EE2703: Assignment 5

Gugulothu Naveen

EE20B039

March 6, 2022

1 Introduction

A wire is soldered to the middle of a copper plate and its voltage is held at 1 Volt. One side of the plate is grounded, while the remaining are floating. The plate is 1 cm by 1 cm in size. As a result, current flows. The current at each point can be described by a "current density" $\sim j$.

This current density is related to the local Electric Field by the conductivity:

$$\sim j = \sigma E\sim \tag{1}$$

Electric field is the gradient of the potential:

$$E\sim = -\nabla\varphi \tag{2}$$

The Continuity Equation:

$$\nabla \cdot \vec{j} = -\frac{\partial \rho}{\partial t} \tag{3}$$

Combining these equations we obtain:

$$\nabla^2 \phi = \frac{1}{\rho}\frac{\partial \rho}{\partial t} \tag{4}$$

For DC currents, the right side is zero, and we obtain:

$$\nabla^2\varphi = 0 \tag{5}$$

2 Assignment

2.1 Taking arguments through commandline

```
if (len( sys . argv)==5):
    Nx, Ny,     radius ,          Niter = [ int (x) for x in sys . argv [ 1 : 5 ] ]

#default arguments else :
```

Nx = 25
Ny = 25 radius = 8
Niter = 1500

2.2 Allocating the potential array and initializing it

We start by creating an zero 2-D array of size Nx x Ny. then a list of coordinates lying within the radius is generated and these points are initialized to 1.

```
phi = np. zeros ((Nx,Ny) , dtype = float ) x = np. linspace ( =0.5 ,0.5
,num=Nx, dtype=float ) y = np. linspace ( =0.5 ,0.5 ,num=Ny,
dtype=float ) Y,X = np. meshgrid (y ,x , sparse=False ) phi [np.
where(X**2+Y**2 <(0.35)**2)]=1.0 i i = np. where(X**2+Y**2
<(0.35)**2)
phi[ii]=1.0
```

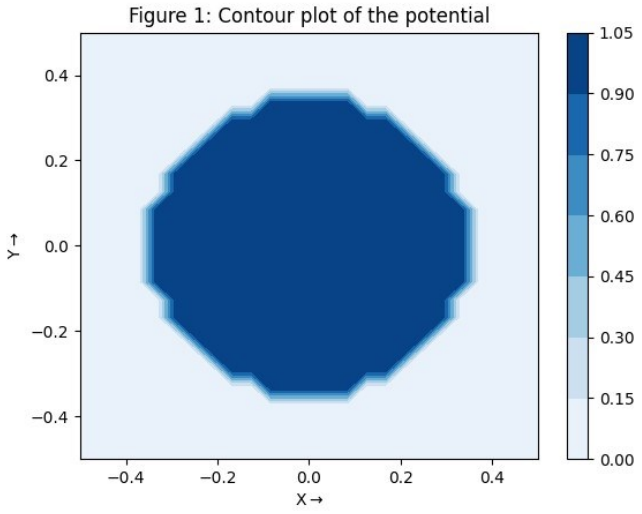

Figure 1: Contour Plot of the Potential

2.3 Performing Iterations

To update the potential we need to first convert it to a difference equation as all of our code is in discrete domain. Thus this is written as:

$$\varphi_{i,j} = 0.25 * (\varphi_{i-1,j} + \varphi_{i+1,j} + \varphi_{i,j+1} + \varphi_{i,j-1})$$

```
err = np. zeros ( Niter , dtype = float ) for k in range(
Niter ): oldphi = phi . copy ()

    #updating      potential phi [1: =1 ,1: =1] = 0.25*( phi [1: =1 ,0: =2] + phi [1: =1 ,2:] + phi [0: =2 ,1: =1] + phi [2: ,1: =1])

    #applying boudary conditions phi [: ,0]=
    phi [ : , 1 ] phi [: ,Nx=1]=phi [: ,Nx=2] phi [0
    ,:]= phi [ 1 , : ] phi [Ny=1,:]=0

    phi [ i i ]=1.0
    err [ k ] = np.max(np. abs(phi=oldphi ))
```

2.4 Plotting the errors

Note that the error falls very slowly and so it is discouraged to use this method for solving the Laplace equation.

