

EE2703: Assignment 5

Tanay Dixit
EE19B123

March 25, 2021

Aim

The aim is to find the solution to potential on a plate by solving the Laplace equation

Introduction

We consider the resistor problem where a cylindrical wire is soldered to the middle of a copper plate and its voltage is held at constant value. The bottom side of the plate is grounded.

We make use of the 3 fundamental equation to solve this problem

The Continuity Equation:

$$\nabla \cdot \vec{j} = -\frac{\partial \rho}{\partial t}$$

Ohms Law:

$$\vec{j} = \sigma \vec{E}$$

On combining these 2 equations and applying constraints we get

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (1)$$

An approximate solution for the above equation for a 2-dimensional grid of points would be

$$\phi_{i,j} = \frac{\phi_{i,j-1} + \phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j+1}}{4} \quad (2)$$

i.e. the solution at any point is a average of values at neighbouring points. The algorithm implemented makes use of the above equation to update ϕ over many iterations.

To define the plate dimensions and wire size we take inputs from the user and use the following code to setup the configuration.

```
1 def grid(self):
2     """
3     Function to setup the grid
4     """
5     phi = np.zeros((self.Ny, self.Nx))
6     #self.x1 = np.linspace(-(self.Ny-1)/2, (self.Ny-1)/2, self.Ny)
7     #self.y1 = np.linspace(-(self.Nx-1)/2, (self.Nx-1)/2, self.Nx)
8     self.x1 = np.linspace(-0.5, 0.5, self.Ny)
9     self.y1 = np.linspace(-0.5, 0.5, self.Nx)
10    self.Y, self.X = np.meshgrid(self.y1, self.x1)
11    id = np.where((self.X**2 + self.Y**2) < (self.radius/self.Nx)**2)
12    phi[id] = 1.0
13    return phi, id
```

Potential function

We consider the potential functions to be an NxN dimensional array. The coordinates are defined such that the origin is in the center

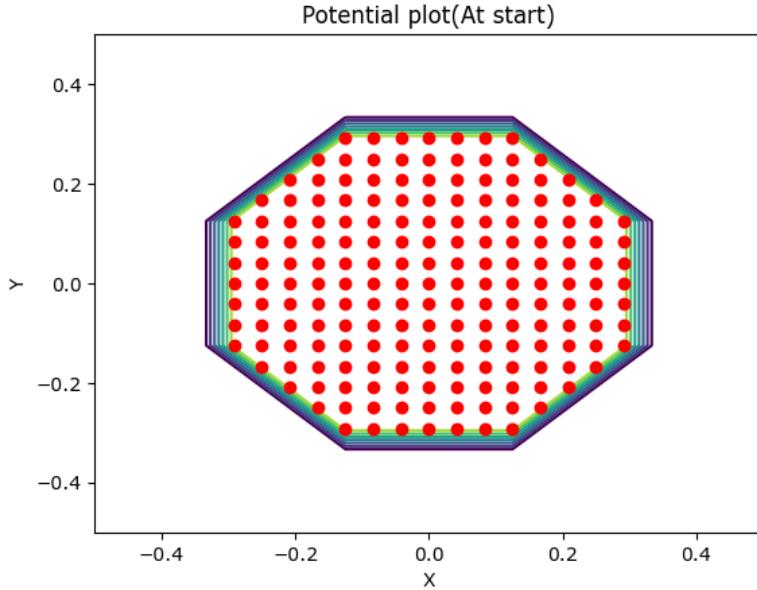


Figure 1: Plot of the Initial potential configuration

In the above plot red dots refer to the points on the wire and rest are at 0 potential hence contours are all attached to the wire's boundary.

