EE2703 Assignment 5 - The Resistor Problem

Nihal John George (EE19B131)

March 20, 2021

Abstract

In this assignment, the Laplace Equation for electric fields is solved using an iterative method with the help of PyLab wrapping over Numpy, Scipy, and Matplotlib in Python.
To solve for the potential function of a circular voltage probe on a square resistive sheet, firstly the objects referring to the physical elements and dimensions were initialized. The potential function was initialized with appropriate boundary conditions. Then, a formulation of the differential equation in the array index space allowed for estimating the potential using the cell's neighbours. The update rule was applied over many iterations to get the potential. Convergence to the correct potential was verified from the decreasing error plot. The potential was visualized on surface and contour plots. The current was then calculated and plotted in vector form. Most of the current is in the bottom half of the plate. The heat plot was then obtained using a similar iterative method.

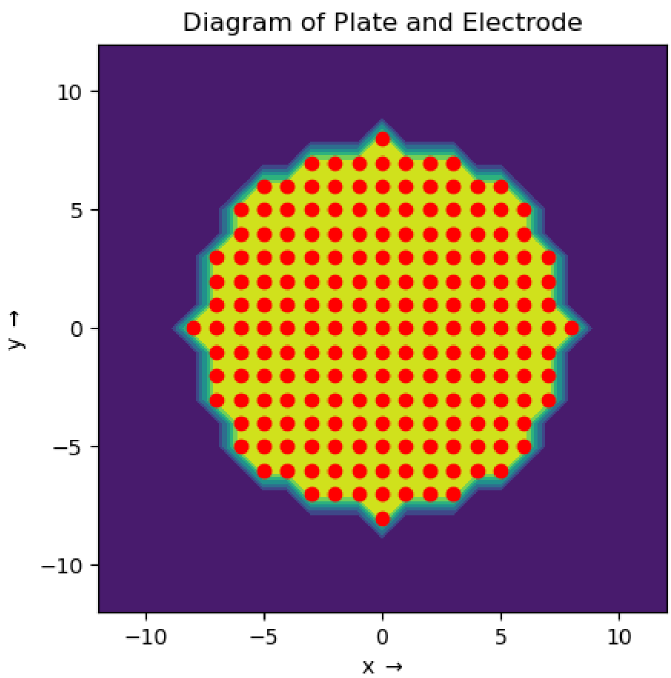
The detailed process and observations from each of the plots is given below.

1 Initial Setup

As specified by the question paper, a $N_x \times N_y = 25 \times 25$ potential array was initialized, and the positions of the circular voltage probe ii were procured using the following line

```
ii = np.where(X*X + Y*Y <= Nr*Nr)
```

The contour plot of this potential was obtained



2 Iterative Updation

The relation between true potentials in the difference equation format is used as the update equation. A vectorized approach was adopted to speed up execution. The boundary conditions were also enforced using vectorized code, and the error at each iteration was tracked.
The update rule is given by

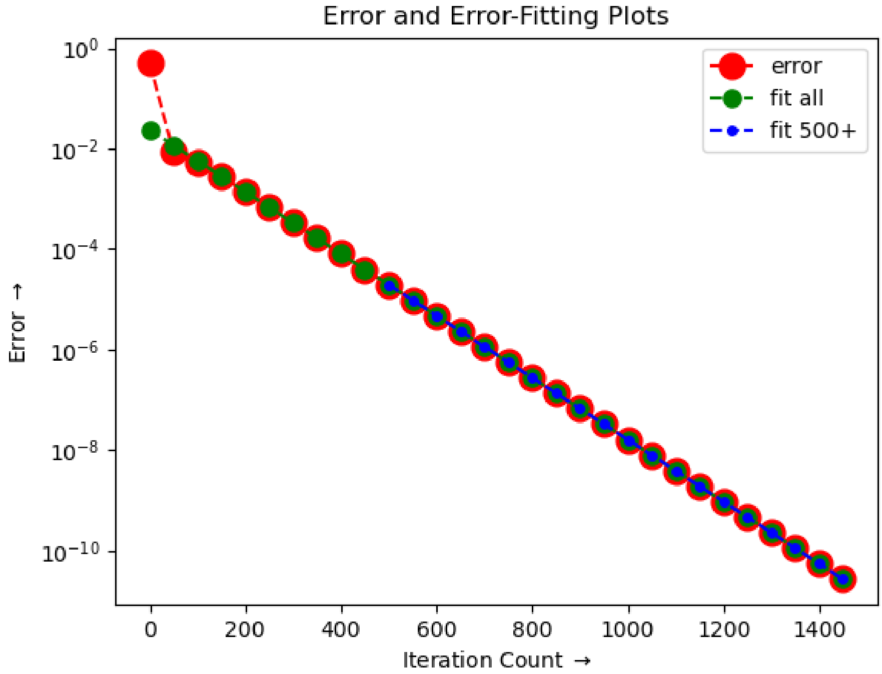
$$\phi_{i,j} \longleftarrow \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}}{4}$$

