Solving Laplace Equation

Mohammed Muqeeth
EE16B026
Electrical Engineering
IIT Madras

March 6, 2018

Abstract

This report presents a method for solving laplace equation for 2 dimensions.

1 Introduction

The current in the resistor depend its shape. This report models a resistor by a wire soldered to the middle of copper plate.

1.1 Resistor Modelling

The copper plate is 1 cm by 1 cm in size.The wire is held at 1 volt. The wire is of diameter 0.7cm. One side of the plate is grounded, while the remaining are floating.

1.2 Derive Laplace equation for given problem

1. From conductivity equation:
$$\overrightarrow{j} = \sigma \overrightarrow{E}$$
2. electrical field is given by :
$$\overrightarrow{E} = -\nabla\phi$$
3. charge continuity equation :
$$\nabla.\overrightarrow{j} = -\frac{\partial\rho}{\partial t}$$
4. Above all equations result in (assuming σ is constant):
$$\nabla^2\phi = \frac{1}{\sigma}\frac{\partial\rho}{\partial t}$$
5. for DC currents $\frac{\partial\rho}{\partial t} = 0$ this reduces above equation to $\nabla^2\phi = 0$

1.3 Solve laplace equation

1. For 2 dimension case the above equation in cartesian coordinates becomes $\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2} = 0$
2. From central difference technique,
$$\frac{\partial\phi}{\partial x}\,|_{(x_i,y_j)}= \frac{\phi(x_{i+1/2},y_j)-\phi(x_{i-1/2},y_j)}{\triangle x} \tag{1}$$
$$\frac{\partial^2\phi}{\partial x^2}\,|_{(x_i,y_j)}= \frac{\phi(x_{i+1},y_j)-2\phi(x_i,y_j)+\phi(x_{i-1},y_j)}{\triangle x^2} \tag{2}$$

3. From above two equations $\phi_{i,j}$can be calculated as

$$\phi_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}}{4} \tag{3}$$

4. From equation 3 it is clear that ϕ at any point is average of its neighbours.

2 Method

2.1 Import packages required

```
#import libraries
from pylab import *
from numpy import *
import mpl_toolkits.mplot3d.axes3d as p3
import matplotlib.pyplot as plt
```

2.2 Set the model

1. The parameters can also be taken from user by using sys.argv[i] where the parameters are passed from terminal but here the values taken are

```
#parameters
Nx=25 # size along x
Ny=25 # size along y
radius=8# radius of central lead
Niter=1500# number of iterations to perform
```

2. The potential matrix is made with zeros initally of size Ny rows and Nx columns

```
#allocate potential array
phi = zeros((Ny,Nx))
```

3. Define x and y coordinates for the plate

```
#define x and y coordinates
x = linspace(-0.5,0.5,Nx)
y = linspace(-0.5,0.5,Ny)
```

4. Next get coordinates where wire which is of 1volt is present and set the values in corresponding matrix to be 1

```
#get coordinates where voltage is 1
Y,X = meshgrid(y,x)
ii = where(X*X +Y*Y <= 0.35*0.35)
#set their value to 1
phi[ii] = 1.0
```

5. Plot the locus of points where voltage =1, The corresponding plot is shown in figure 1.

```
#plot the contour
plt.scatter(Y,X,(phi==1),color='r')
plt.title('points where voltage=1')
plt.grid()
plt.show()
```

2.3 Update Potential using equation 3

1. Before updating phi take copy of phi into phiold. The neighbour values can be obtained by left shift, right shift, down shift, up shift of phi matrix.phi is then calculated by sum of all those shifted matrices multiplied by 0.25.

2. Since one side of plate is grounded it remains at 0. The other three boundaries of plate take their neighbour values ie above surface takes values just one below it. right surface takes values of the one just right to it.