We approximate the solution to 1 with 2, as a result it means that potential at any point is just the average of its neighbours, we keep iterating this process for N times as defined for the user. But every iteration we maintain the boundary conditions, and also the side of the plate which is grounded must remain 0 all the time and same holds for the surface in contact with the wire. Also note on edges the current needs to be tangential hence the gradient ϕ also needs to be tangential hence its normal components should be zero on the edges

The following functions are used to update the potential function

```
1 def step(phi, ophi):
2     #phi_new = 1/4 *(left +right +top +bottom)
3     phi[1:-1,1:-1] = 0.25*(ophi[1:-1,0:-2]+ophi[1:-1,2:]+ophi[0:-2,1:-1]+ophi
4     [2:,1:-1])
5     return phi
6
7 def callback(phi, ids):
8     """
9         function to reset the boundary conditions
10    """
10    phi[1:-1,0] = phi[1:-1,1]    #left side boundary condition
```

```

11     phi[1:-1,-1] = phi[1:-1,-2] #right side boundary condition
12     phi[0,1:-1] = phi[1,1:-1]    #top side
13     phi[-1, 1:-1] = 0          #bottom side as its grounded
14     phi[ids] = 1.0
15     return phi
16
17 def trainer(self, epochs):
18     """
19         function used to iterate and compute the potential
20         epochs: number of times to iterate
21     """
22     error = np.empty(epochs)
23     for i in range(epochs):
24         old_phi = self.phi.copy()
25         self.phi = self.step(self.phi, old_phi)
26         self.phi = self.callback(self.phi, self.ids)
27         error[i] = np.max(np.abs(self.phi - old_phi))
28     return error

```

we iteratively call these 2 functions(*step()* and *callback()* for Niter times as defined by the user *default value is 1000* and we study the convergence of the estimate.

Error Analysis

We first plot the error(ie: maximum absolute difference between old ϕ and new ϕ at every step) with number of iterations on a linear scale and see that it changes very slowly after a while, hence we also analyse the plots in a semi-log scale and log-log scale.

