Assignment5-EE2703

Nihal Gajjala

March 2021

Introduction

The objective of this assignment is to understand the flow of current in a resistor and analyse which part of the resistor gets heated the most. We calculate the potential at every point in the resistor using poisson equation.

Numerical Solution

Laplace's equation in 2-dimension can be written in Cartesian coordinates as

$$\frac{\partial^2 \phi}{\partial^2 x} + \frac{\partial^2 \phi}{\partial^2 y} = 0 \tag{1}$$

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

$$\phi_{i,j} = \frac{\phi_{i,j-1} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i+1,j}}{4} \tag{2}$$

The potential at any point is the average of potential values at its neighbouring points. At each point, we replace the potential by the average of its neighbours. We continue to iterate till the solution converges (i.e. The maximum change in elements of *phi* is less than some tolerance).

At boundaries where the electrode is present, we don't change the value of potential. At boundaries where there is no electrode, the current should be tangential because charge can't leap out of the material into air. Since current is proportional to the Electric Field, the gradient of ϕ should be tangential.

Parameters

- Nx=25 - Size along x direction
- Ny=25 - Size along y direction
- radius=8 - Radius of central lead
- Niter=1500 - Number of iterations to be performed

These parameters can be changed by the user via command-line arguments.

Potential

- Saving a copy of phi

```
# Copying phi
oldphi=phi.copy()
```

- Updating phi array

```
# Function To Update phi
def update(phi,oldpfi):
    phi[1:-1,1:-1]=0.25*(oldphi[0:-2,1:-1]+oldphi[2:,1:-1]
                        +oldphi[1:-1,0:-2]+oldphi[1:-1,2:])
    return phi
```

- Asserting boundaries

```
# Function To Assert Boundaries
def boundary(phi,index):
    phi[1:-1,0]=phi[1:-1,1]
    phi[0,1:-1]=phi[1,1:-1]
    phi[1:-1,-1]=phi[1:-1,-2]
    phi[-1,1:-1]=0
    phi[index]=1.0
    return phi
```

- Errors

```
# Errors
errors[i]=(abs(phi-oldphi)).max()
```

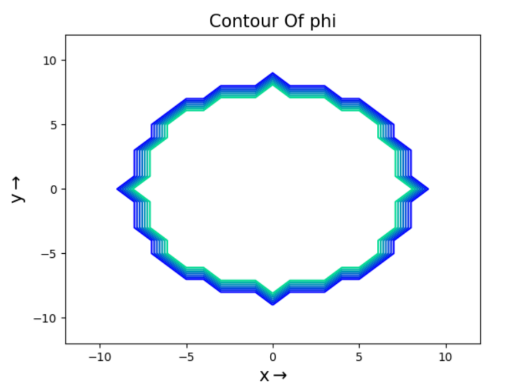

Figure 1: Contour Of phi

Error Plot

We now calculate the error for each iteration. This error is defined as the maximum difference in the value of phi in the phi array. We will plot this error for every iteration on normal axis, semilogy axis and loglog axis.

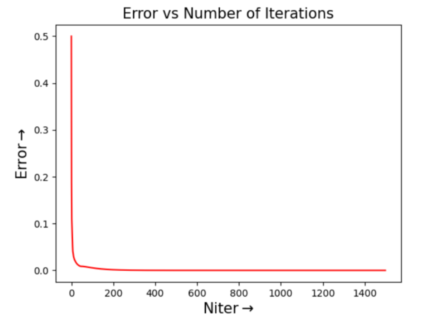

Figure 2: Error vs Niter

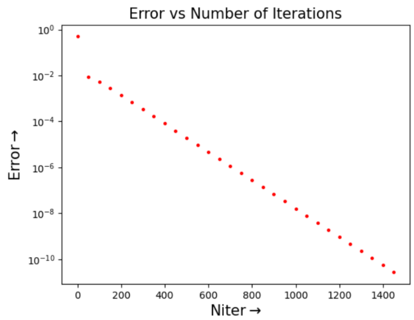

Figure 3: Error vs Niter

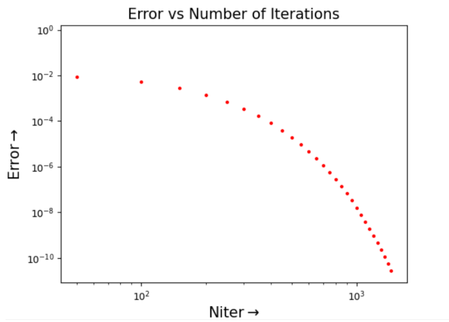

Figure 4: Error vs Niter

We see that the error is initially almost linear in the loglog plot but becomes linear in semilogy plot.

