Assignment No. 5

Srivenkat A (EE18B038)

March 4, 2020

1 Introduction

The Laplace equation in 2 dimensions can be written as:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

The potential function in the metal plate follows this equation and numerically solving it gives the distribution as:

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

assuming a coordinate grid of 25x25 size and a differential element is 1 unit of the grid.
The boundary conditions are:

- phi = 1.0 inside the region of the metal wire
- phi = 0 on the bottom side.

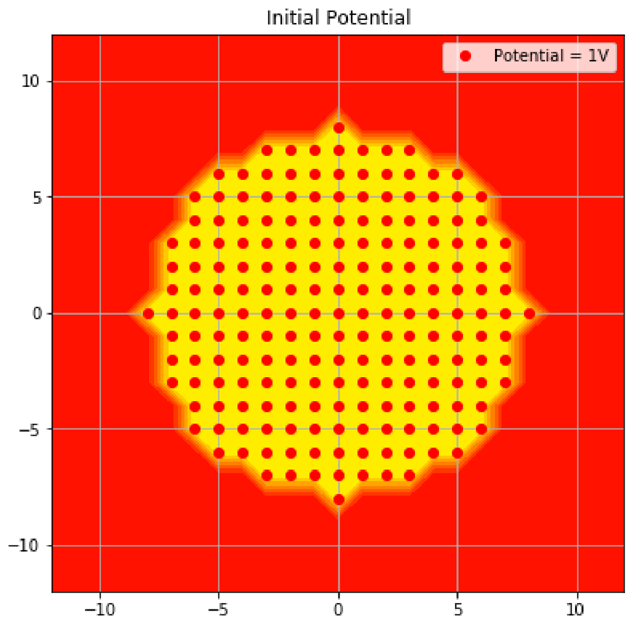
2 Solving the Laplace Equation

The following code updates the matrix phi over *Niter* (N iterations). It also appends the subsequent error incurred in each iteration in the array *err*. The boundary conditions are reinforced after every iteration.

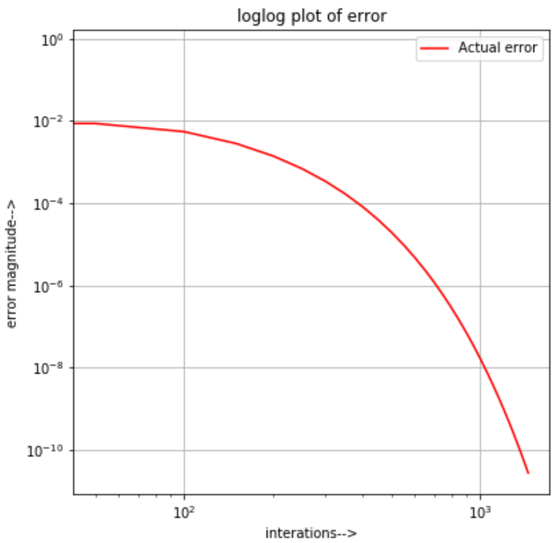
```
phi = np.zeros(shape=[Nx,Ny])
x = np.arange(int(-1*Nx/2),int(Nx/2)+1)
y = np.arange(int(-1*Ny/2),int(Ny/2)+1)
Y,X = np.meshgrid(y,x)

ii = np.where(X*X + Y*Y <= radius*radius)
phi[ii] = 1.0
phi[1:-1,Ny-1] = 0.0
```

```
for k in range(Niter):
    oldphi = phi.copy()
    phi[1:-1,1:-1] = 0.25*(phi[0:-2,1:-1]+phi[2:,1:-1]+phi[1:-1,2:]+phi[1:-1,0:-2])
    phi[:,0] = phi[:,1]
    phi[:,Nx-1] = phi[1:-1,Nx-2]
    phi[0,1:-1] = phi[1,1:-1]
    phi[Ny-1,1:-1] = phi[Ny-2,1:-1]
    phi[ii] = 1.0
    phi[1:-1,Ny-1] = 0.0
    err.append(abs(phi-oldphi).max())
```

The initial contour plot of the potential distribution is as follows. The 1V region is denoted by red dots.



With increasing number of iterations, the error in convergence can be plotted in a log-log plot as



From the log-log plot, it is observed that the graph is approximately linear for the first 200-300 iterations.