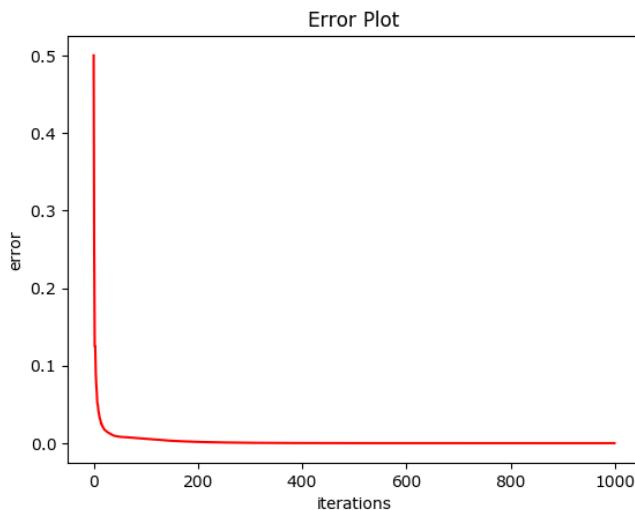


Figure 2: Error plot in linear scale

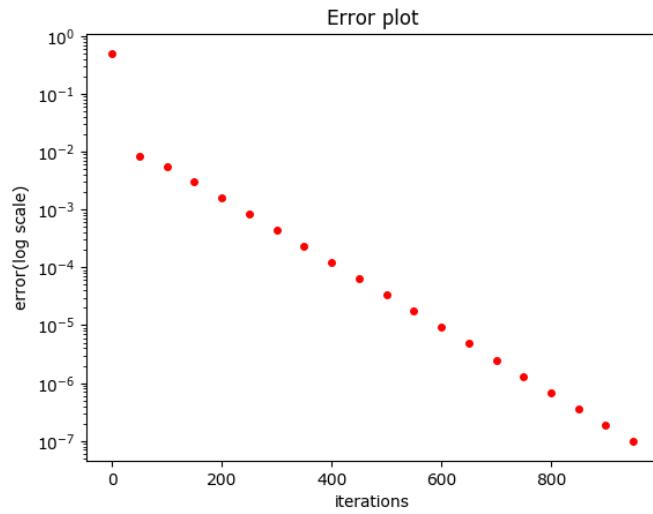


Figure 3: Error plot in semi-log scale

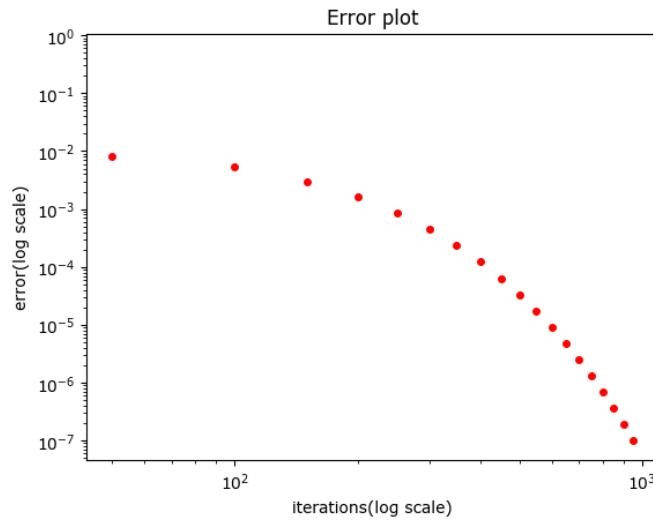


Figure 4: Error plot in log-log scale

From the above figure(s) [Figure 3](#) we notice that the error decays exponentially as the semi-log plot is linear

We study 2 different cases and fit an exponential Ae^{Bx} to the error vector in order to extract the exponents.

- error vector from 1st iteration
- error vector from 500th iteration

We solve the following equation to obtain the least square estimates of A and B using code given below

$$\log y = \log A + Bx$$

```

1 def errorFit(self,x,y):
2     logy=np.log(y)
3     x_arr = np.ones((len(y), 2))
4     x_arr[:,0] = x
5     B,logA=np.linalg.lstsq(x_arr, np.transpose(logy), rcond = None)[0]
6
7     return np.exp(logA),B

```

The following plot the 2 error fits as mentioned in

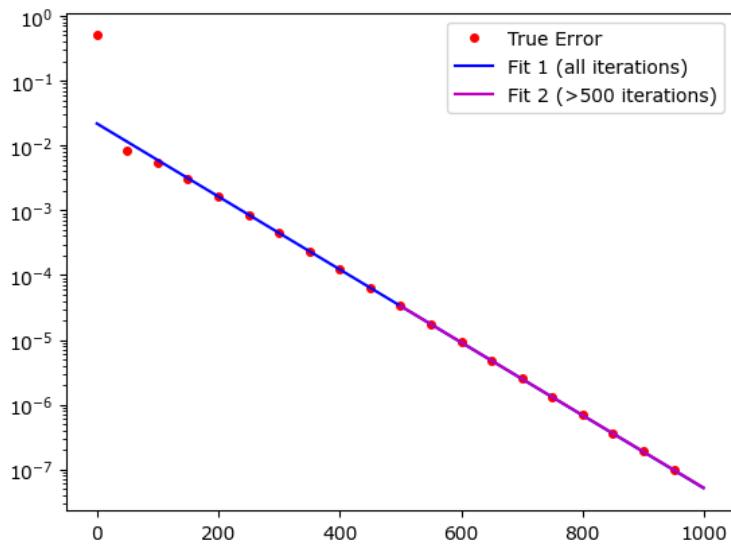


Figure 5: Error/Fit plot in semi-log scale

Stopping Criteria

We notice that even though error is very small of the order of $1e - 5$ it can be misleading as what matter is the time required to decay by 1 time constant, which is large. One can even see in [Figure 3](#) it decays slowly after 500 iterations

Time constant for default values of Nx,Ny,radius and Niter is 71.87

this value is subject to change based on user input

We calculate the cumulative error and show that even though error is decreasing at a steady rate yet the cumulative error is large hence this method is **not** a good way to approximate solving the laplace equation.

$$CumulativeError = -\frac{A}{B} \exp B(N + 0.5) \quad (3)$$

Potential Plot

After completing the iterations as mentioned previously we plot the surface potential as a 3D plot with the potential value on the z axis

