

EE2703 : Applied Programming Lab
Assignment 5
The Resistor Problem

Arun Krishna A M S
EE19B001

March 25, 2021

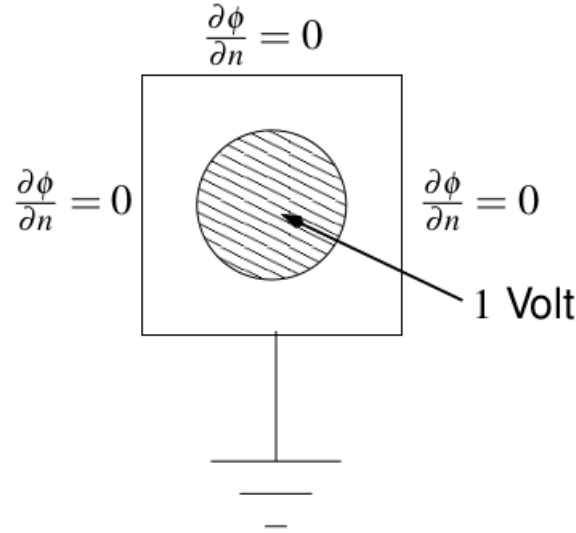
Aim

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in electronic equipment. Practical resistors as discrete components can be composed of various compounds and forms. In this assignment, we mainly focus on:

- Finding the potential on each point on the resistor through Laplace's equation by numerical methods
- Determining the profile of the current flowing through the resistor i.e., dependence of current on the shape of the resistor
- Find which part of the resistor is likely to get hottest, by solving Poisson's equation numerically
- Vectorization of for-loops for faster computing

Introduction

Consider the following apparatus: 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.



Apparatus

As a result of the potential difference, current flows - given by the current density \vec{J} . This current density is related to the local electric field by Ohm's Law:

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

The **continuity equation** states:

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

and we know that, the electric field varies with electrostatic potential as

$$\vec{E} = -\nabla \phi$$

Combining the above three equations we arrive at

$$\nabla \cdot (-\sigma \nabla \phi) = -\frac{\partial \rho}{\partial t} \implies \nabla^2 \phi = \frac{1}{\sigma} \frac{\partial \rho}{\partial t}$$

Where σ is the material's constant conductivity. For DC currents, the right hand side is zero, and we obtain the **laplace equation**:

$$\nabla^2 \phi = 0$$

This equation is a special case of Poisson's equation:

$$\nabla^2 \phi = f$$

Laplace's equation in two-dimensions can be written in Cartesian coordinates as

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

When written in discrete form

$$\begin{aligned} \frac{\partial \phi}{\partial x} \Big|_{(x_i, y_j)} &= \frac{\phi(x_{i+1/2}, y_j) - \phi(x_{i-1/2}, y_j)}{\Delta x} \\ \frac{\partial^2 \phi}{\partial x^2} \Big|_{(x_i, y_j)} &= \frac{\phi(x_{i+1}, y_j) - 2\phi(x_i, y_j) + \phi(x_{i-1}, y_j)}{\Delta x^2} \end{aligned}$$

Similarly

$$\frac{\partial^2 \phi}{\partial y^2} \Big|_{(x_i, y_j)} = \frac{\phi(x_i, y_{j+1}) - 2\phi(x_i, y_j) + \phi(x_i, y_{j-1}))}{\Delta y^2}$$

Combining both equations, we arrive at

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

Thus, if the solution holds, the potential at any point should be the average of its neighbours. One can also argue that the function ϕ is a harmonic function, thus the result holds.