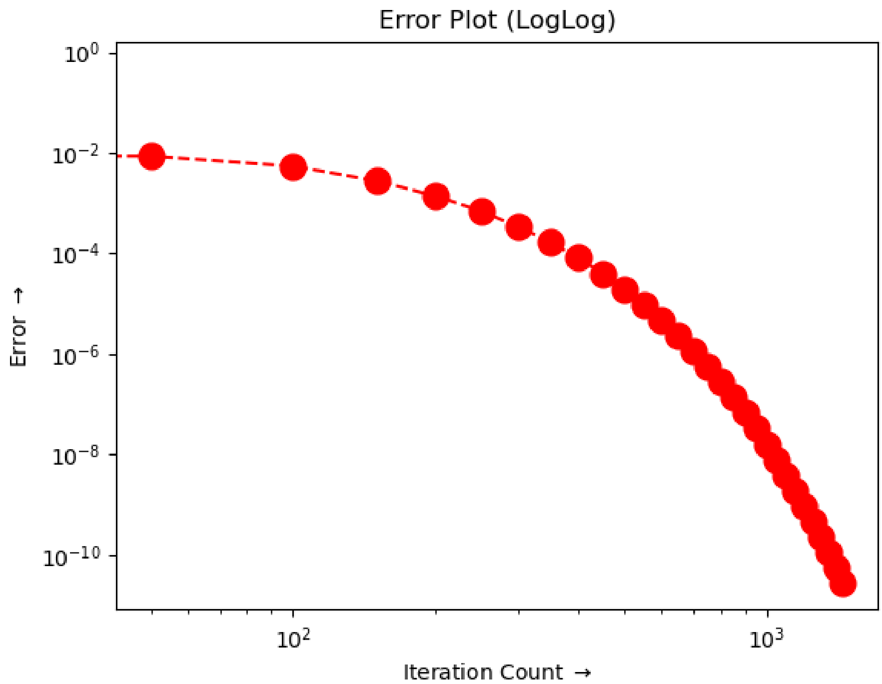
The code is given below

```
for k in range(Niter):
oldphi = phi.copy()
phi[1:-1, 1:-1] = 0.25*(oldphi[1:-1, 0:-2] + \
                        oldphi[1:-1, 2:] + \
                        oldphi[0:-2, 1:-1] + \
                        oldphi[2:, 1:-1] \
                        )

# Boundary update
phi[1:-1, 0] = phi[1:-1,1]          # Left Boundary
phi[1:-1, -1] = phi[1:-1, -2]       # Right Boundary
phi[0, 1:-1] = phi[1, 1:-1]         # Top Boundary
phi[-1, 1:-1] = 0                   # Bottom Boundary (Ground)

phi[ii] = 1.0                       # Wire Boundary

errors[k] = (abs(phi-oldphi)).max()
```

3 Error of Iterative Update

The errors were plotted on semilog and loglog scales. The semilog showed a straight line, indicating an exponential behaviour of error. The errors were fitted using two exponential functions, one using all the values and the other ignoring the first 500 values which did not conform to the straight line semilog.