The 3-D surface plot of the potential

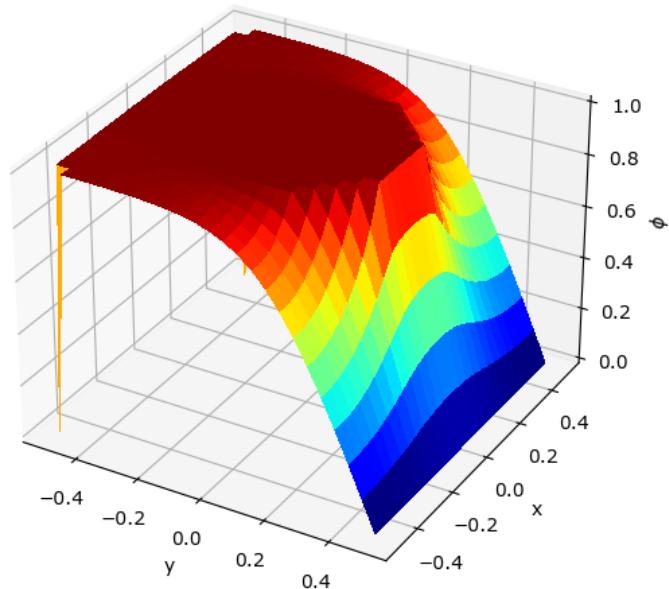


Figure 6: 3D potential plot with ϕ on z axis

From the above plot we can see there is a gradual decay of potential from the wire which is at 1V to ground ie:0V

Contour plots

We do a contour plot of the plate to visualize the equi potential lines on the plate, lighter lines are higher value(1V) while darker are lower(0V)

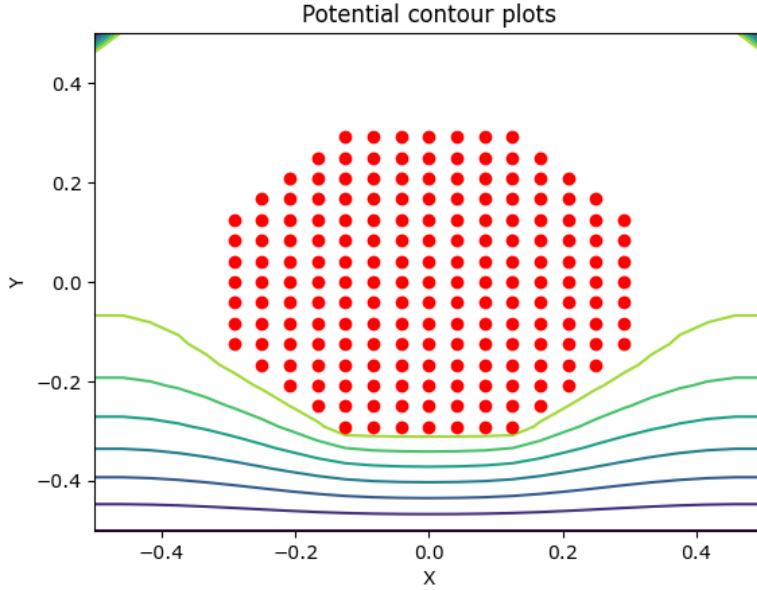


Figure 7: Contour Plot

Analysing Currents

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

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

These equations translated to numerical gives

$$J_{x,ij} = 0.5 * (\phi_{i,j-1} - \phi_{i,j+1})$$

$$J_{y,ij} = 0.5 * (\phi_{i-1,j} - \phi_{i+1,j}) \quad (4)$$

We obtain the J_x and J_y and plot the current vectors using the quiver command

```

1 def currents(self):
2     """
3         function computes the current vectors ie: Jx , Jy
4     """
5     self.Jx = np.zeros((self.Ny,self.Nx))
6     self.Jy = np.zeros((self.Ny,self.Nx))
7
8     self.Jx[:,1:-1] = 0.5*(self.phi[:,0:-2]-self.phi[:,2:])
9     self.Jy[1:-1,:] = 0.5*(self.phi[2:,:]-self.phi[0:-2,:])
10
11    return self.Jx, self.Jy