Initialization

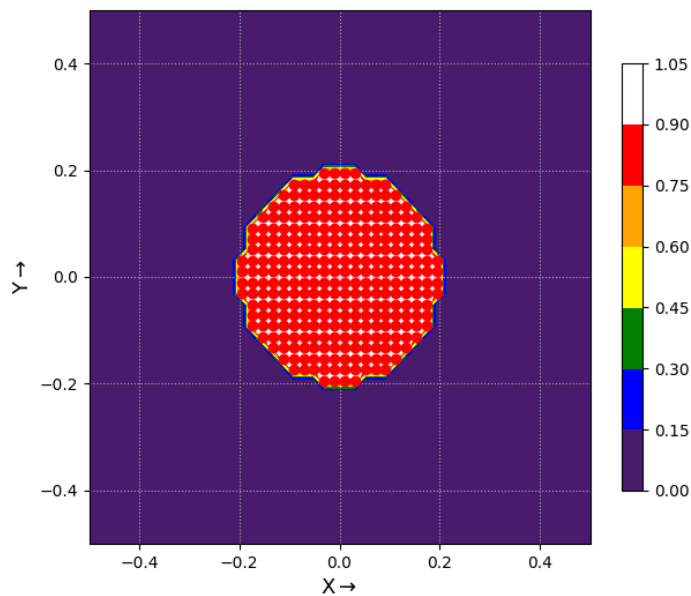
We initialize the coordinate system as follows:

```
Nx = 50
Ny = 50
radius = 10
Niter = 1500
x = np.linspace(0.5,-0.5,Nx,dtype = float);
y = np.linspace(0.5,-0.5,Ny,dtype = float);
Y,X = np.meshgrid(y,x);
```

`Nx` and `Ny` determine the resolution of the coordinate system, and `radius` determines the radius of the soldered wire. `X` and `Y` are used as a coordinate map for further code execution.

The initial values of the system is determined by the conditions specified earlier. The area of contact of the wire to the middle of copper plate is at 1 Volt. We initialize electric potential `phi` as 0 for all other points

```
radius = (1/Nx + 1/Ny)*(1/2)*radius;
ii = np.where(np.multiply(X,X) + np.multiply(Y,Y) < radius**2);
phi = np.zeros([Ny,Nx], dtype = float);
phi[ii] = 1.0;
```



Electric Potential - initial values

Boundary Condition and Laplace equation

At boundaries where the electrode is present, the value of electrostatic potential is same as electrode potential itself. At the bottom boundary where the plate is connected to ground, the potential is always zero. At other boundaries the gradient of ϕ should be tangential i.e., ϕ should not vary in the normal direction. For example, at the left boundary, at $\frac{\partial \phi}{\partial n} = 0$. This is achieved by

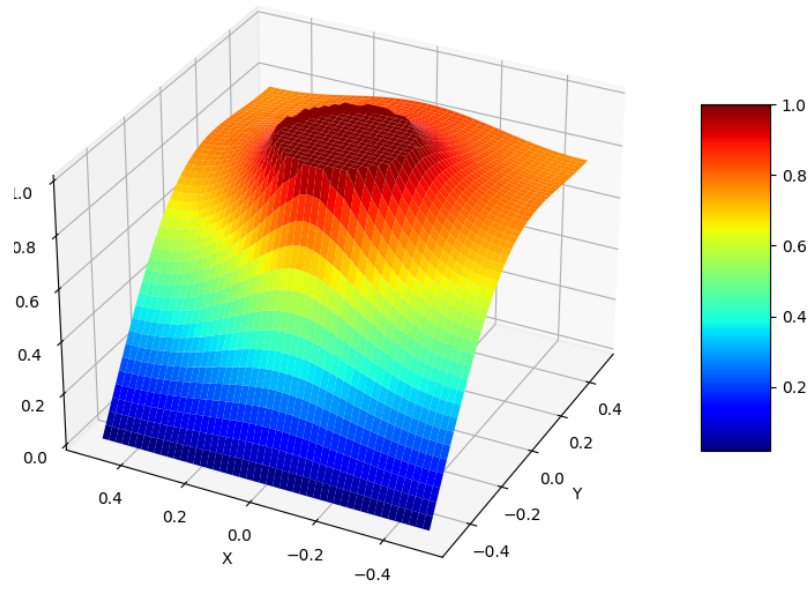
```
phi[:, 0] = phi[:, 1];           #Left margin
phi[:, Nx-1] = phi[:, Nx-2];     #Right margin
phi[0, :] = phi[1, :];           #Top margin
phi[Ny-1, :] = 0;                #Bottom margin
phi[ii] = 1.0;                   #Potential = electrode potential = 1V
```

As mentioned earlier, at each point, replace the potential by the average of its neighbours. Keep iterating till the solution converges.

