EE2703: Applied programming Lab
Assignment 5: Laplace Equation

Vishnu Varma V
EE19B059

March 24, 2021

1 Introduction

Solving Laplace equation in 2D and obtaining the potential configuration in a region, solving currents in a resistor.

2 Laplace Equation and its Approximation

The Laplace Equation to be solved to obtain the potential configuration is :

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\partial^2 \phi}{\partial y^2}$$

The above equation can be approximated as :

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

Thus if the solution holds, the potential at any point should be the average of its neighbours. We update ϕ using the above function in our program.

3 Defining the Parameters

The Dimensions of the plate is taken as 25 x 25 grid with a circle of radius 8 and number of iterations is taken as 1500 and these parameters can be changed using command line arguments, and the origin is located at the midpoint of the plate. The Potential inside the circle is 1V.

4 Allocating and Initialising the Potential

The potential array is allocated with Ny rows and Nx columns. meshgrid function is used to convert the x and y points into two arrays one with

x coordinates and the other with y coordinates with the same shape of the array. Using where command we find out the points in the circle, the potential at these points are initialised to 1V .
Now the Initial contour plot of the potential function is plotted which looks as follows:

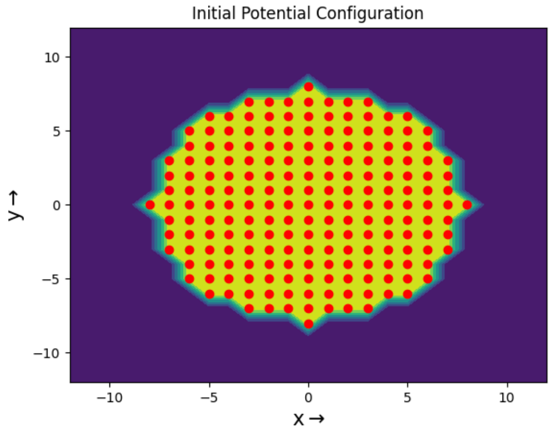

Figure 1: Initial Potential Configuration

5 Updating the Potential Function

Using the approximation for Laplace equation as mentioned earlier and boundary conditions, One side of the plate is grounded while the remaining are floating. So assigning 0V to grounded side and the potential at open ends assigned to the potential adjacent to it, because the normal derivative of potential is to be set to zero as at the boundaries the gradient of ϕ should be tangential. The potential within the circle is also modified so setting it back to 1 V. and finding the error in potential as a function of number of iterations The following code updates ϕ imposes boundary conditions and finds the error in potential:

```
for i in range(Niter):
    oldphi = phi.copy()
    phi[1:-1,1:-1] = 0.25*(oldphi[1:-1,2:]+oldphi[1:-1,0:-2]
                     +oldphi[2:,1:-1]+oldphi[0:-2,1:-1])
    phi[1:-1,0] = phi[1:-1,1]
    phi[1:-1,-1] = phi[1:-1,-2]
    phi[0,1:-1] = phi[1,1:-1]
    phi[-1,1:-1] = 0
    phi[ii] = 1.0
    error[i] = (abs(phi-oldphi)).max()
```

6 Error in Potential

Error obtained above is plotted against the number of iterations in linear, semilog and loglog scales. The Plots are as follows:

