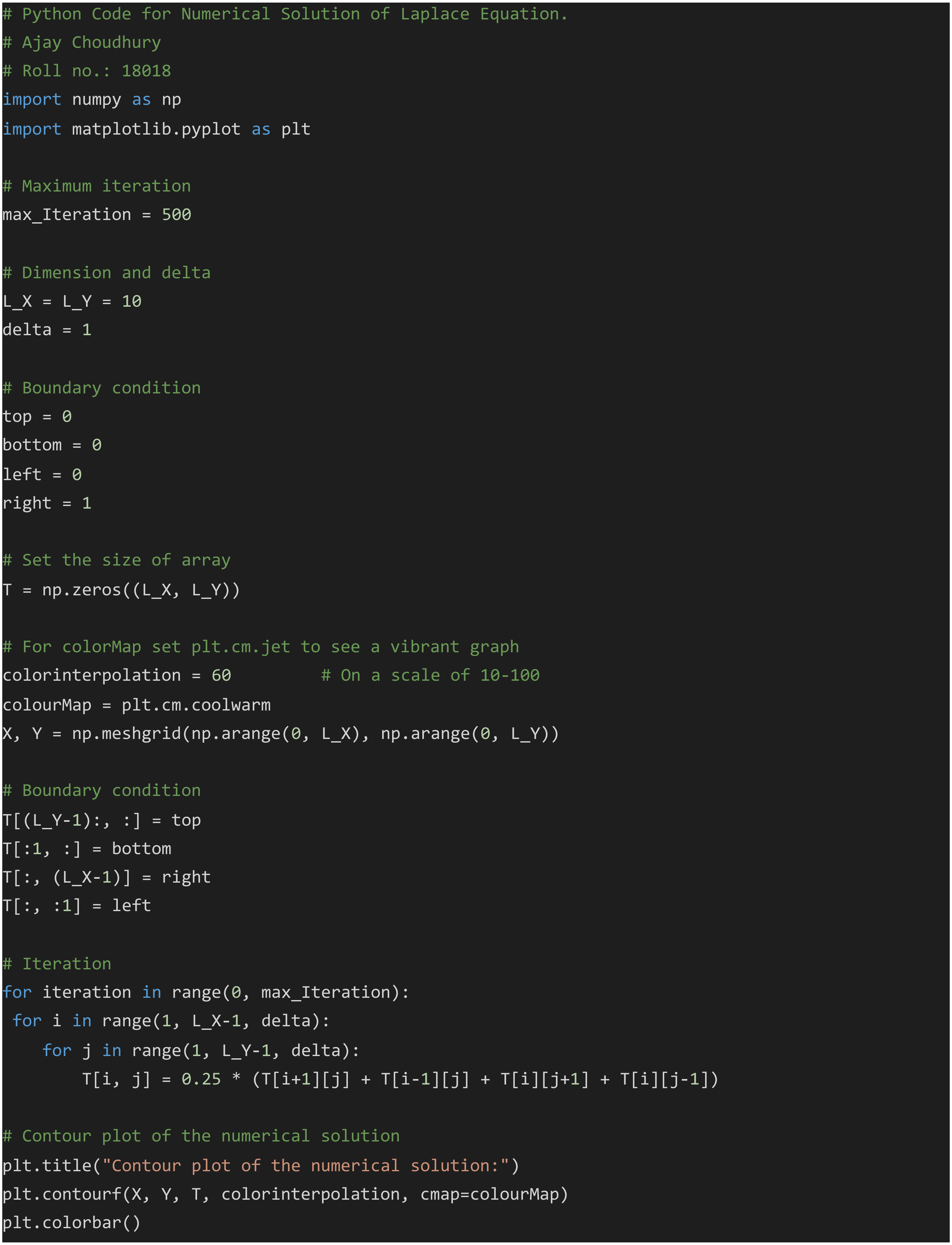
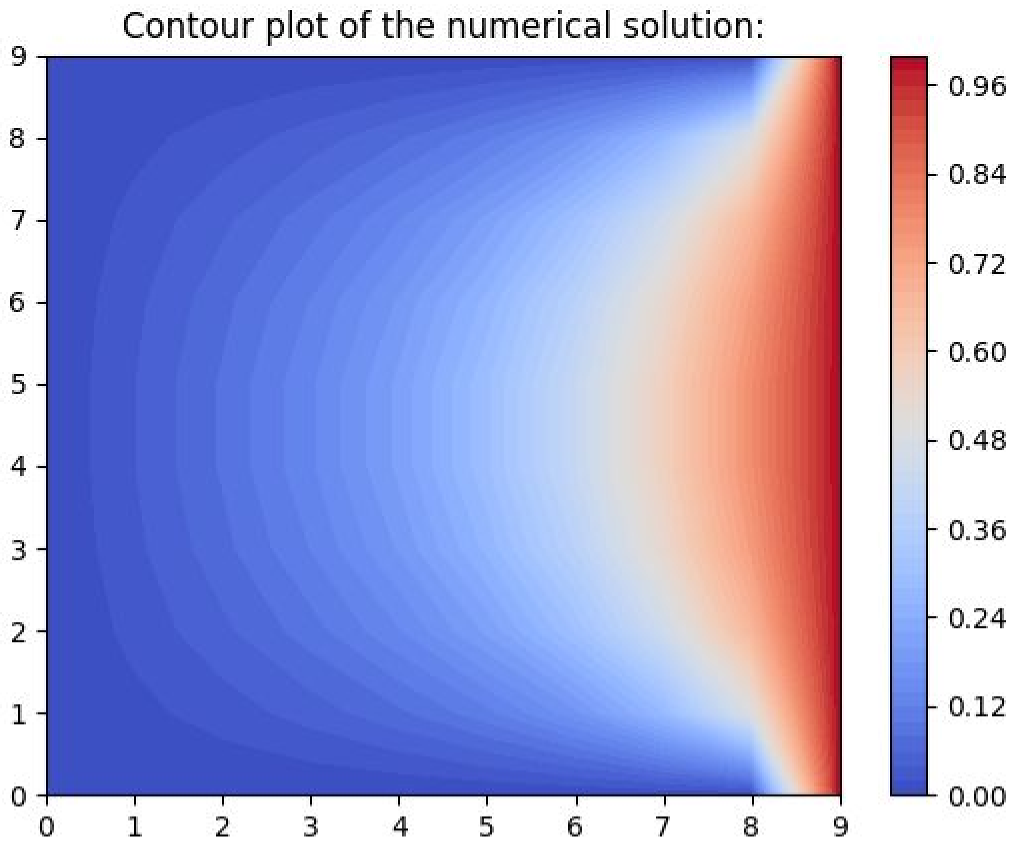
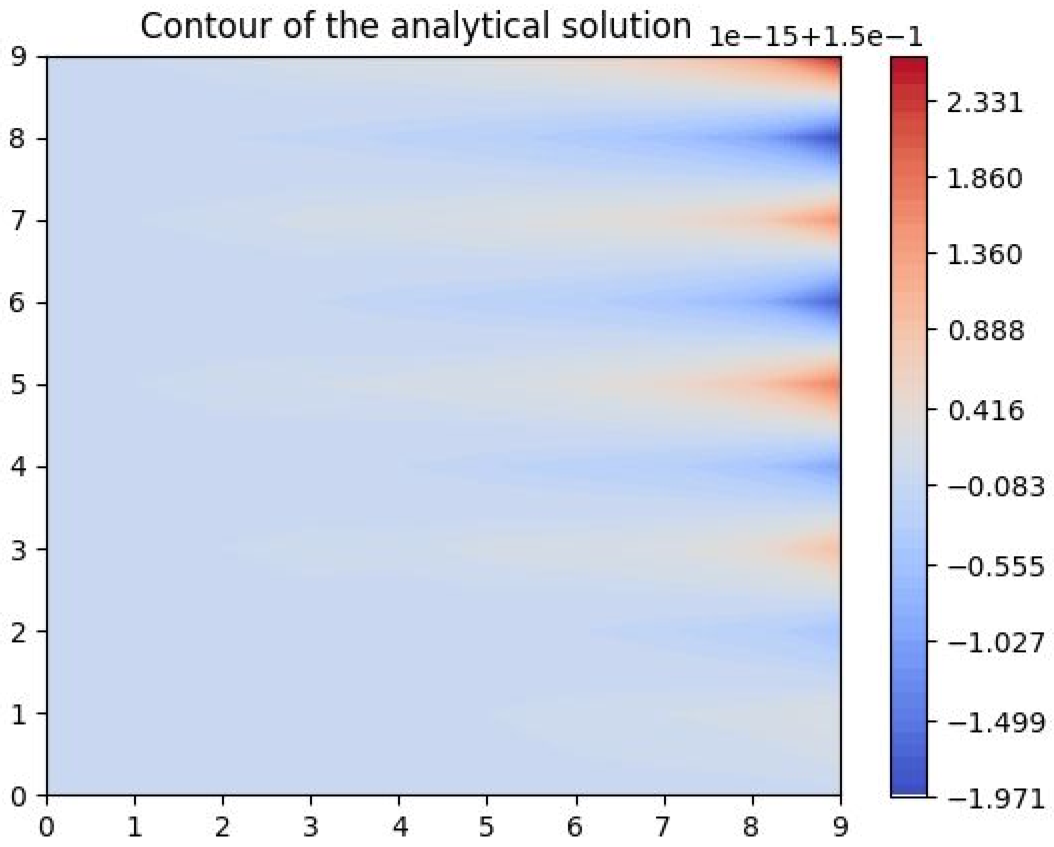


```python
# Python Code for Numerical Solution of Laplace Equation.
# Ajay Choudhury
# Roll no.: 18018
import numpy as np
import matplotlib.pyplot as plt

# Maximum iteration
max_Iteration = 500

# Dimension and delta
L_X = L_Y = 10
delta = 1

# Boundary condition
top = 0
bottom = 0
left = 0
right = 1

# Set the size of array
T = np.zeros((L_X, L_Y))

# For colorMap set plt.cm.jet to see a vibrant graph
colorinterpolation = 60         # On a scale of 10-100
colourMap = plt.cm.coolwarm