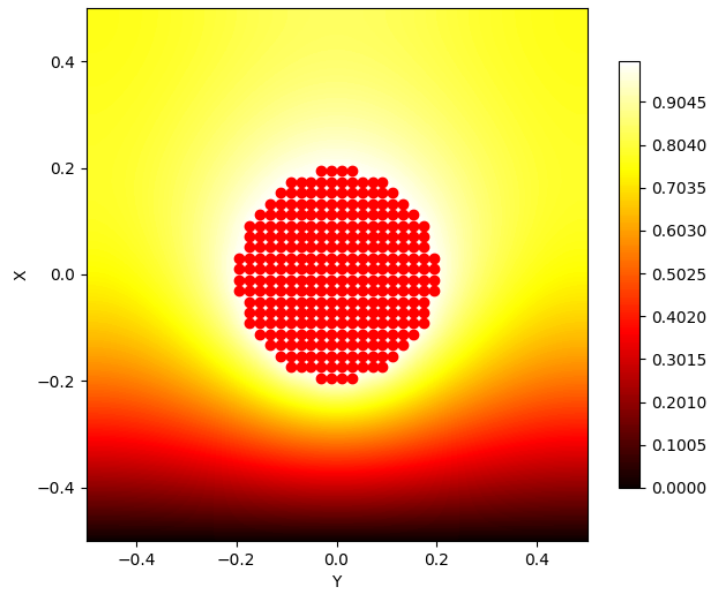
```
oldphi = phi.copy()
phi[1:-1,1:-1] =
0.25*(oldphi[1:-1,0:-2] + oldphi[1:-1,2:] + oldphi[0:-2,1:-1] + oldphi[2:,1:-1]);
```

In each iteration, the ϕ value at each point is copied, and the potential is replaced by the average of its neighbours. The boundary condition is also enforced in each iteration. The error for each iteration is also calculated.

```
for k in range(Niter)
    save copy of phi
    update phi array
    assert boundaries
    errors[k]=(abs(phi-oldphi)).max();
#end
```



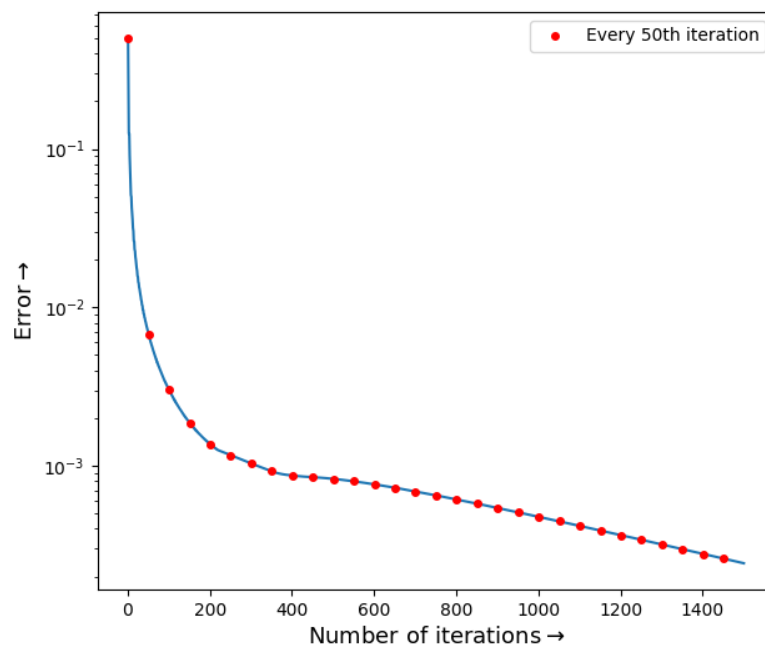
3-D contour plot: Electrostatic Potential



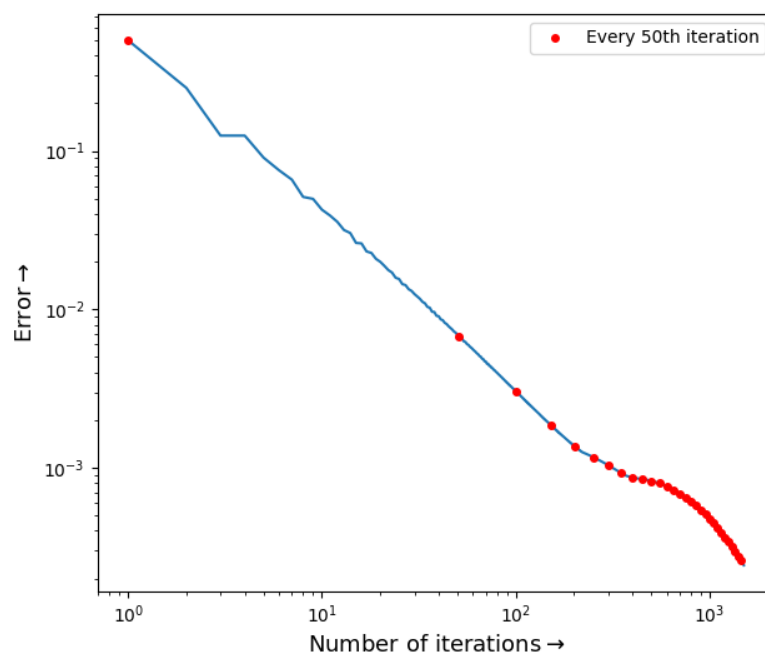
2-D contour plot: Electrostatic Potential