3. The potential of wire which is of diameter 0.7cm remain at 1volt.

4. For every iteration the maximum of absolute of phi and phiold are calculated and stored in errors array.

```
#update potential
count = 0
errors = zeros(Niter)
while(count < Niter):
    phiold = phi.copy()
    phi[1:-1,1:-1] = 0.25*(phiold[1:-1,0:-2]+phiold[1:-1,2:]+phiold[0:-2,1:-1]+phi
    phi[1:-1,0] = phi[1:-1,1]
    phi[1:-1,-1] = phi[1:-1,-2]
    phi[-1,:] = phi [-2,:]
    phi[ii] = 1.0
    errors[count] = (abs(phi-phiold)).max()
    count+=1
```

2.4 Fit the errors to a linear model

1. The model which is approximated for error with iteration is

$$y = A\exp(Bx)$$

$$\log(y) = \log(A) + Bx$$

2. The fit1 corresponds to taking iterations from 1 to 1500, fit2 corresponds to iterations from 500 to 1500.

3. The matrix equations for fit1(equation 4) and fit2(equation 5)are

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_{1500} \end{pmatrix} \begin{pmatrix} \log A \\ B \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_{1500} \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} 1 & x_{500} \\ 1 & x_{501} \\ \dots & \dots \\ 1 & x_{1500} \end{pmatrix} \begin{pmatrix} \log A \\ B \end{pmatrix} = \begin{pmatrix} y_{500} \\ y_{501} \\ \dots \\ y_{1500} \end{pmatrix} \tag{5}$$

4. Using lstsq get A,B .The plots of fit1 and fit2 are in figure 2

```
#iteration vector
x0 = linspace(1,Niter,Niter)
#fit1
P = zeros((Niter,2))
```

```
P[:,0] = 1
P[:,1] = x0
q = log(errors)
A,B = lstsq(P,q)[0]
A = exp(A)
y1 = A*exp(B*x0)
#fit2
P = zeros((Niter-500,2))
P[:,0] = 1
P[:,1] = x0[500:]
q = log(errors[500:])
A,B = lstsq(P,q)[0]
A = exp(A)
y2 = A*exp(B*x0)
#plot both models
plt.xlim(-50,1550)
plt.semilogy(x0[::50],errors[::50],'bo')
plt.semilogy(x0[::50],y1[::50],'r^')
plt.semilogy(x0[500::50],y2[500::50],'g--')
plt.legend(['errors','fit A','fit B which is after 500th iteration'],loc="best")
plt.grid()
plt.xlabel('number of iterations')
plt.ylabel('error')
plt.show()
```

2.5 Surface plot of potential

1. The surface plot of potential is in figure 3.

```
#surface plot of voltage
fig1 = figure(1)
ax = p3.Axes3D(fig1)
title('the 3-D surface plot of the potential')
surf=ax.plot_surface(X,Y,phi.T,rstride=1,cstride=1,cmap=cm.jet)
xlabel('X')
ylabel('Y')
ax.set_zlabel('potential')
show()
```

2.6 Contour plot of potential

1. Both contour and contourf does the job of contour plot of potential.

2. The contour plot of potential is in figure 4.

```
#contour plot
contour(phi)
title('contour plot of potential')
show()
```

2.7 Modelling currents

1. The current densities $J_{x,ij}, J_{y,ij}$ are given as

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

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

2. The quiver functions plots current density .The corresponding plot is in figure 5

```
#quiver plot
Jx = zeros((Ny,Nx))
Jy = zeros((Ny,Nx))
Jy[1:-1,1:-1] = 0.5*(phi[0:-2,1:-1]-phi[2:,1:-1])
Jx[1:-1,1:-1] = 0.5*(phi[1:-1,0:-2]-phi[1:-1,2:])
plt.plot(ii[1],ii[0],'ro')
plt.quiver(Jx[:,:],Jy[:,:])
plt.title('The vector plot of current flow')
plt.show()
```