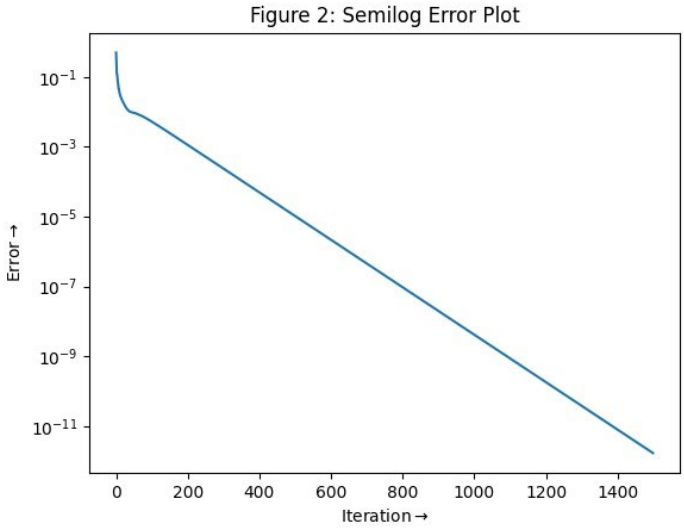

Figure 2: Semilog Error Plot

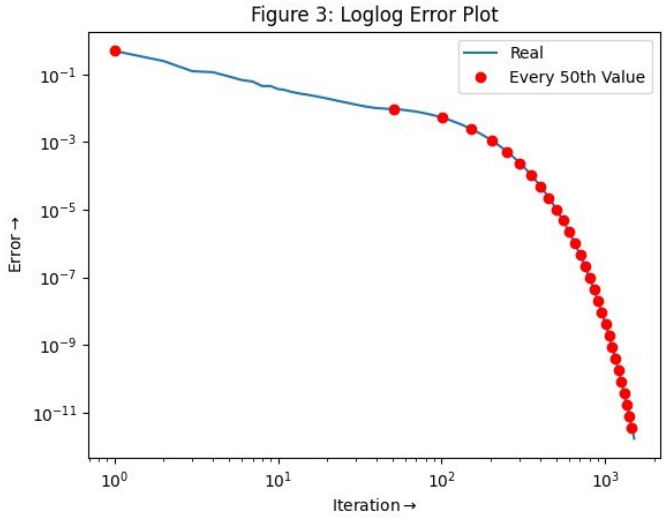

Figure 3: Loglog Error Plot

2.5 Fitting the error

Fit1 is drawn by considering all the iterations and Fit2 is drawn without considering the first 500 iterations. There is very little difference between the two fits.

