// Author: Nawaf k. Abdullah

// Date: 12-April-2017

// Simulation of Equilibrium Potential

Table of Content

1. Summery

This MATLAB script simulates electric equilibrium potential in a region of space that does not contain any electric charge, using Jacobi relaxation method.

2. Discretization

Electric potentials in a region with no interfering electric charges obeys Laplace's equation:

$$\nabla^2 V = 0$$

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0 \tag{2.1}$$

Which is an elliptic-class, second order, linear, partial differential equation. Before modeling the system, equation (2.1) must be discretized first in order to be suitable for solving numerically, which gives:

$$\frac{\partial V}{\partial x} = \frac{V(i+1,j,k) - V(i,j,k)}{\Delta x} \tag{2.2}$$

Which can also be written as:

$$\frac{\partial V}{\partial x} = \frac{V(i,j,k) - V(i-1,j,k)}{\Delta x}$$

Or:

$$\frac{\partial V}{\partial x} = \frac{V(i+1,j,k) - V(i-1,j,k)}{2\Delta x}$$

One may select the most suitable of any of the forms above depending on the problem, but essentially, they're all the same. Next, we get:

$$\frac{\partial^2 V}{\partial x^2} = \frac{1}{\Delta x}\left[\frac{\partial V}{\partial x}\left(i + \frac{1}{2}\right) - \frac{\partial V}{\partial x}\left(i - \frac{1}{2}\right)\right] \tag{2.3}$$

We plug (2.2) into (2.3):

$$\frac{\partial^2 V}{\partial x^2} = \frac{1}{\Delta x}\left[\frac{V(i+1,j,k)-V(i,j,k)}{\Delta x} - \frac{V(i,j,k)-V(i-1,j,k)}{\Delta x}\right] \tag{2.4}$$

A little rearrangement of (2.4) gives:

$$\frac{\partial^2 V}{\partial x^2} = \frac{V(i+1,j,k)+V(i-1,j,k)-2V(i,j,k)}{(\Delta x)^2} \tag{2.5}$$

We get the same form for y and z. Next, plugging (2.5) into (2.1), and then solving for V gives:

$$V(i,j,k) = \frac{1}{6}[V(i+1,j,k) + V(i-1,j,k) + V(i,j+1,k) + V(i,j-1,k) + V(i,j,k+1) + V(i,j,k-1)] \tag{2.6}$$

And that's a form that we can implement in our code, to model the system.

3. Algorithm

Since we are dealing with an elliptic equation, a good way to model our system will be to use Jacobi Relaxation method. It starts by first making an initial "guess" for the value of V at all points in i, j, and k, this guess can be anything. It can be, for example, zero for all V values at all i, j, and k, except the far sides on matrix representing the positive and negative "metal plates", illustrated in figure (3.1):

1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1

Figure (3.1): a 2D, 10x10, example of initial guess for V.

Next, equation (2.6) is applied and the previous value for V (the initial guess) is overwritten with a slightly better "guess" as shown in figure (3.2):

1	0	0	0	0	0	0	0	0	-1
1	0.25	0.0625	0	0	0	0	0	0	-1
1	0.3125	0.094	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1
1	0	0	0	0	0	0	0	0	-1

Figure (3.2): a slightly better guess for V, after applying equation (2.6) once.

This process is repeated many time, giving a better guess with each iteration, until some accuracy criterion is satisfied, and equilibrium potential is met per that criterion, as in figure (3.3).

1	0	0	0	0	0	0	0	0	-1
1	0.472	0.247	0.124	0.038	-0.038	-0.124	-0.247	-0.473	-1
1	0.644	0.391	0.210	0.066	-0.066	-0.210	-0.391	-0.644	-1
1	0.710	0.464	0.260	0.083	-0.083	-0.260	-0.464	-0.710	-1
1	0.735	0.493	0.281	0.091	-0.091	-0.281	-0.493	-0.735	-1
1	0.735	0.493	0.281	0.091	-0.091	-0.281	-0.493	-0.735	-1
1	0.710	0.464	0.260	0.083	-0.083	-0.260	-0.464	-0.7104	-1
1	0.644	0.391	0.210	0.066	-0.066	-0.210	-0.391	-0.644	-1
1	0.473	0.247	0.124	0.038	-0.038	-0.124	-0.247	-0.473	-1
1	0	0	0	0	0	0	0	0	-1

Figure (3.3): equation (2.6) is applied 400 times, giving a value for $V(i,j,k)$ that is much closer to what an exact solution would look like. Notice that a symmetry in the values of $V(i,j,k)$ has started to form, and would continue to become more obvious at more decimal places if we run algorithm at much more iterations.

