Parcialpunto2

May 27, 2020

[1]:
```
from google.colab import drive
drive.mount('/content/gdrive/')
import sys
sys.path.append('/conten/gdrive/My Drive/metod')
```

```
Drive already mounted at /content/gdrive/; to attempt to forcibly remount, call
drive.mount("/content/gdrive/", force_remount=True).
```

[2]:
```
import numpy as np
from numpy import pi, mgrid, array, empty, meshgrid
import math
from math import sqrt
from pylab import plot, imshow, linspace, legend, gray, hot, jet,show
import matplotlib.pyplot as plt

print('librerias')
```

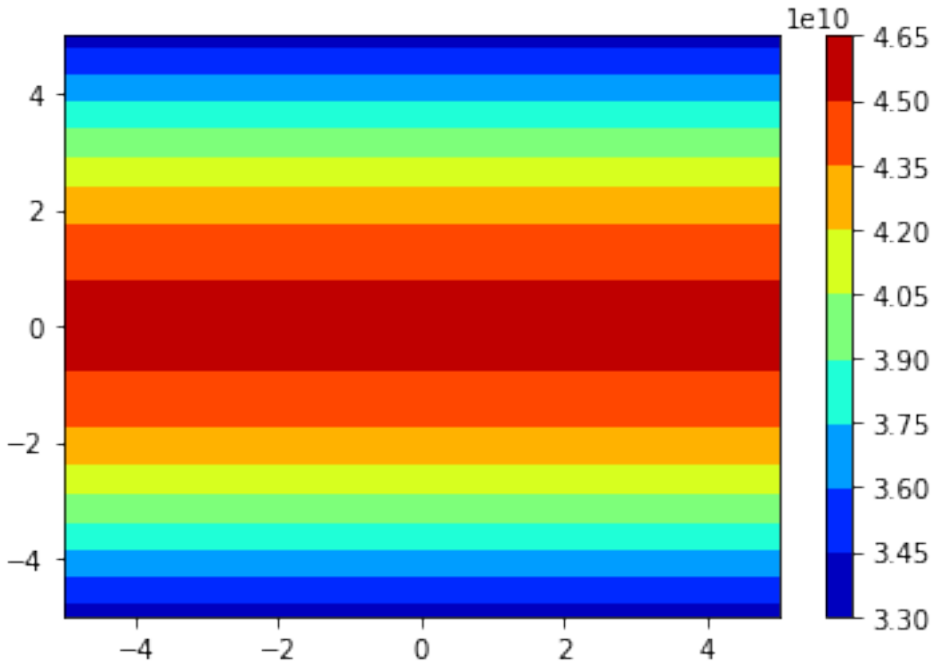
```
librerias
```

[3]:
```
ep = 8.8541878176e-12
l=5
N = 10

for y in range(1,N,1):
  for x in range(1,N,1):


    def f(x):                                          #def. función dependiente x
      return ((x)/(((x**2)+(y**2))))
    def g(y):                                          #def. función dependiente y
      return ((y)/(((x**2)+(y**2))))


a=-5                                                #limites integrales
b=5
c=-5
```

```
d=5
hx=(b-a)/N
hy=(c-b)/N

print('definición de funciones')
```

```
definición de funciones
```

[5]:
```
px=0
py=0
ix=0
iy=0
Ix=0
Iy=0


                                        #Solucionar integral por Simpson
for k in range(1,N,2):                  #pares
  ix += f(a+(k*hx))
  iy += g(c+(k*hy))

for k in range (2,N,2):                 #impares
  px += f(a+(k*hx))
  py += g(c+(k*hy))

Ix=(hx/3)*(f(a)+f(b)+(4*ix)+(2*px))     #Solución integrales
Iy=(hy/3)*(g(c)+g(d)+(4*iy)+(2*py))

E=(1/(4*pi*ep))*(Ix+Iy)                 #Total campo eléctrico
X = linspace(-5,5,500)                  #arreglo que va de -5 hasta 5 con␣
 ↪500 pasos
Y = linspace(-5,5,500)
x,y=meshgrid(X,Y)                       #crea matriz de coordenadas␣
 ↪apartir de los vectores


plt.contourf(y,x,E,cmap='jet')          #plot densidad de carga
plt.colorbar()
plt.show()
```


1e10
4.65
4.50
4.35
4.20
4.05
3.90
3.75
3.60
3.45
3.30
4
2
0
−2
−4
−4
−2
0
2
4