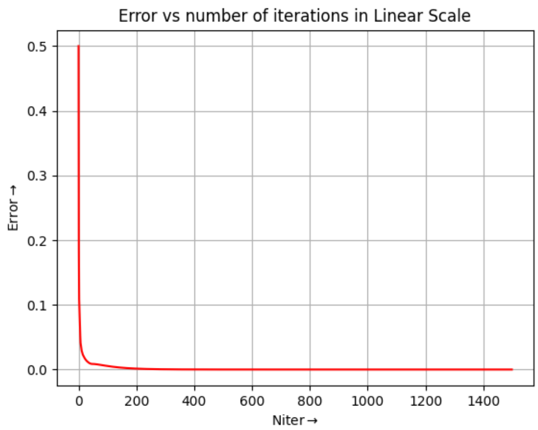

Figure 2: Error vs number of iterations Linear Scale

Plotting every 50th point, to see individual data points

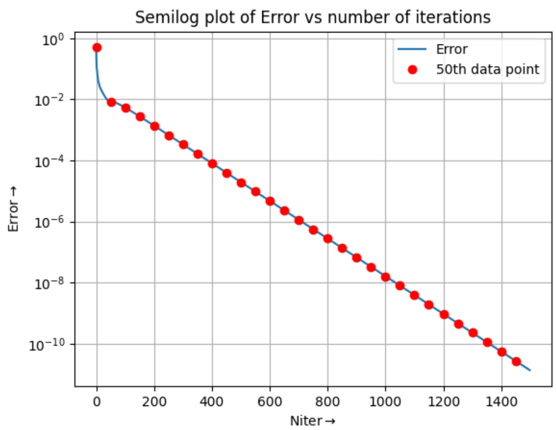

Figure 3: Error vs number of iterations Semilog Scale

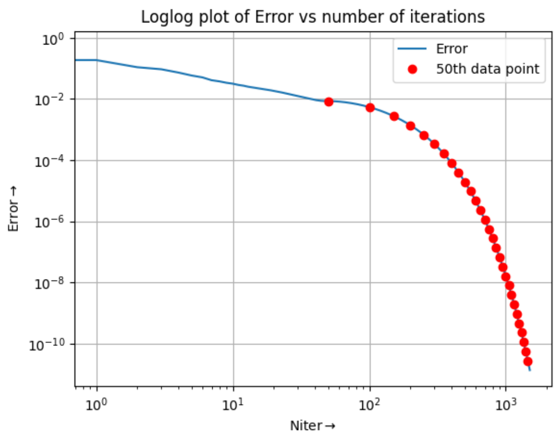

Figure 4: Error vs number of iterations Loglog Scale

The error seems to be exponential, anytime we have something like an exponential we extract the exponent.

$$y = Ae^{Bx}$$

Thus ,

$$logy = logA + Bx$$

$logA$ and B can be estimated using the least squares method. But it resembles exponential at larger number of iterations. So fitting the entire vector of errors ($fit1$) and error entries after 500th iteration ($fit2$) Now plotting $fit1$,$fit2$ and the true value of the error against number of iterations in loglog scale gives us:

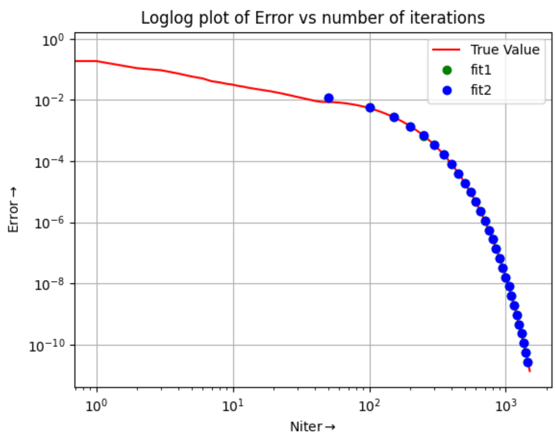

Figure 5: Loglog Plot of Error vs Number of iterations

From the graph both $fit1$ and $fit2$ coincide with very little difference hardly noticeable.

7 Stopping Condition

The maximum error scales as

$$error = Ae^{Bk}$$

Summing up the terms gives:

$$Error = -\frac{A}{B}exp(B(N+0.5))$$

This Cumulative error plotted against the number of iterations in a Semilog scale yield the following plot:

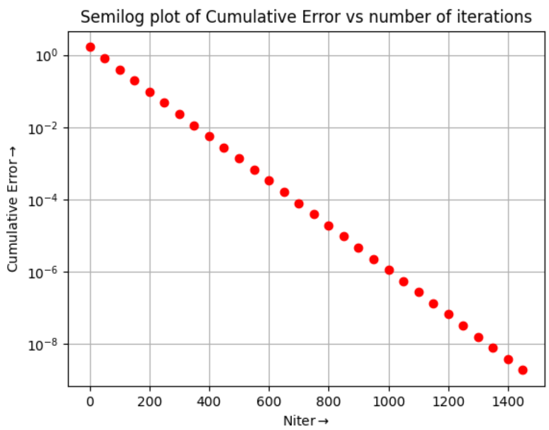

Figure 6: Semilog plot of Cumulative Error vs number of iterations

8 Surface Plot of Potential

3D plot of the potential variation is plotted using plot surface function.
The code to plot is as follows:

```
fig = plt.figure(6)
#Axes3D is the means to do a surface plot
ax = p3.Axes3D(fig)
plt.title('The 3-D surface plot of the potential')
#plot_surface function does the plotting
surf=ax.plot_surface(Y, X, phi.T, rstride=1, cstride=1, cmap=plt.cm.jet)
plt.xlabel(r'x$\rightarrow$',fontsize=15)
```

```
plt.ylabel(r'y$\rightarrow$',fontsize=15)
ax.set_zlabel(r'$\phi\rightarrow$',fontsize=15)
plt.show()
```