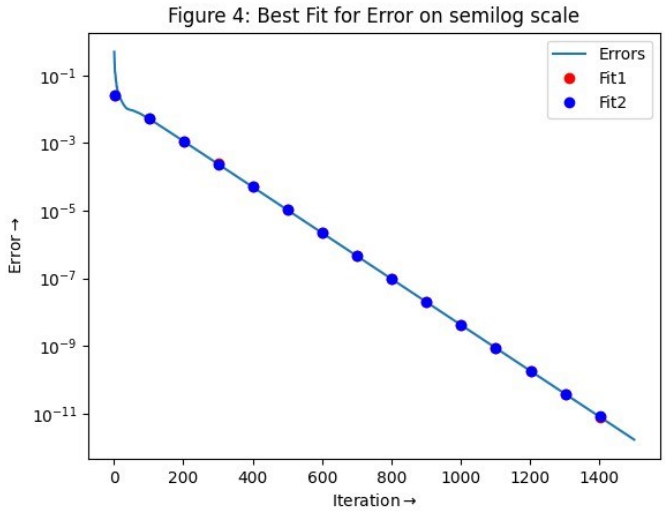

Figure 4: Best Fit for Error in Semilog scale

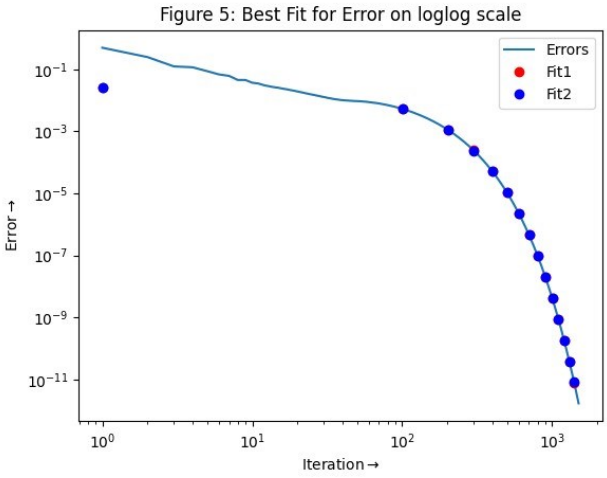

Figure 5: Best Fit for Error in loglog scale

2.6 Plotting Maximum Possible Error

```
def NetError(a,b,Niter):
    return —a/b ∗np.exp(b    ∗( Niter    +0.5))
```

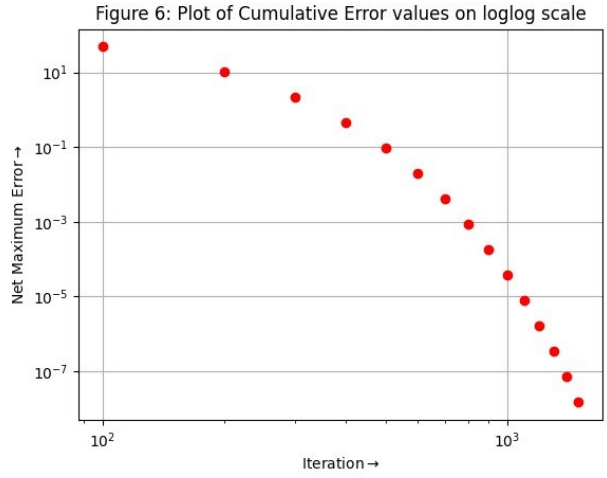

Figure 6: Cumulative error values on a loglog scale

2.7 Surface Plot of Potential

```
fig1=plt.figure(4)
ax=p3.Axes3D(fig1)
surf=ax.plot - surface(Y,X, phi.T, rstride=1,cstride=1,cmap=plt.cm.jet)
plt.show()
```

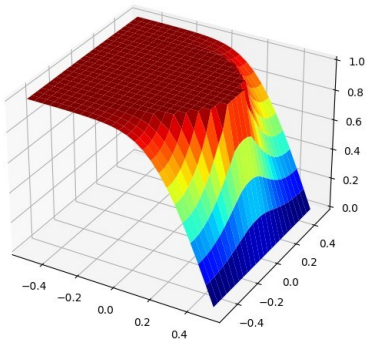

Figure 7: Surface Plot of Potential

2.8 Contour Plot of the Potential

```
x -c,y -c=np.where(X **2+ Y **2 < (0.35) **2)
plt.plot((x -c —Nx/2)/Nx,(y -c —Ny/2)/Ny,'ro')
plt.contourf(Y,X[:: — 1] ,phi )
plt.show()
```

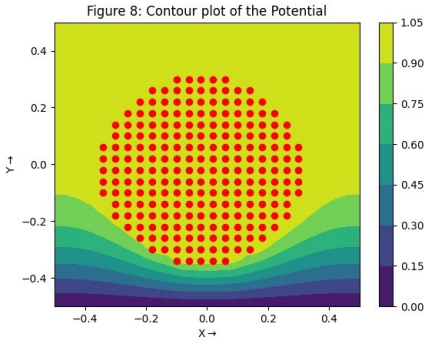

Figure 8: Contour Plot of the Potential

2.9 Vector Plot of Currents

Now we obtain the currents. This requires computing the gradient. The actual value of σ does not matter to the shape of the current profile, so we set it to unity. Our equations are

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

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

```
plt.quiver(Y[1:  — 1 , 1: — 1] , —X[1:  — 1 , 1: — 1] , — Jx[:,::  — 1] , — Jy)
x -c,y  -c=np.where(X  **2+ Y **2 < (0.35) **2)
plt.plot((x  -c —Nx/2)/Nx,(y  -c —Ny/2)/Ny,'ro')
plt.show()
```

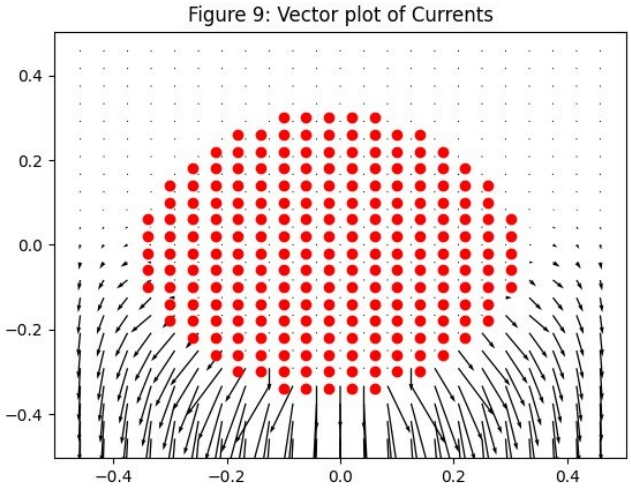

Figure 9: Vector plot of Currents

3 Conclusion

Using finite differentiation approximation we have found a solution to the Laplace's equation for a given system. The error is seen to decay at a highly gradual pace. Thus the chosen method of solving Laplace's equation is inefficient. On analysing the quiver plot of the currents, it was noticed that the current was mostly restricted to the bottom of the wire, and was perpendicular to the surface of the electrode and the conductor.