3 Plots and Results

Figure 1: Points where voltage=1

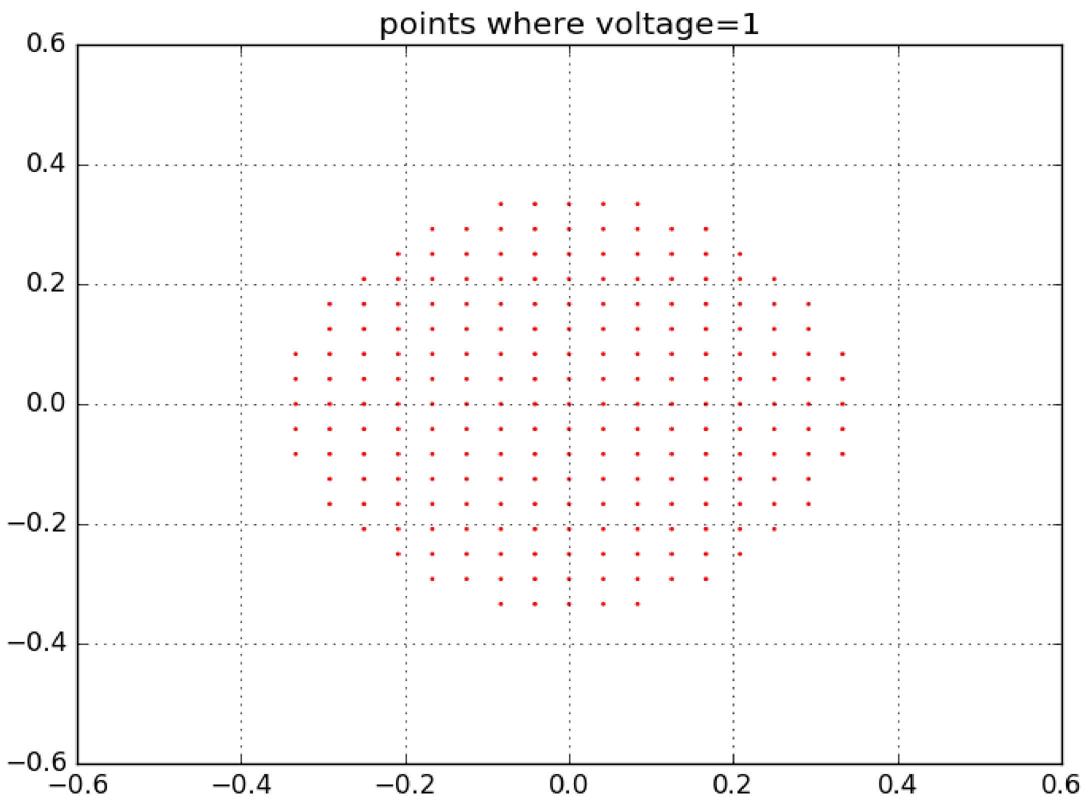


Figure 2: 3-D surface plot of potential