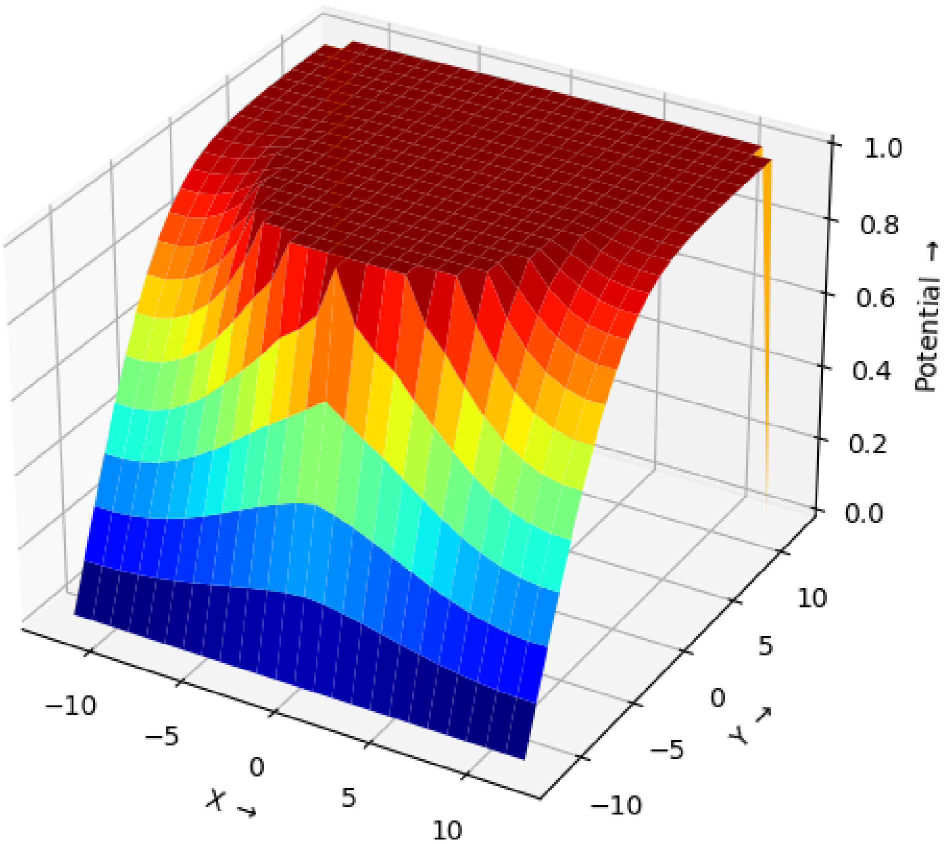


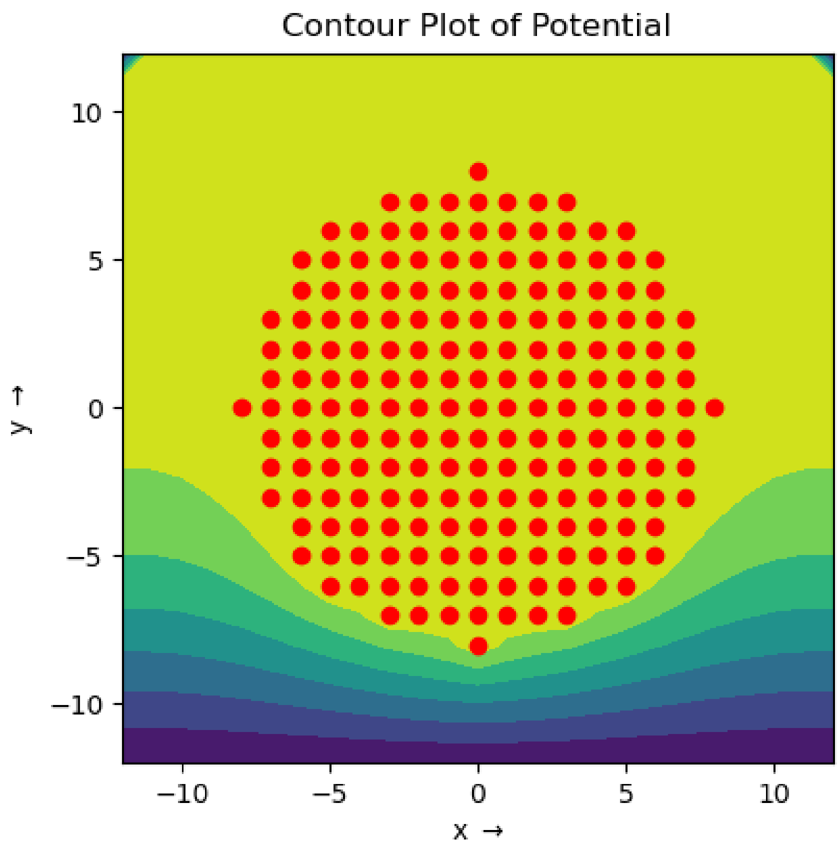
4 Stopping Condition

We find a bound on the error using the formula $-\frac{A}{B}exp(B(N+0.5))$. The function is exponential in $Niter$, implying that the error decreases exponentially with number of iterations. In an advanced implementation of iterative Laplace solving, instead of a fixed number of iterations, this error bound can be used as the stopping condition.

5 Visualizing the Potential

The estimated potential is plotted using surface and contour plots. The potential is highest at the wire and the area above it, while it decreases towards the bottom


1.0
0.8
0.6
0.4
0.2
0.0
Potential →
−10
−5
0
5
10
x →
10
5
0
−5
−10
y →


Contour Plot of Potential
10
5
0
−5
−10
y →
−10
−5
0
5
10
x →

