Assignment 5

Pruthvi Raj R G,EE17B114

March 10, 2019

1 The Resistor Problem

1.1 Introduction

We wish to solve for the currents in a resistor. The currents depend on the shape of the resistor and we also want to know which part of the resistor is likely to get hottest.

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.

1.2 Numerical Solution in 2-Dimensions

Laplaces equation is easily transformed into a difference equation. The equation can be written out in Cartesian coordinates and solved easily using matrix operations. One of the method to obtain the final distribution of the potential is to do averaging seperately for all the points of th plate. But this process can be extremely computationally expensive in python as updating a single value of an array forces python to check for the rest of the indeces too.This problem can be easily countered using the method of vectorization.We difine the potential as a matrix and then we keep on updating the matrix only using vectorized operations.

The code is as follows:

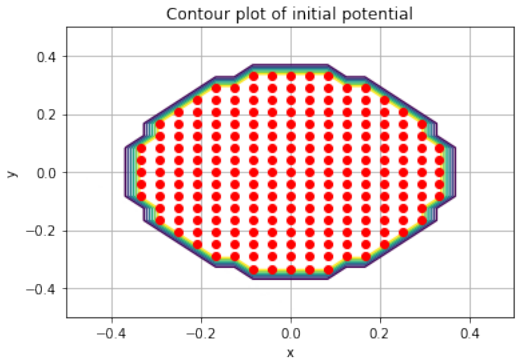


Figure 1: Visualisation of the initial potential on the surface

```
from pylab import *
import mpl_toolkits.mplot3d.axes3d as p3


Nx = 25
Ny = 25
radius = 0.35
Niter = 1500
errors = np.zeros(Niter)


x = np.linspace(-0.5,0.5,25)
y = np.linspace(0.5,-0.5,25)
X,Y = meshgrid(x,y)
phi = np.zeros((Nx,Ny))
ii = where(X*X + Y*Y <= radius*radius)
phi[ii] = 1.0

contour(X,Y,phi)
plot(x[ii[0]],y[ii[1]],'ro')
grid()
title('Contour plot of initial potential')
xlabel('x')
ylabel('y')
show()
```

1.3 Update the potential matrix along with the error in each iteration

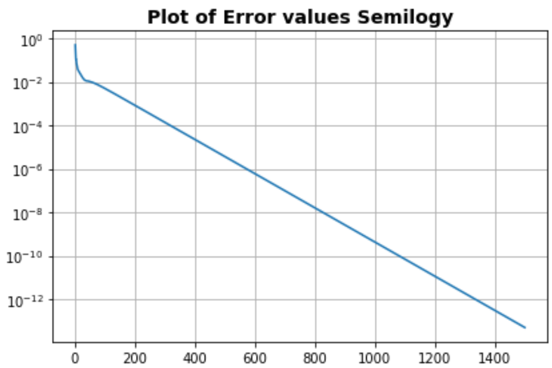
Here were are calculating the error of the current iteration compared to the previous iteration so that we can get an intution about the saturation of the fuction so that we can decide when we can stop iterating the potential.

Code for getting Calculating the error:

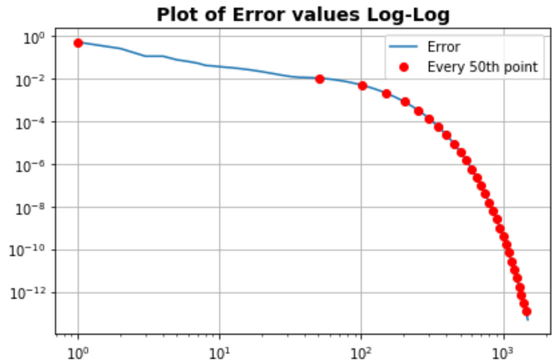
```
newphi = np.zeros((Nx,Ny))
for k in range(Niter):
    oldphi = phi.copy()
    newphi[1:-1,1:-1] = 0.25*(phi[1:-1,0:-2] + phi[1:-1,2:]
     + phi[0:-2,1:-1] + phi[2:,1:-1])

    newphi[1:-1,0] = newphi[1:-1,1]
    newphi[1:-1,Nx-1] = newphi[1:-1,Nx-2]
    newphi[0,1:-1] = newphi[1,1:-1]
    newphi[ii] = 1.0

    errors[k] = max(np.absolute(np.subtract(oldphi.flatten(),newphi.flatten())))
    phi = newphi.copy()
```

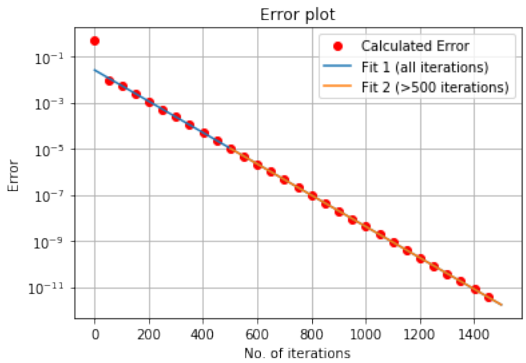
1.4 Plot of error vs iterations

Actual plot of the error with iteration:



The actual error calculated from the above loop is plotted for every 50th iteration.Then, in addition to that, two other lines are also plotted, the best fit line taking all error values obtained and that by taking all error values obtained after 500 iterations.