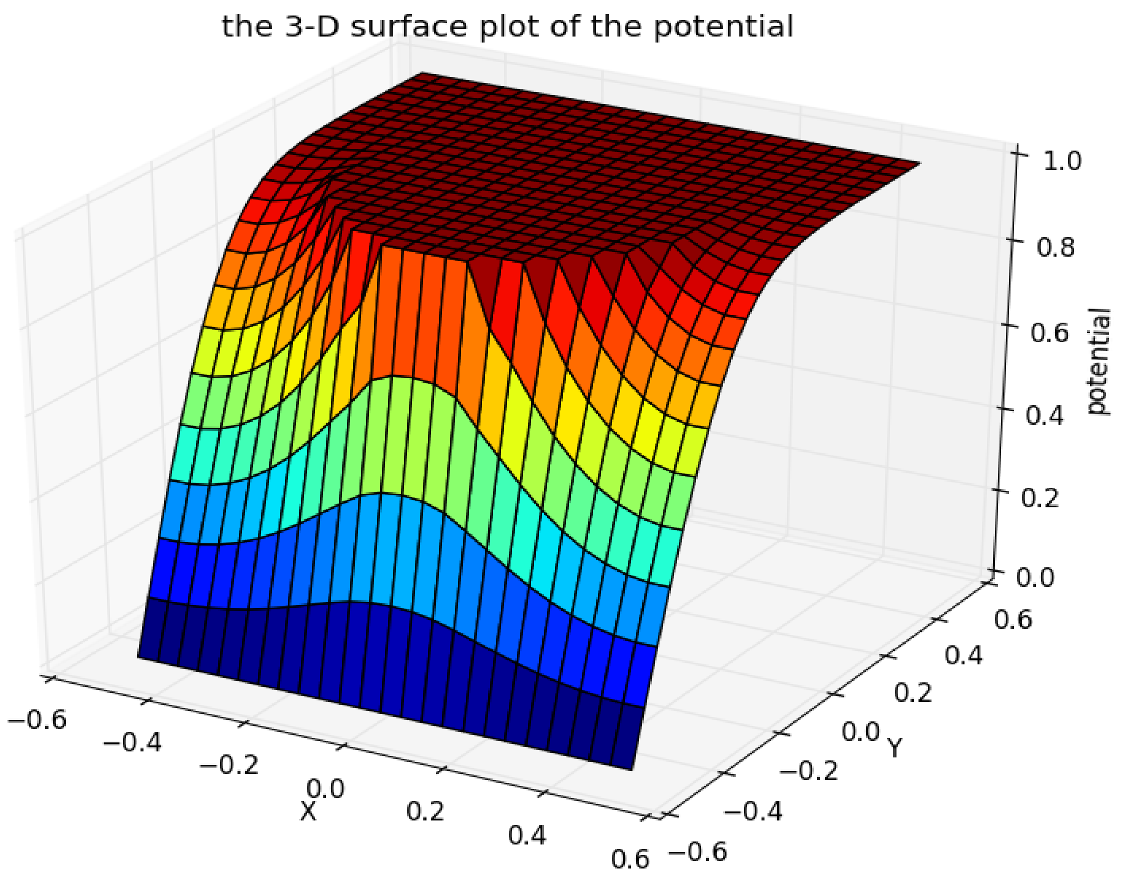


Figure 3: errors vs iterations

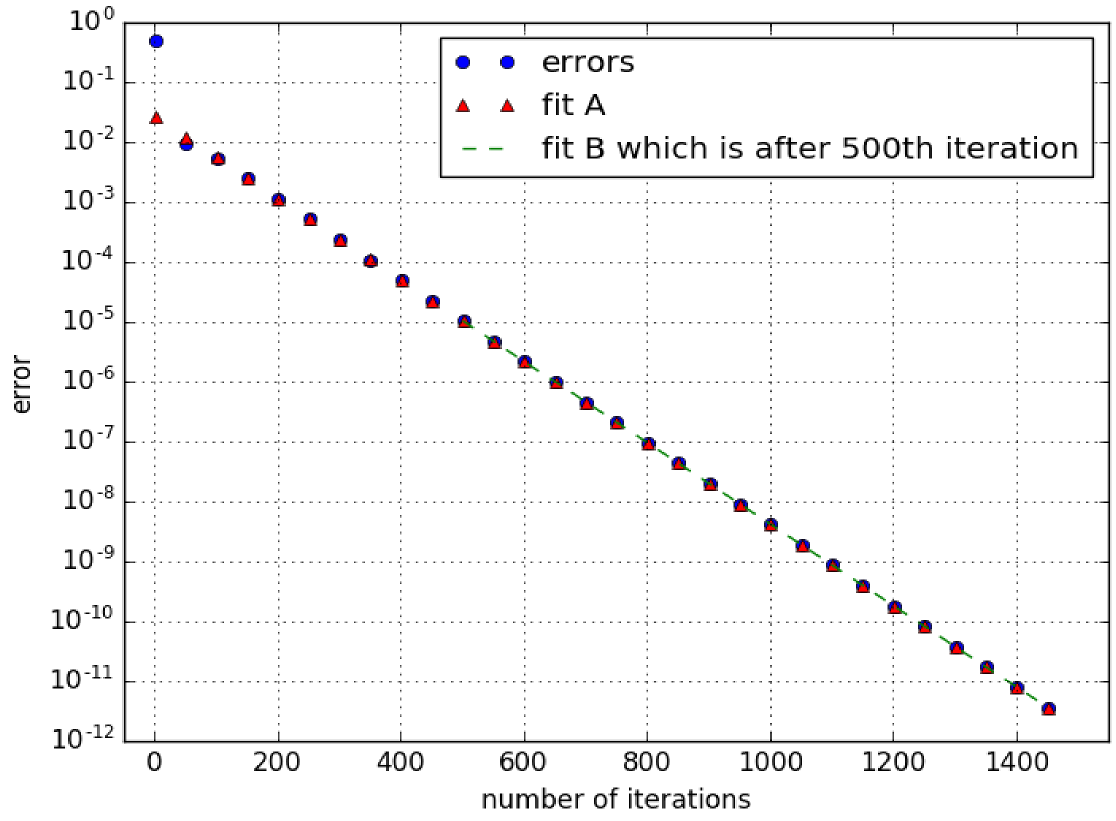