6 Visualizing the Current

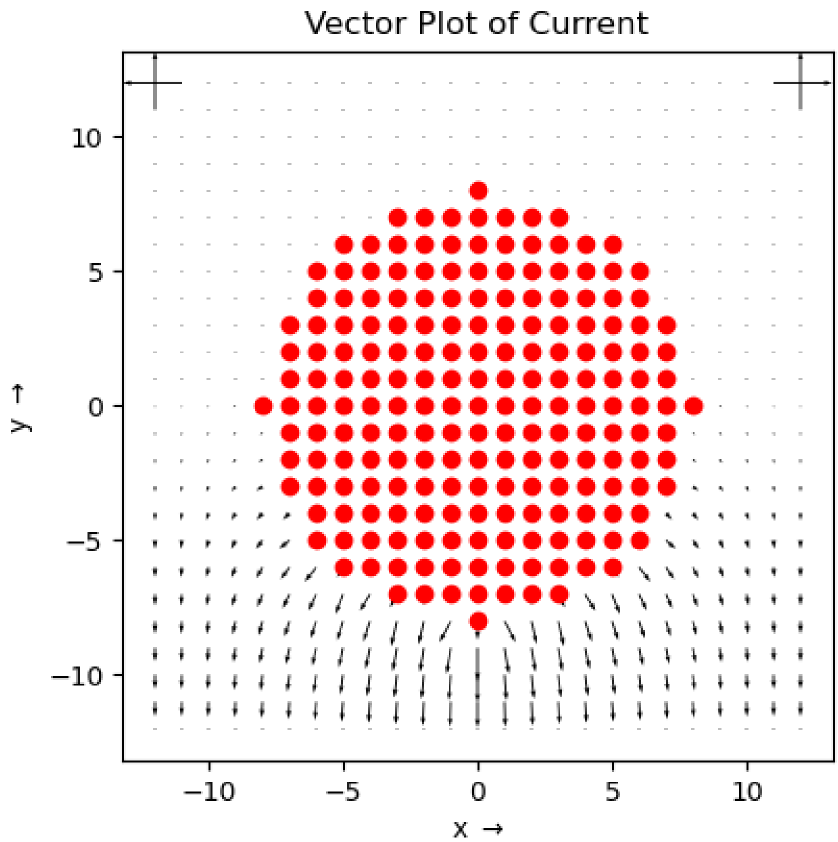
The current is found using the equations

$$J_x = -\sigma \frac{d\phi}{dx} \qquad J_y = -\sigma \frac{d\phi}{dy}$$

Taking $\sigma = 1$, in the index space this translates to

$$J_{x,(i,j)} = \frac{1}{2}\phi_{i,j-1} - \phi_{i,j+1} \qquad J_{y,(i,j)} = \frac{1}{2}\phi_{i-1,j} - \phi_{i+1,j}$$

A quiver plot (vector plot) of the current was obtained



Most of the current exists in the bottom half of the plate. The current is nearly zero in the top half of the plate. This can be reasoned using equivalent resistance – paths with more resistance contain lesser current than paths parallel to it with lower resistance. Since the ground is at the bottom of the plate, the shortest and therefore the least resistance path is to move downwards.