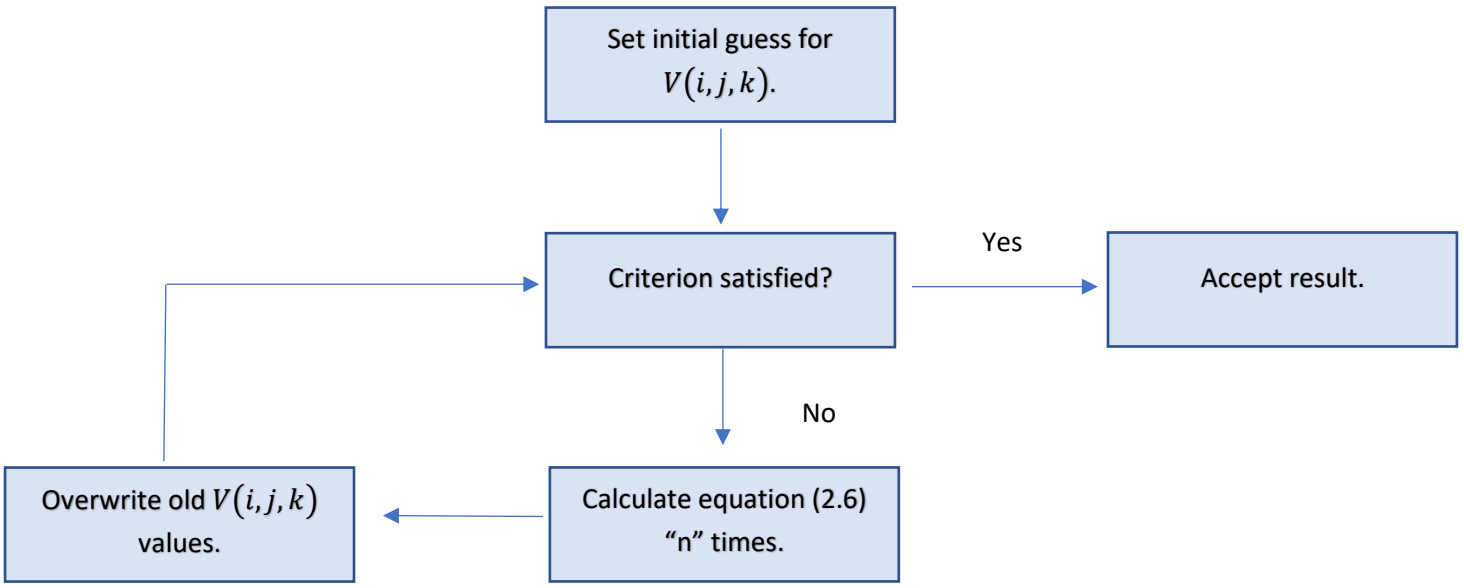

Figure (3.4): Algorithm diagram for MATLAB script.

4. 2D MATLAB Script

4.1 Script:

```
N=100; %Side length of the Box surface
V = zeros(N,N);
V(:,1)=1;
V(:,N)=-1;

for n=1:400 %Iterations for Jacobi method
    for i=2:N-1
        for j=2:N-1
            V(i,j)=(V(i-1,j)+V(i+1,j)+V(i,j+1)+V(i,j-1))*0.25;
%Discretized Laplace equation in 2D
        end
    end
end

figure(1)%Graphing equilibrium potential
surface(V)
colorbar
title('Equillibrium Potential');
hold on
set(gca,'XTick',0:10:100)
set(gca,'XTickLabel',1:-0.2:-1)
xlabel('Voltage (V)');
hold off


figure(2) %Visualization of the electric field vectors
whitebg('black')
x = -V(:,j); % Necessary to make the electric field goes from
the positive to the negative pole.
y = -V(i,:);
[xg,yg] = meshgrid(x,y);
quiver(xg',yg,'yellow');
xlim([0 102])
ylim([0 102])
title('Electric Field Direction')
hold on
set(gca,'XTick',0:10:100)
set(gca,'XTickLabel',1:-0.2:-1)
xlabel('Voltage (V)');
hold off
```