Since the error can be predicted to exponentially reduce with the no. of iterations, we try to model the error function as

$$y = Ae^{Bx}$$

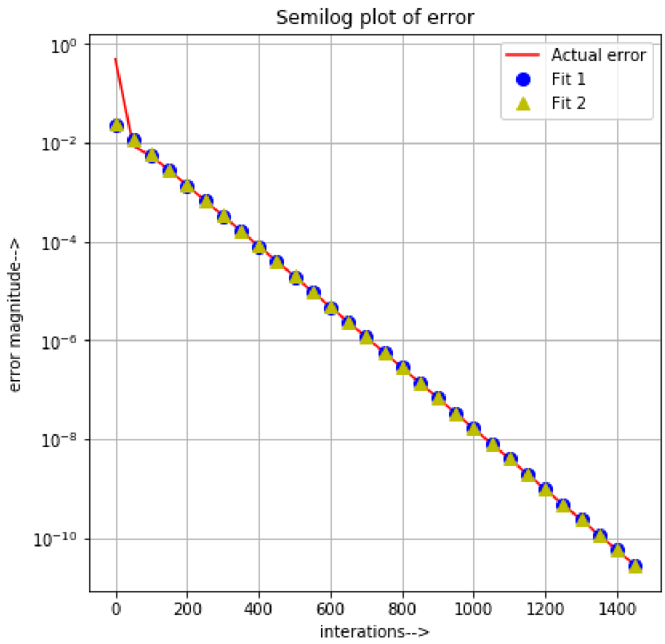
$$logy = logA + Bx$$

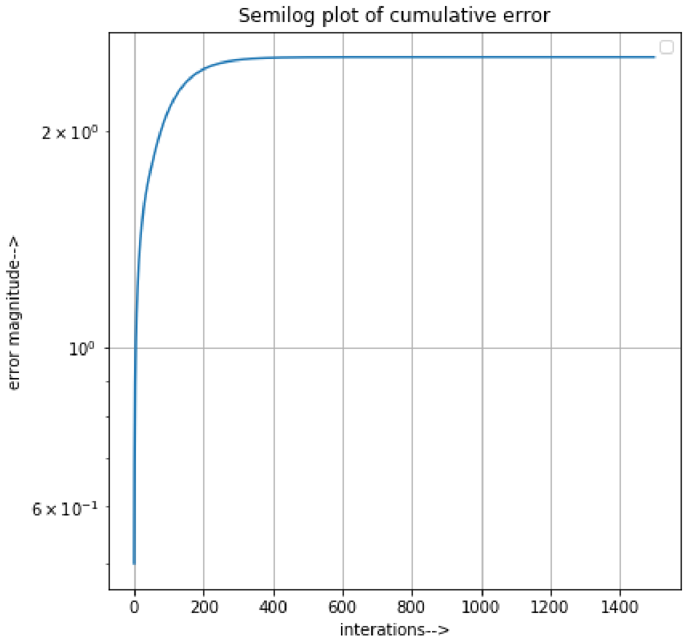
$logA$ and B can be estimated using the least squares method. The following code block accomplishes the same :

```
log_y_mat = np.log(np.transpose(err))
x_mat = np.c_[np.arange(0,Niter),np.ones(shape = [Niter,1])]
fit1 = sp.lstsq(x_mat,log_y_mat)[0]
fit2 = sp.lstsq(x_mat[500:-1],log_y_mat[500:-1])[0]
```

Fit1 is the line obtained by fitting the entire array while Fit2 is obtained for the array beyond 500 iterations. Plotting them in a semilog plot we get:

where the red line is the actual fit, the circles is the exponential fit for higher iterations and the triangle marker is an exponential fit for for the entire set. It can be noticed that they almost coincide.
Also plotting the cumulative error in predicting the distribution of phi,





The upper bound for the error estimated with each iteration is given by

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

From the cumulative error plot, the graph saturates after around 500 iterations, So, to get a considerable estimate of the distribution, only 500 iterations can be done.