The plot is as follows:

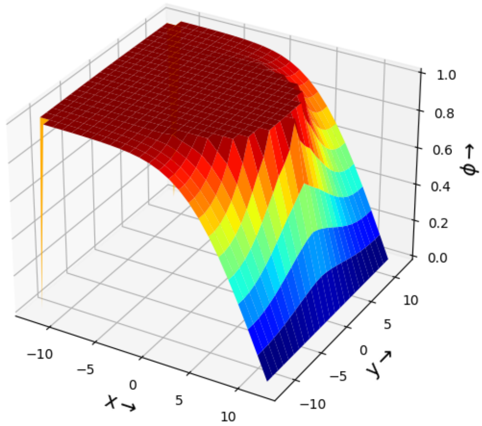

Figure 7: The 3-D surface plot of the potential

9 Contour Plot of the Potential

After updating and imposing the boundary conditions now the potential function is different from the inital potential. The code to plot the contour plot of potential is as follows:

```
# xn and yn are the coordinates of the interior of the circle
plt.plot(x[xn],y[yn],"ro")
plt.contourf(Y,X[::-1],phi)
plt.xlabel(r'x$\rightarrow$' , fontsize=15)
plt.ylabel(r'y$\rightarrow$',fontsize=15)
plt.title('Potential Configuration')
plt.show()
```

The Potential function obtained is as follows:

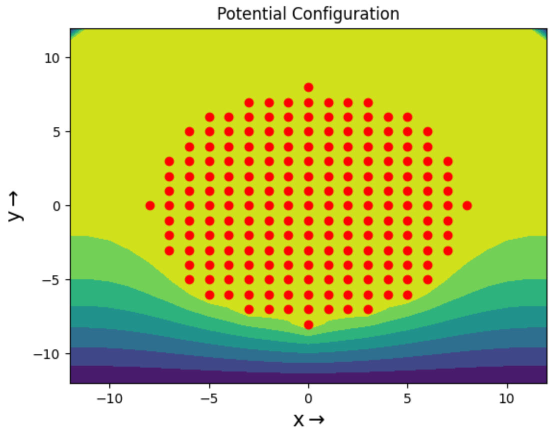

Figure 8: Final Potential Configuration

10 Vector Plot of Currents

The current density as a function of potential is expressed as follows:

$$J_x = -\frac{\partial \phi}{\partial x}$$

$$J_y = -\frac{\partial \phi}{\partial y}$$

which can be approximated as :

$$J_{x,ij} = \frac{\phi_{i,j-1} - \phi_{i,j+1}}{2}$$

$$J_{y,ij} = \frac{\phi_{i-1,j} - \phi_{i+1,j}}{2}$$

Using the approximation as above and finding the current density using the following lines of code:

```
Jx[:,1:-1] = 0.5*(phi[:,0:-2]-phi[:,2:])
Jy[1:-1,:] = 0.5*(phi[2:, :]-phi[0:-2,:])
```

The Current density vector is plotted using quiver function. The plot is as follows:

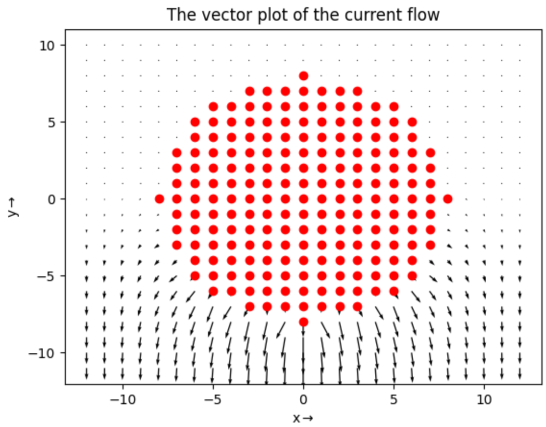

Figure 9: Current Density

We can observe that all the current flows through the bottom part of the plate which is grounded, the charge carriers move from the electrode towards ground potential via the lower half of the plate avoiding any other longer path, this part of the plate would be the hottest.

11 Conclusion

The Laplace equation was solved using an approximation and imposing boundary conditions. The error plotted against number of iterations show that it is reducing but very slowly. Thus the given method of solving is inefficient. A loglog plot of error gives a reasonably straight line upto about 500 iterations and beyond that exponential. Fitting the error data to an exponential two fits are obtained one using all the error entries and the other using entries after 500th iteration, but the fits almost coincide with very little difference. The Current Density plot shows that the current flows through the botom half of the plate towards the ground potential, and the current is perpendicular to the surface of electrode and the plate.