4.2 Sample run:

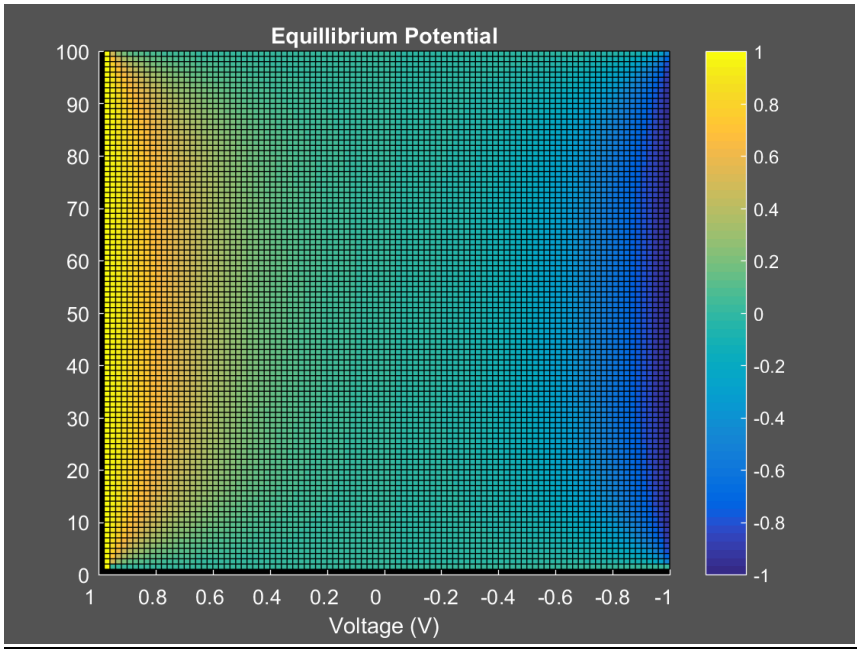

Figure (4.1): Graph of electric equilibrium potential. Yellow is the positively charged side, while the blue is the negatively charged side.

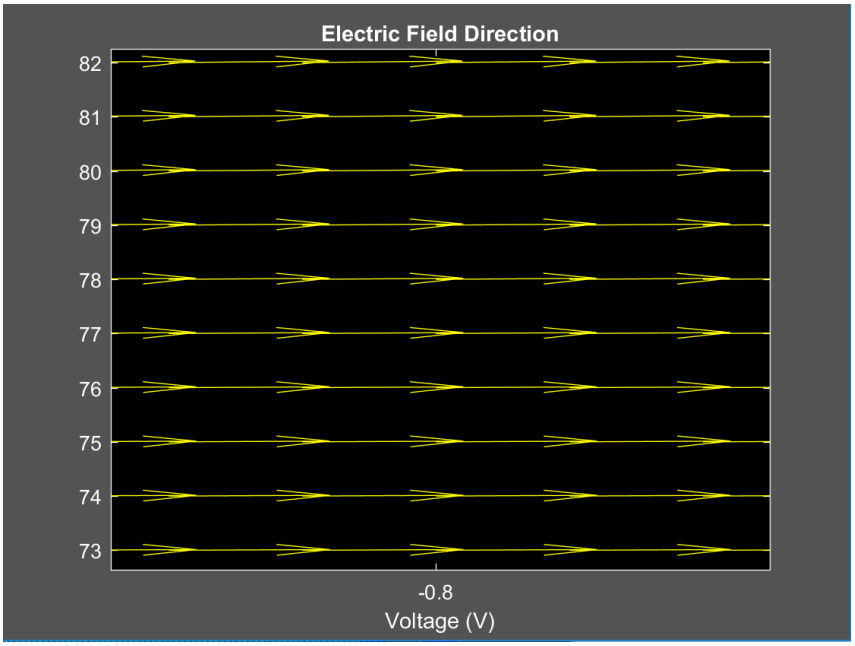

Figure (4.2): Zoomed in visualization of the electric field vectors between the two metal plates, coming from the positive side to the negative on the right.