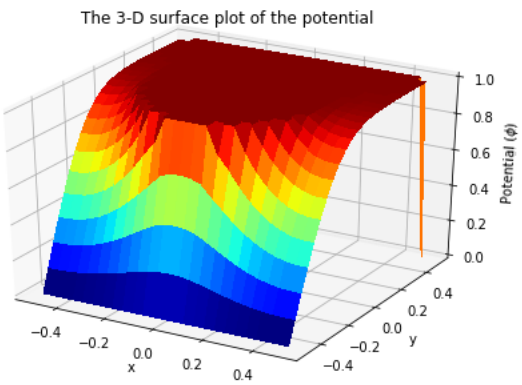
The code for calculating the errors and then plotting them:

```
xError = np.linspace(1,Niter,1500)
yError = np.log(errors)
A=np.zeros((Niter,2))
A[:,0] = 1
A[:,1] = xError
const = lstsq(A,yError)[0]
yError = const[0] + const[1]*xError
yError = np.exp(yError)

semilogy(xError,errors)
show()

loglog(np.arange(1,1501,50),errors[0::50],'ro')
loglog(xError,errors)
show()


xError2 = np.linspace(501,Niter,1000)
```

```
yError2 = np.log(errors[500:])
B=np.zeros((Niter-500,2))
B[:,0] = 1
B[:,1] = xError2
const = lstsq(B,yError2)[0]
yError2 = const[0] + const[1]*xError2
yError2 = np.exp(yError2)

semilogy(np.arange(1,1501,50),errors[0::50],'ro')
plot(xError,yError)
plot(xError2, yError2)
grid()
title('Error plot')
xlabel('No. of iterations')
ylabel('Error')
legend(('Calculated Error','Fit 1 (all iterations)','Fit 2 (>500 iterations)'))
show()
```

1.5 Plotting

We plot 3-D surface plots, contour plots of . In additon to, this the current denisities may be calculated and plotted.We also plot the contour plot to see the spacial variation of the potential.




```
fig1 = figure(4)
ax = p3.Axes3D(fig1)
title('The 3-D surface plot of the potential')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('Potential $(\phi)$')
```

```
surf = ax.plot_surface(X, Y, phi, rstride=1, cstride=1,
 cmap=cm.jet,linewidth=0, antialiased=False)
show()
```

```
contour(x,y,phi)
plot(x[ii[0]],y[ii[1]],'ro')
xlabel('x')
ylabel('y')
title('Contour plot of final potential')
grid()
show()
```

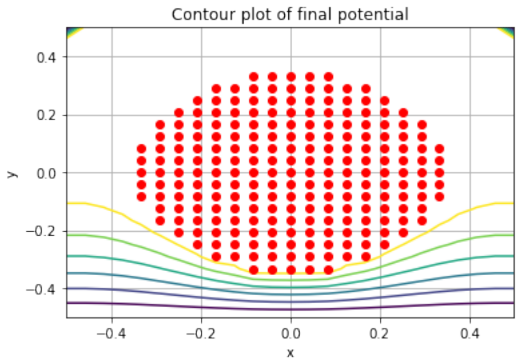


Figure 2: Plot of the contours of potential.

Now we also plot the current values using the calculated gradient of the potential.We use quiver plots to show the currents.

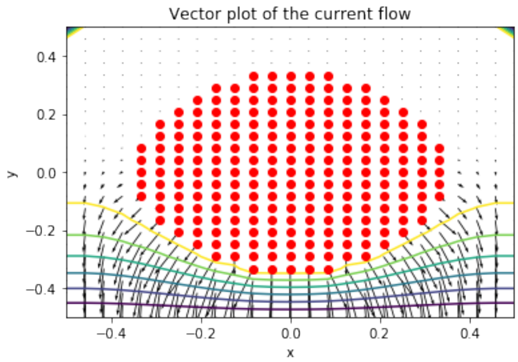


Figure 3: Quiver plot of current values in the region along with the contours of potential.

1.6 Results and Discussion

Here we have used laplace equation to solve for the potential and the currents.
We have employed boundary condition to make solution possible. We have studied the contour plots and 3-D surface plots for the same.
Analysing the quiver plots for current densities affirms that:
current is normal to the surfaceof the conductor,
both the wire and the metal plate