The marked red dots is the area of contact between electrode and the copper plate.

Error Plot



Error vs Number of Iterations: semilogY plot



Error vs Number of Iterations: loglog plot

We can find that a log-log plot gives a reasonably straight line upto about 500 iterations, but beyond that, it falls into the exponential regime. While plotting semilog plot, we can find that it is an exponential decay only for larger iteration numbers.

Since it is exponential, we extract the exponent and fit into a semilog plot of the form:

$$y = Ae^{bx}$$

$$\log y = \log A + b \log x$$

To estimate the values of A and b we use **least square method** such that the error in Y is minimum.

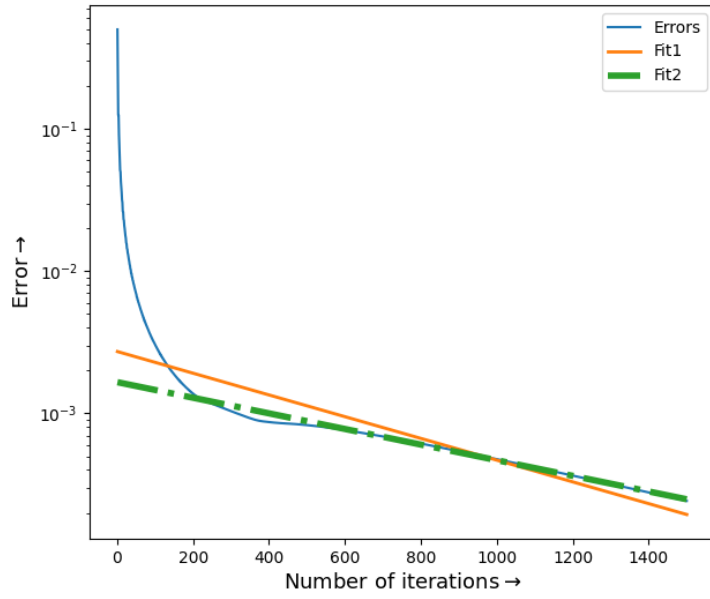
$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \dots & \dots \\ 1 & x_n \end{pmatrix} \cdot \begin{pmatrix} \log(A) \\ B \end{pmatrix} = \begin{pmatrix} \log(y_1) \\ \log(y_2) \\ \dots \\ \log(y_n) \end{pmatrix}$$

Since, we find that it is an exponential decay only for larger iteration numbers, we try to fit the curve for both the entire vector and for iterations greater than 500.

```
def Fitting_Function(X,Y):
    logY = np.transpose(log(Y));
    X = np.c_[np.ones([np.size(X),1]),X];
    return scipy.linalg.lstsq(X,logY)[0];

A1, B1 = Fitting_Function(iterations, errors);
A1 = exp(A1);
A2, B2 = Fitting_Function(iterations[500:], errors[500:]);
A2 = exp(A2);
print(A1, B1, A2, B2);
```