Figure 4: Contour plot of potential

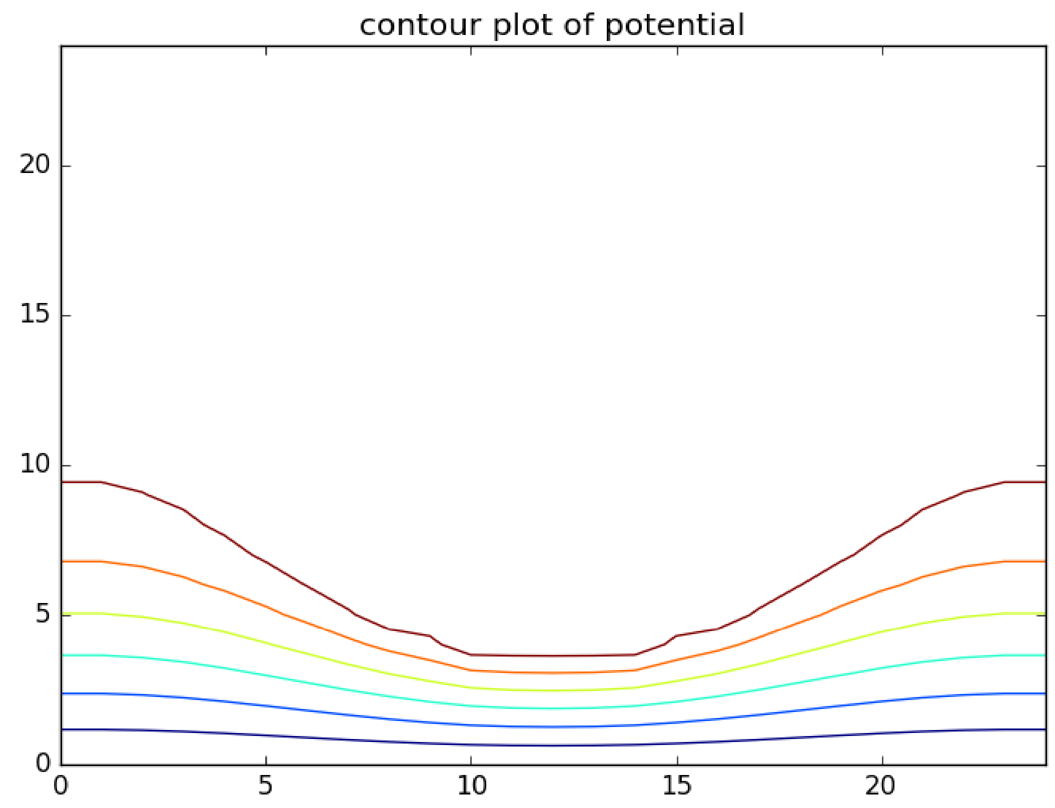
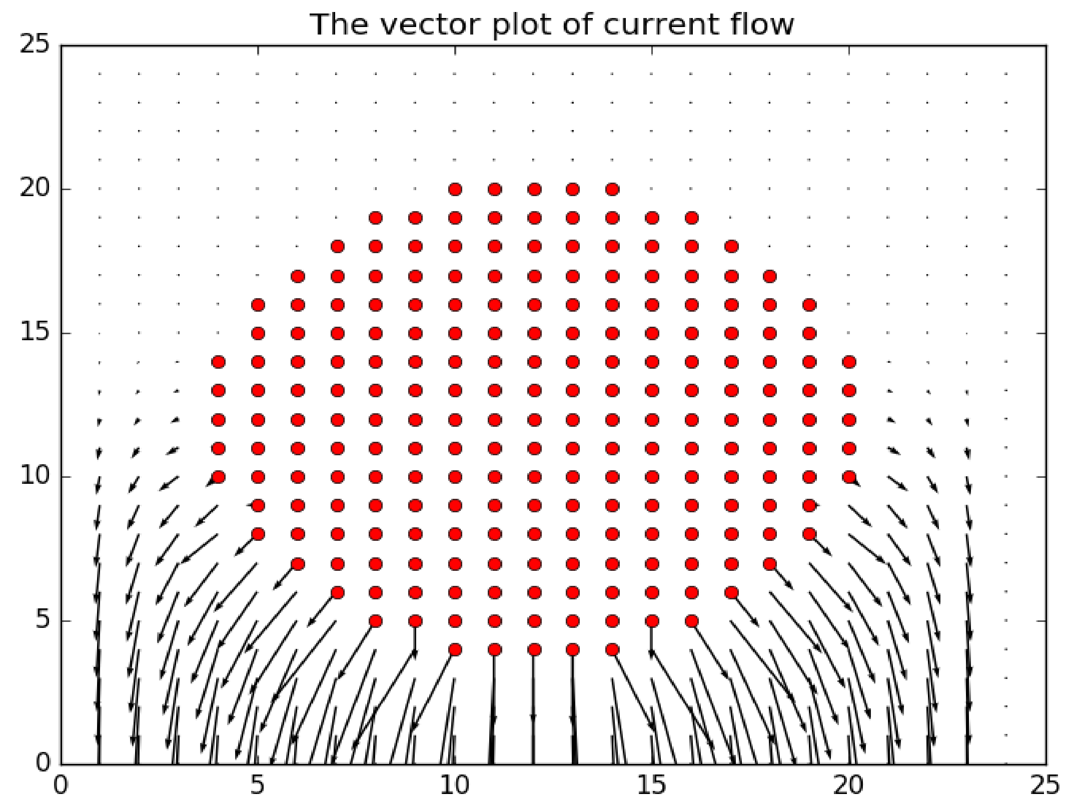


