the SciPy integrate function is exactly the same as the analytical result, although there may be some round off errors, but these errors are relatively small so that two lines look like the same. By increasing the number of data generate by the np.linspace function, the graph will be more smooth. However, the computational time will be increased. Moreover, by adding the arguments epsrel and epsabs, the accuracy of the SciPy integrate function will be further increased, but the time it takes will also increase. Therefore, it might not be necessary to do so since the accuracy of the results are already very high. If we further increase the number of data or lower the errors of SciPy integrate function, it will take a long time to output the results.

In part (b), since it is very difficult to get the analytical result, the only way to do is to use the SciPy integrate function to numerically compute the electric potential at an arbitrary point due to different objects. By consider the continuous charged object as infinity many small particle charges, because of the principle of superposition, all the electric potential can be summed up at a point due to these small particles. And for the dblquad functions, the time it takes to evaluate the result will increase a lot. Although, the equipotential lines of the continuous charged distribution are not accurate due to the limit amount of data, these results are already enough to show the equipotential lines and the magnitude of electric potential.