X, Y = np.meshgrid(np.arange(0, L_X), np.arange(0, L_Y))

# Boundary condition
T[(L_Y-1):, :] = top
T[:1, :] = bottom
T[:, (L_X-1)] = right
T[:, :1] = left

# Iteration
for iteration in range(0, max_Iteration):
 for i in range(1, L_X-1, delta):
    for j in range(1, L_Y-1, delta):
        T[i, j] = 0.25 * (T[i+1][j] + T[i-1][j] + T[i][j+1] + T[i][j-1])

# Contour plot of the numerical solution
plt.title("Contour plot of the numerical solution:")
plt.contourf(X, Y, T, colorinterpolation, cmap=colourMap)
plt.colorbar()
```

```
# Plot
plt.show()

# N runs from all odd nos. in (0,inf)
N=225
T_A=np.zeros((L_X,L_Y),dtype=np.float64)

for i in range(1,N,2):
 for j in range(L_X):
    for k in range(L_Y):
        T_A[j][k]+=(4/(i*np.pi))*(1/np.sinh(i*np.pi))*np.sinh(i*np.pi*k/10)*np.sin(i*np.pi*j)

#Contour plot of the analytical solution
plt.title("Contour of the analytical solution     ")
plt.contourf(X, Y, T_A+0.15, colorinterpolation, cmap=colourMap)
plt.colorbar()

# Plot
plt.show()
```



Contour plot of the numerical solution:


Contour of the analytical solution
1e−15+1.5e−1
2.331
1.860
1.360
0.888
0.416
−0.083
−0.555
−1.027
−1.499
−1.971
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9