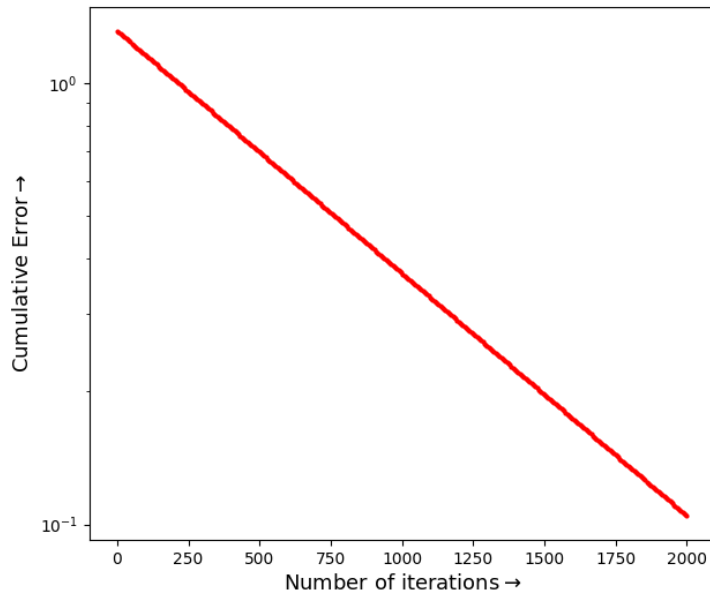
The determined coefficients is then used to plot the error. From the graph we can see that the values of A and b estimated by considering iterations greater than 500 is accurate compared to fitting the curve using the entire vector.



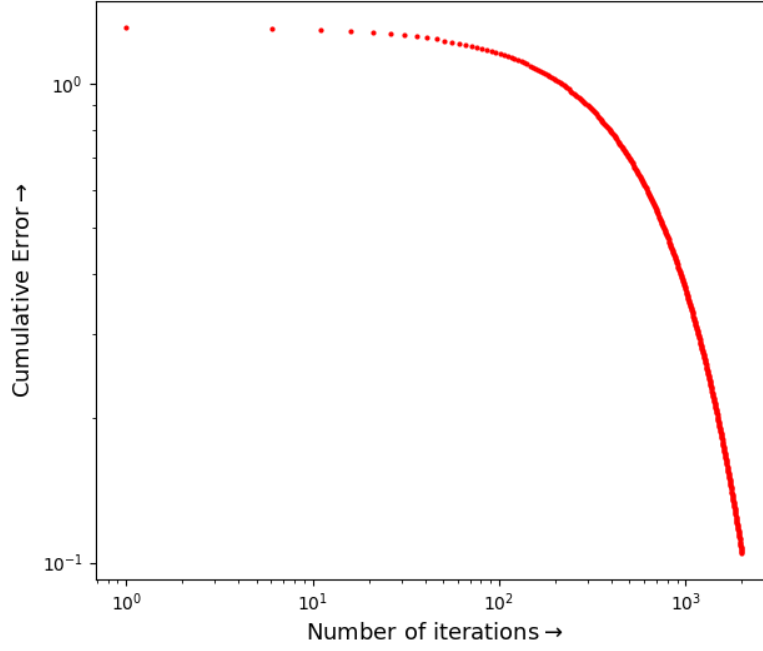
Fitting errors: semilog plot

The cumulative error is given by:

$$\text{Cumulative error} = \sum_{k=N+1}^{\infty} \text{Error} < \int_{N+1}^{\infty} A e^{bx} = \frac{A}{B} \exp(B(N + 0.5))$$



Cumulative Error: semilogy plot



Cumulative Error: loglog plot

Current Distribution

To obtain the current profile, we compute the gradient. Since the major aim of this assignment is to find the shape of the current profile, we set the conductivity σ to unity. The equations are then:

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

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

This numerically translates to

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

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

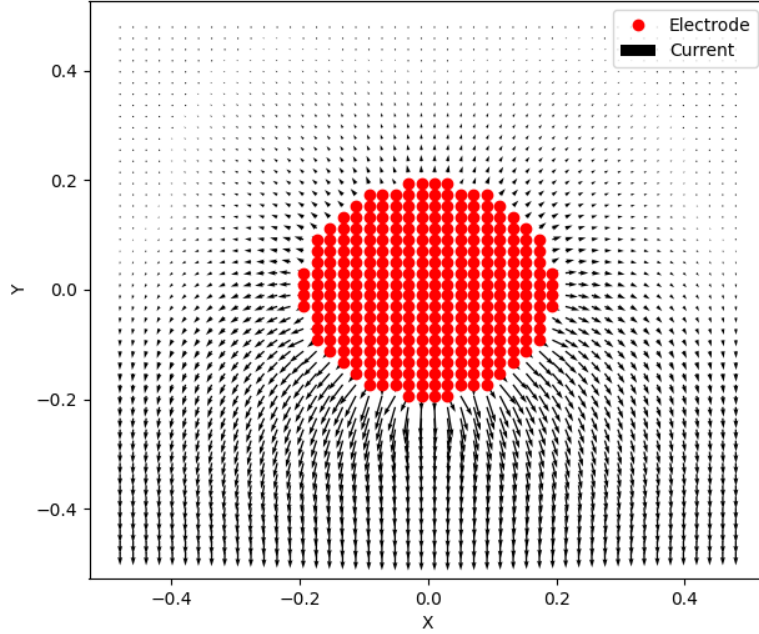
The arrays J_x and J_y are created. The current density is plotted using `quiver` function and the electrode is marked via red dots.

```
Jx = 1/2*(phi[1:-1, :-2]-phi[1:-1, 2:])
Jy = 1/2*(phi[: -2, 1:-1]-phi[2:, 1:-1])
```

