EE2703: Applied Programming Lab
Assignment 5
The Resistor Problem

Devaganthan S S
EE19B018

April 21, 2021

0.1 Abstract

This Assignment aims to

- Solve current in a resistor using python
- Learn to plot Quiver, 3D, and Contour Plot.

0.2 Introduction

We have a metal plate. A circular electrode (of 1V) is placed at the center of the metal Plate. The bottom of the Metal plate is grounded. We find the potential of the plate using Laplace Equation and with the given boundary conditions. We plot the Contour, 3D plot of the Potential Function. We find the Current in the Metal, and plot the quiver plot for the same. We also find the Error in the potential after each Iteration and then plot the *loglog* and *semilog* plots.

0.3 Results and Implementation

0.3.1 Solving and Contour and 3D Plotting for the Potential.

We assume the Metal Plate as a grid having NX rows and NY Columns. We find points that lie within a circular region of given radius using 'where()' . We perform Niter Iterations and in each iteration, we perform the following actions, Copy the Potential Matrix to a New matrix, Update the Phi Matrix, Assert Boundary Conditions, And compute the error. In the end, we have a good approximation of the Potential Function. We then plot the contour plot of the Potential Function. We plot the Contour and 3D plot of the Potential Function. The Below code Performs the above tasks.

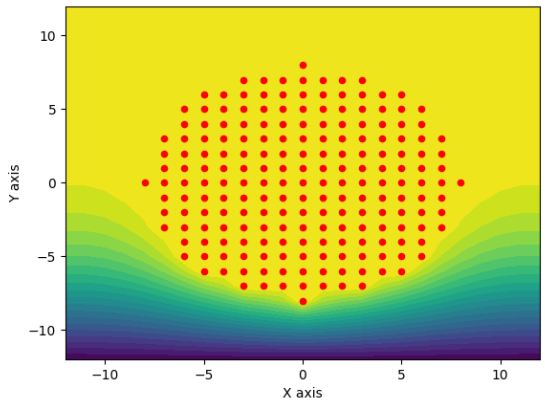
```
# Initializing the Potential Array
phi = zeros([Nx, Ny])
x = arange(-(Nx-1)/2, (Nx+1)/2)
y = arange(-(Ny-1)/2, (Ny+1)/2)
X, Y = meshgrid(x, y)
ii = where(X*X+Y*Y <= radius*radius)
phi[ii] = 1
error = []

# Updating the Potential, Computing the Error , Updating the
    Boundary Conditions
for i in range(Niter):
```

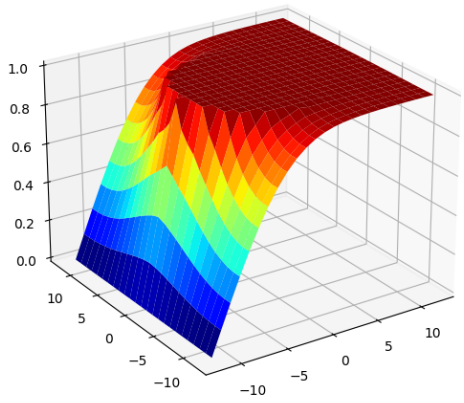
```
    oldphi = phi.copy()
    phi[1:-1, 1:-1] = 0.25 * \
        (phi[1:-1, 0:-2] + phi[1:-1, 2:] + phi[0:-2, 1:-1] + phi
    [2:, 1:-1])
    phi[:, 0] = phi[:, 1]
    phi[:, -1] = phi[:, -2]
    phi[0, :] = phi[1, :]
    phi[ii] = 1
    error = error+[(abs(phi-oldphi)).max()]
phi1 = phi[::-1]

# Contour plot of the Potential Function
contourf(X, Y, phi1, levels=20)
scatter(ii[0] - (Nx - 1) / 2, ii[1] - (Ny - 1) / 2, color='r', s
    =20)
xlabel('X axis')
ylabel('Y axis')
title('Contour Plot for the Potential')
show()

# Surface Plot for the potential
fig1 = figure(4)  # open a new figure
ax = p3.Axes3D(fig1)  # Axes3D is the means to do a surface plot
title('The 3-D surface plot of the potential')
surf = ax.plot_surface(X, Y, phi1.T, rstride=1, cstride=1, cmap=cm
    .jet)
show()
```



(a) Figure 1



(b) Figure 2

0.3.2 Fitting the Error as a Function of no. of Iterations.

We fit the Error into the function,

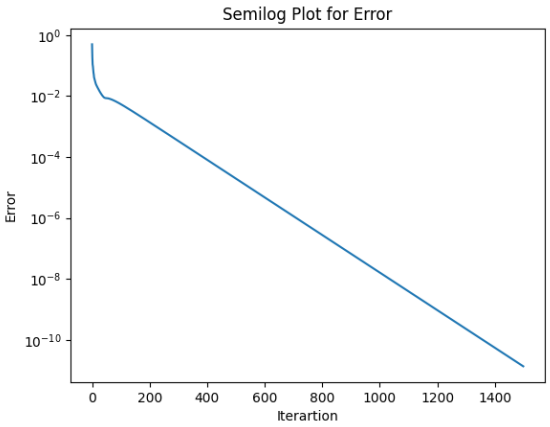
$$log(y) = log(A) + Bx \tag{1}$$

Where y is the error, A and B are parameters to be computed and x is the Iteration Number. We find the parameters with the help of the function 'lstsq()' . With the predicted function, we plot the errors, along with the real error as a function of the number of Iterations. We also plot the semilog and loglog plots of the real Error. The below code accomplishes the above tasks.