```

From the current density plot, we notice that hardly any current flows through the top part of the wire. We observe that as the lower surface is grounded and the easiest way for current

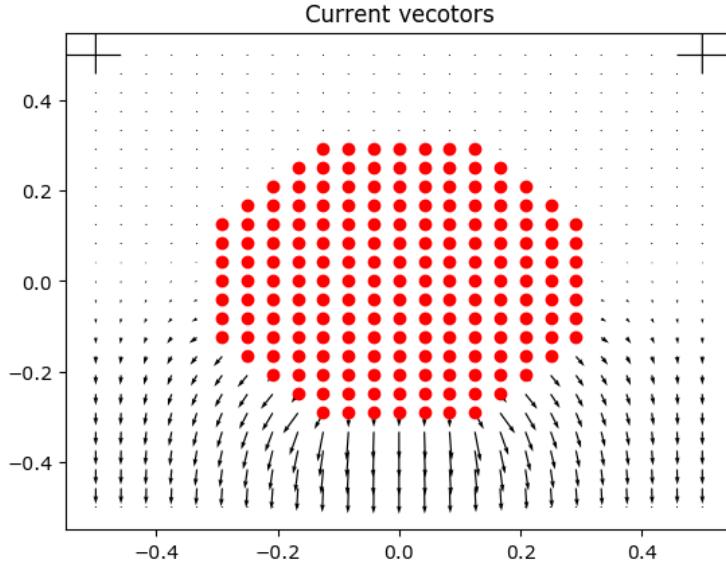


Figure 8: Current Flow plot

to flow would be directly through the lower half plane. Thus all current vectors are located in the lower half (half with the grounded side) and thus avoid they longer more resistive path ie: from top half to the wire

Temperature Plots

Having understood how currents flow in the plate we now study the heating pattern. As the primary cause of heating is ohmic loss we expect to see a majority of heating happening where currents are present. To find out Temperaure variation we solve the following equation using the same Laplace approximation as done before.

$$\nabla^2 T = -\frac{q}{\kappa} = -\frac{|J|^2}{\sigma \kappa} \quad (5)$$

We apply the constraints that Temperature at the wire boundary and ground is 300K and at the edges

$$\frac{\partial T}{\partial n} = 0$$

From the above plot we see that our intuition was correct and the majority of heating happens where current are flowing

Temperature Plot

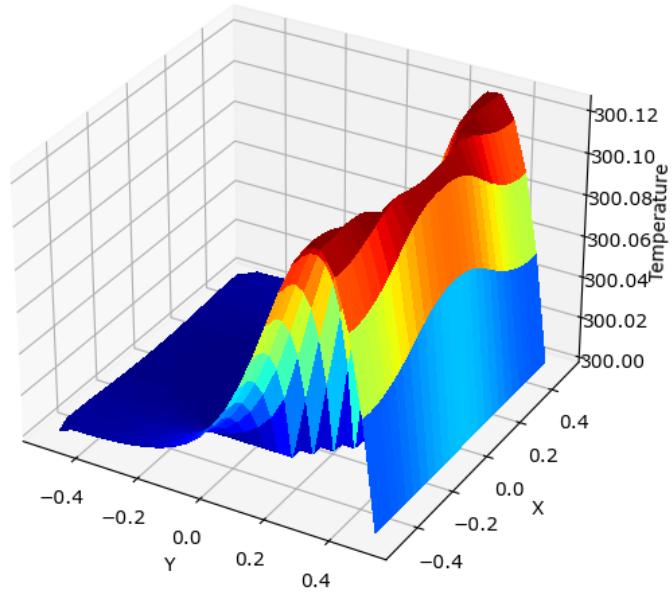


Figure 9: Temperature Variation PLOT

Conclusion

Using a finite differentiation approximation , we find a solution to Laplace's equation for a given system. The error is seen to decay very slowly and hence the chosen method of solving Laplace's equation is inefficient. On analysing the quiver plot of the currents , we notice that the current was mostly restricted to the bottom of the wire and was perpendicular to the surface of the electrode and the conductor. As a result heating effect is also mostly seen on the lower half of the plate.