.

.

```
quiver(Y[1:-1,1:-1],X[1:-1,1:-1],-Jx,-Jy, label = "Current")
```



Vector Plot: Current Distribution Profile

We can observe that the current flow is from higher potential(i.e., electrode 1V) to lower potential (ground). There is no current within the area of contact with electrode potential, since the electrostatic potential is constant i.e., 1V. Hardly any current flows to the top part of the copper plate.

Heat Map of the conductor

Most of the current is in the bottom region, making the region relatively hot.

$$\nabla \cdot (\kappa \nabla T) = -\frac{1}{\sigma} |\vec{J}|^2$$

where the right side represents heat generated from ohmic loss. The above equation is a **Poisson's equation**. The boundary condition is $T = 300$ at the wire and the ground while $\frac{\partial T}{\partial n} = 0$ at the other three edges.

```
temperature = 300*np.ones([Ny,Nx], dtype = float);
```

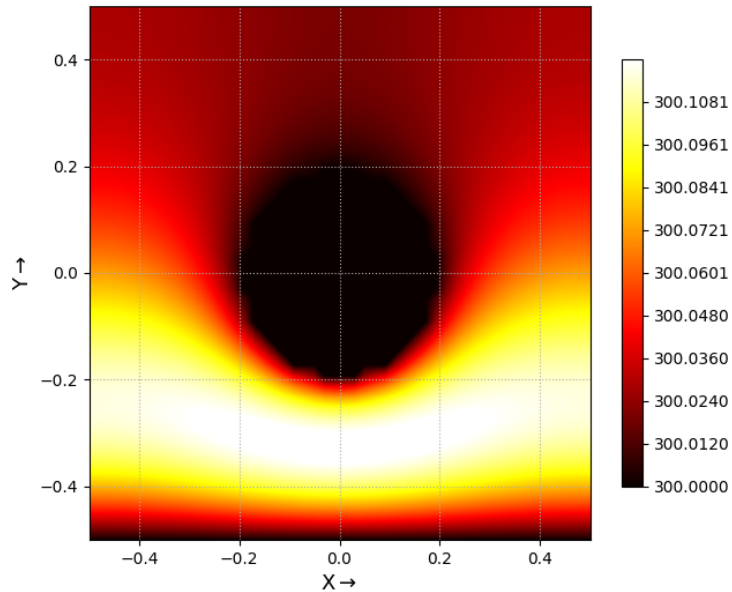
```

for k in range(Niter):
    oldtemperature = temperature.copy()
    var = (Jx**2 + Jy**2);

    temperature[1:-1,1:-1]=0.25*(oldtemperature[1:-1,0:-2]+ oldtemperature[1:-1,2:]+
    oldtemperature[0:-2,1:-1] + oldtemperature[2:,1:-1] + var);

    temperature[:, 0] = temperature[:, 1];           #Left margin
    temperature[:, Nx-1] = temperature[:, Nx-2];     #Right margin
    temperature[0, :] = temperature[1, :];           #Top margin
    temperature[Ny-1, :] = 300;                       #Bottom margin
    temperature[ii] = 300;

```



Heat Signature: Temperature: Contour plot

Conclusion

Solution for the Laplace's equation for the given system is found numerically. As the decay of error is very slow this method of solving Laplace's equation is highly inefficient. Errors were plotted and we found while plotting semilog plot that it is an exponential decay for larger iteration numbers. The vector plot of the currents are plotted, and we found that the current flow is towards the bottom part of the plane - resulting in increased heating in the region.