Figure 5: Vector plot of current flow



4 Code

```
#import libraries
from pylab import *
from numpy import *
import mpl_toolkits.mplot3d.axes3d as p3
import matplotlib.pyplot as plt
#parameters
Nx=25# size along x
Ny=25# size along y
radius=8# radius of central lead
Niter=1500 # number of iterations to perform
#allocate potential array
phi = zeros((Ny,Nx))
#define x and y coordinates
x = linspace(-0.5,0.5,Nx)
y = linspace(-0.5,0.5,Ny)
#get coordinates where voltage is 1
Y,X = meshgrid(y,x)
ii = where(X*X +Y*Y <= 0.35*0.35)
#set their value to 1
phi[ii] = 1.0
#plot the contour
plt.scatter(Y,X,(phi==1),color='r')
plt.title('points where voltage=1')
plt.grid()
plt.show()
#update potential
count = 0
errors = zeros(Niter)
while(count < Niter):
    phiold = phi.copy()
    phi[1:-1,1:-1] = 0.25*(phiold[1:-1,0:-2]+phiold[1:-1,2:]+phiold[0:-2,1:-1]+phiold[
    phi[1:-1,0] = phi[1:-1,1]
    phi[1:-1,-1] = phi[1:-1,-2]
    phi[-1,:] = phi [-2,:]
    phi[ii] = 1.0
    errors[count] = (abs(phi-phiold)).max()
    count+=1
#surface plot of voltage
fig1 = figure(1)
ax = p3.Axes3D(fig1)
title('the 3-D surface plot of the potential')
surf=ax.plot_surface(X,Y,phi.T,rstride=1,cstride=1,cmap=cm.jet)
xlabel('X')
ylabel('Y')
ax.set_zlabel('potential')
show()
#iteration vector
x0 = linspace(1,Niter,Niter)
#fit1
P = zeros((Niter,2))
P[:,0] = 1
```

```
P[:,1] = x0
q = log(errors)
A,B = lstsq(P,q)[0]
A = exp(A)
y1 = A*exp(B*x0)
#fit2
P = zeros((Niter-500,2))
P[:,0] = 1
P[:,1] = x0[500:]
q = log(errors[500:])
A,B = lstsq(P,q)[0]
A = exp(A)
y2 = A*exp(B*x0)
#plot both models
plt.xlim(-50,1550)
plt.semilogy(x0[::50],errors[::50],'bo')
plt.semilogy(x0[::50],y1[::50],'r^')
plt.semilogy(x0[500::50],y2[500::50],'g--')
plt.legend(['errors','fit A','fit B which is after 500th iteration'],loc="best")
plt.grid()
plt.xlabel('number of iterations')
plt.ylabel('error')
plt.show()
#contour plot
contour(phi)
title('contour plot of potential')
show()
#quiver plot
Jx = zeros((Ny,Nx))
Jy = zeros((Ny,Nx))
Jy[1:-1,1:-1] = 0.5*(phi[0:-2,1:-1]-phi[2:,1:-1])
Jx[1:-1,1:-1] = 0.5*(phi[1:-1,0:-2]-phi[1:-1,2:])
plt.plot(ii[1],ii[0],'ro')
plt.quiver(Jx[:,:],Jy[:,:])
plt.title('The vector plot of current flow')
plt.show()
```