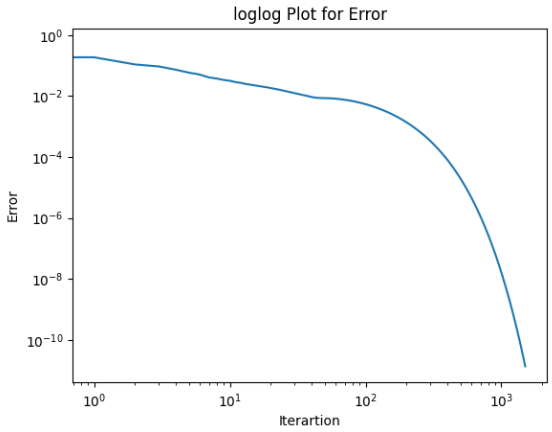
```
# Fitting the Entire Vector
c = ones([Niter], dtype=float)
Amatrix = transpose(array([x, c]))
yMatrix = log(error)
xMatrix = sp.lstsq(Amatrix, transpose(yMatrix))[0]

# Fitting for the portion after 500th Iter
c = ones([len(error[500:])], dtype=float)
x = array(arange(500, Niter))
Amatrix2 = transpose(array([x, c]))
yMatrix2 = log(error[500:])
xMatrix2 = sp.lstsq(Amatrix2, transpose(yMatrix2))[0]

# Plotting the Fits
x = array(arange(Niter))
y1 = exp(dot(Amatrix, xMatrix))
y2 = exp(dot(Amatrix, xMatrix2))
semilogy(x, y1, label='Fit1')
semilogy(x, y2, label='Fit2')
semilogy(x, error, label='Error')  # X is the real error
legend()
show()
```



(a) Figure 5



(b) Figure 6

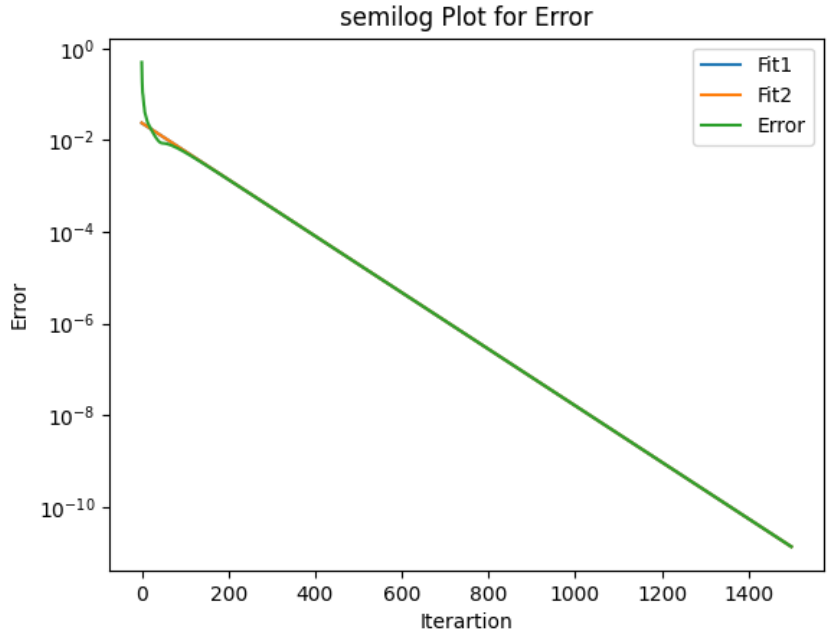


Figure 3

0.3.3 Finding and Vector Plotting the Currents

The Currents are found by Computing the gradient of the Potential Function. Then using the function quiver, the Vector plots of the Current are plotted. The below code accomplishes the above tasks,

```
Jx = zeros([Nx, Ny])
Jy = zeros([Nx, Ny])
Jy[1:-1, :] = -0.5 * (phi[:-2, :] - phi[2:, :])
Jx[:, 1:-1] = 0.5 * (phi[:, :-2] - phi[:, 2:])
Jx = Jx[::-1]
Jy = Jy[::-1]
quiver(X, Y, Jx, Jy, scale=5)
scatter(ii[0] - (Nx - 1) / 2, ii[1] - (Ny - 1) / 2, color='r', s
    =10)
title('Vector Plot for Current')
gca().set_aspect('equal', adjustable='box')
show()
```

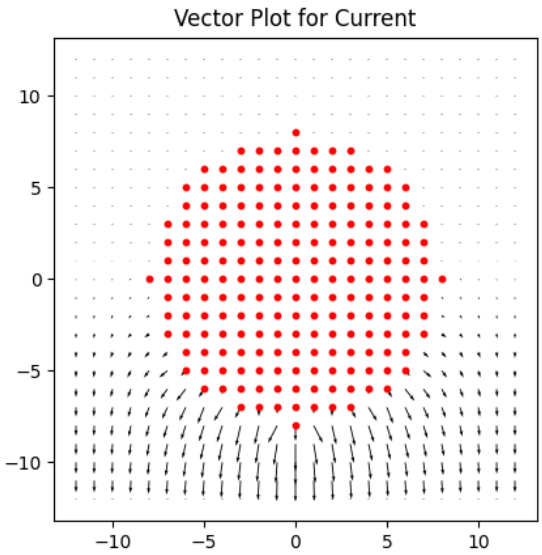


Figure 4

0.4 Conclusion

The Resistor problem is solved using the Laplace equation. Contour and 3D plots for the potential Function are plotted. The max error is computed and semilog and loglog plots are plotted. We find the current densities. The heat produced is maximum where there the current is maximum