After 1500 iterations, The surface plot of the potential function can be obtained by

```
surf = ax.plot_surface(Y,-X, phi, rstride=1, cstride=1,cmap='jet')
```

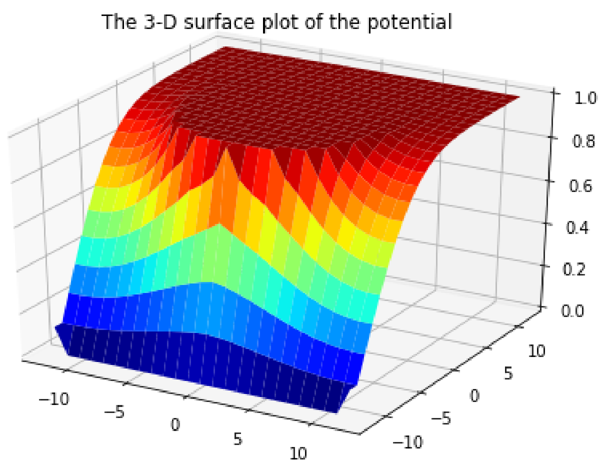
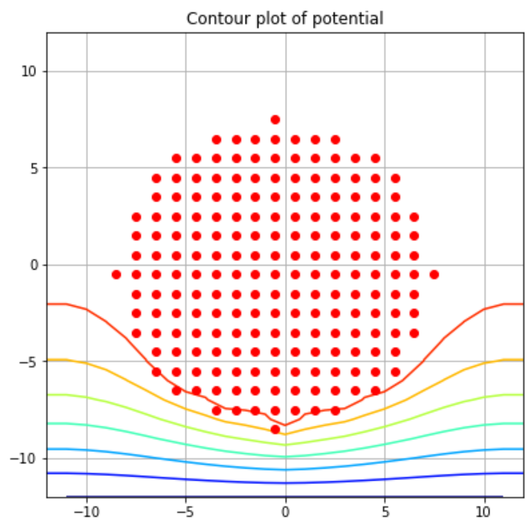


Figure 1: Caption

The 2-d projection of the surface plot can be obtained by the contour plot as generated by




```
fig4.contour(Y,X,phi,cmap='jet')
```

Solving for the vector expressions of current,

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

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

Numerically, this can be expressed as

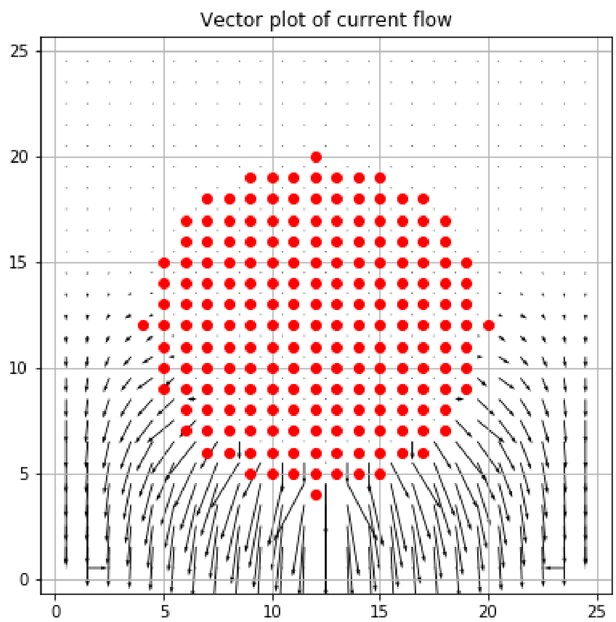
$$J_{x,ij} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2}$$

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

The following code block evaluates the currents J_x and J_y

```
Jx = np.zeros(shape=[Ny,Nx])
Jy = np.zeros(shape=[Ny,Nx])

Jx[1:-1,:] = 0.5*(phi[2:,:]-phi[:-2 ,:])
Jy[:,1:-1] = 0.5*(phi[:,:-2]-phi[:,2:])
fig5.quiver(y+Ny/2,x+Nx/2,Jy[::-1,:],Jx[::-1,:])
```

Plotting the direction of current with arrows, the obtained plot is



From the plot it is obtained that almost the entire current is confined in the lower half of the metal plate.

3 Conclusion

In this method of solving the Laplace equation, the error decays at a highly gradual pace. So, it takes a large number of iterations to reduce the error considerably. So, it is an ineffective way to solve for the distribution.
From the plot of the distribution, it can be noticed that current is mainly confined between the electrode and the bottom of the plate and hence only the lower half gets considerably hot due to ohmic losses.