7 Visualizing the Temperature

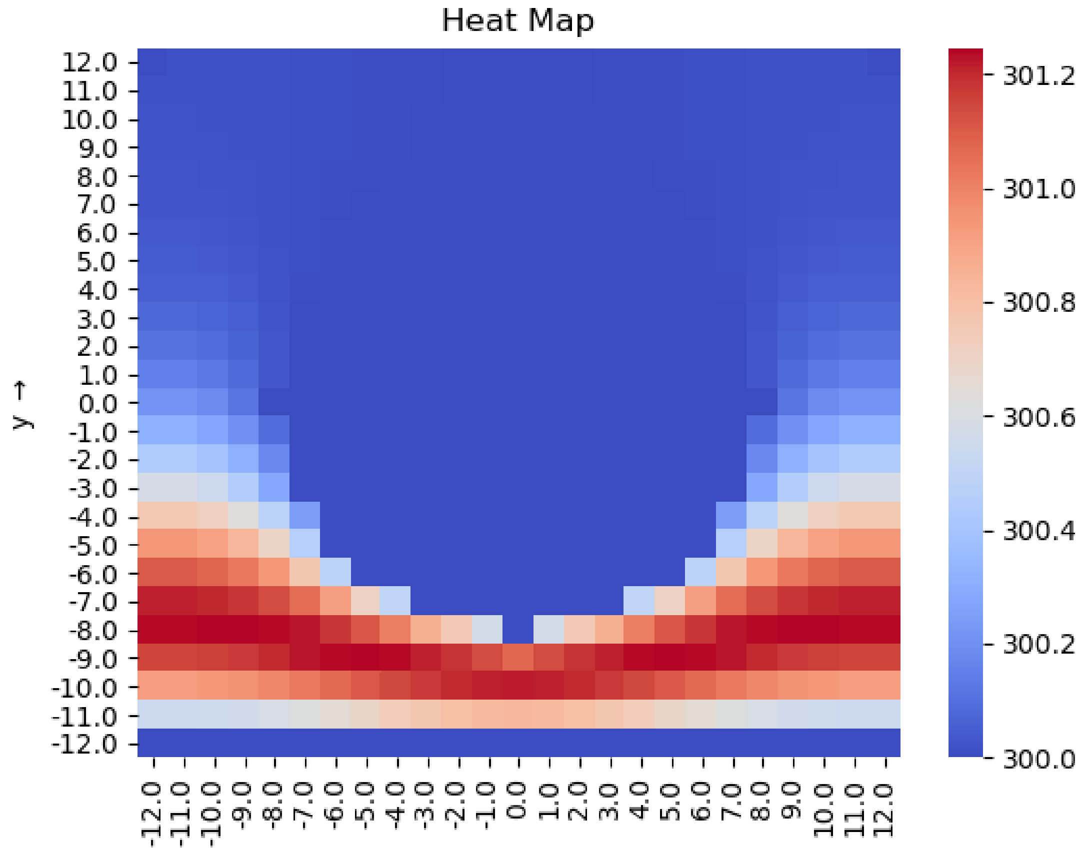
The differential equation (a Poisson Equation) given as a hint is solved using the same iterative method for the Laplace Equation, but with an extra term in the update equation. It is assumed that heat and electric conductivities are such that $\frac{\sigma}{k} = 10$ everywhere. The scaling factor of 10 is arbitrarily chosen for better plot scale. Expanding the del operator, we get

$$\frac{d^2T}{dx^2} + \frac{d^2T}{dy^2} = 10|J|^2$$

The update rule is given by

$$T_{i,j} \longleftarrow \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} + 10|J|^2}{4}$$

Using this, we obtain the following heatmap using the Seaborn library



Conclusion

This assignment explored the iterative method of solving the Laplace Equation. The potential, current and heat plots for the given configuration were obtained. It is noticed that the principle of least resistance intuitively explains the current distribution.
Through this assignment, the student has been acquainted with vectorizing tricks in Numpy, and useful plotting functions in Matplotlib. A variety of plots like semilog, 3D surface, contours and heatmaps were used to effectively convey the key ideas.