The error function appears to vary exponentially with Niter. We attempt to extract the exponent of this dependence. In particular, we attempt to fit a function of the form

$$y = Ae^{Bx} \tag{3}$$

$$logy = logA + Bx \tag{4}$$

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

```
# Function To Evaluate The Parameter Of An Exponent
def error(a,b):]
```

```
    logb=np.log(b)]
    vec=np.zeros((len(a),2))
    vec[:,0]=a
    vec[:,1]=1
    B,logA=np.linalg.lstsq(vec,np.transpose(logb))[0]]
    return (np.exp(logA),B)
```

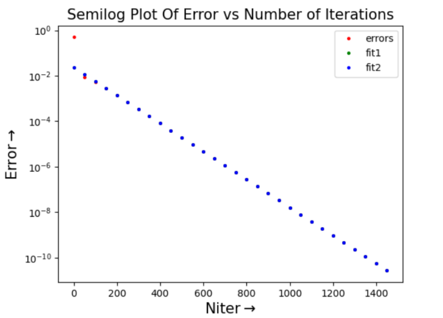

Figure 5: Error vs Niter

We see that the error becomes almost 10^{-11} after number of iterations reach 1400. The plots almost exactly coincide, with differences hardly visible.

Surface Plot Of Potential

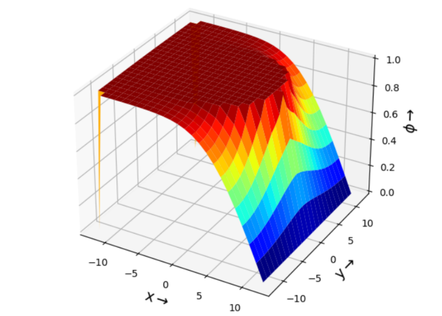

Figure 6: 3-D Plot

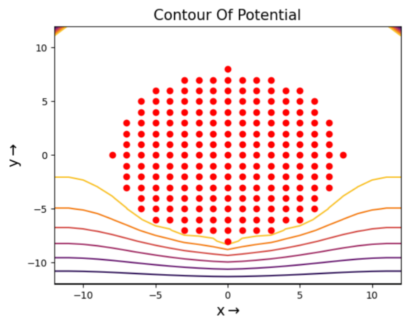

Figure 7: Contour Of Potential

In Figure 6, the values of potential is plotted in the Z-axis as a function of x and y.

In Figure 7, the values of potential is plotted in a contour plot as a function of x and y. The lines in the plot represent equipotential curves. The current should be perpendicular to these lines.

Stopping Potential

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

$$Error = \frac{-A}{B} e^{B(N+0.5))} \tag{5}$$

Plotting Currents

The currents in the system in Cartesian form can be expressed as:

$$J_x = \frac{-\partial\phi}{\partial x} \tag{6}$$

$$J_y = \frac{-\partial\phi}{\partial y} \tag{7}$$

Numerically, this can be expressed as:

$$J_{x,ij} = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2} \tag{8}$$

$$J_{y,ij} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2} \tag{9}$$

On plotting the current density using quiver, we get:

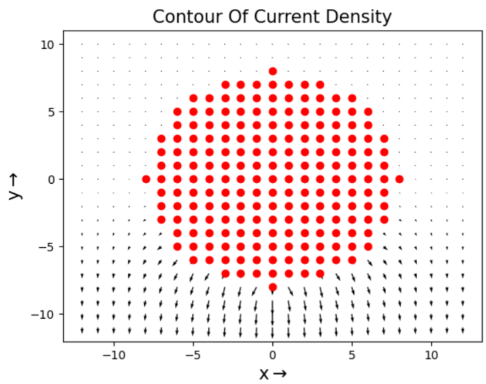

Figure 8: Current Density

We observe that the current is in the direction opposite of gradient of potential.

Conclusion

In this assignment, we have learnt how to measure the potential at any given point with the help of poisson's equations and boundary conditions. We observed how the values of error reduces as the number of iterations increases, We also learnt how to plot 3D plots for values which depend on 2 or more variables. We also learnt how to plot vectors and used it to plot current. We found out which points of the resistor had higher current flowing through it and hence able to discover which area got heated the most.