5. 3D MATLAB script

5.1 Script:

```
N=100; %Box side length
V = zeros(N,N,N);
V(:,:,1)=1;
V(:,:,N)=-1;

for n=1:1000 %Iterations for Jacobi method
    for i=2:N-1
        for j=2:N-1
            for k=2:N-1
                V(i,j,k)=(V(i+1,j,k)+V(i-1,j,k) +V(i,j+1,k)+
V(i,j-1,k)+V(i,j,k+1)+V(i,j,k-1))*(1/6);
            end
        end
    end
end


figure(1) %Graphing equilibrium potential
xslice = [100, 50];                 % location of y-z planes
yslice = [100, 50];                 % location of x-z plane
zslice = [1, 20, 50, 80];           % location of x-y planes
slice(V,xslice,yslice,zslice)
colorbar
hold on
title('Equillibrium Potential');
set(gca,'ZTick',0:10:100)
set(gca,'ZTickLabel',1:-0.2:-1)
zlabel('Voltage (V)');
hold off

figure(2)%Visualization of the electric field vectors
x=-V(:,j,k);
y=-V(i,:,k);
z=V(i,j,:);
h=zeros(3,3);
[x,y,z]=surfnorm(h);
quiver3(h,x,y,z,'yellow')
hold on
title('Electric Field Direction')
set(gca,'xTick',[]);
set(gca,'YTick',[]);
set(gca,'zTick',[0:10:100]);
set(gca,'ZTickLabel',1:-0.2:-1)
hold off
```

5.2 Sample Run:

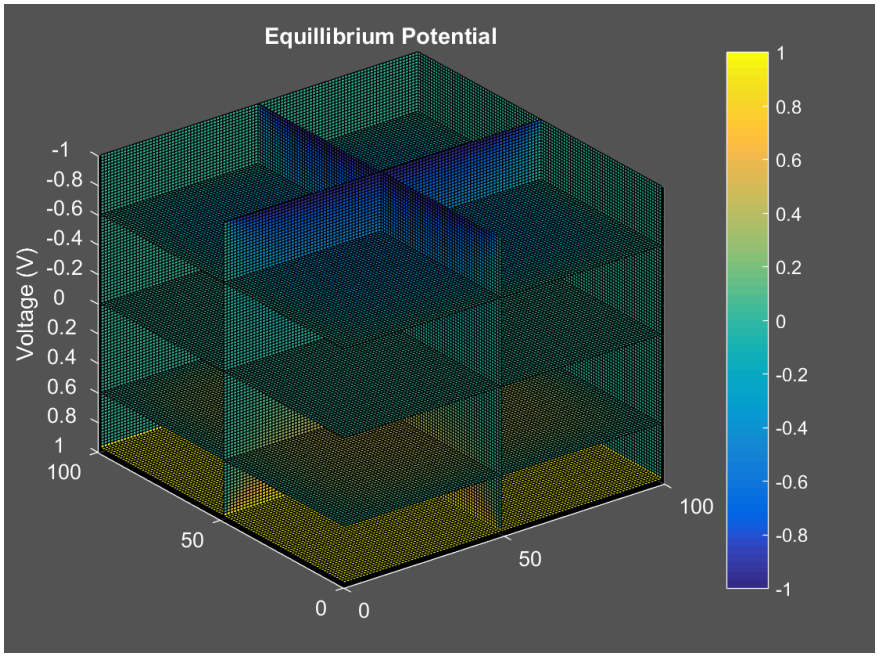

Figure (5.1): 3D Visualization of the electric equilibrium potential. Different slices of the "box" illustrates the voltage at different locations in the Z-axis.

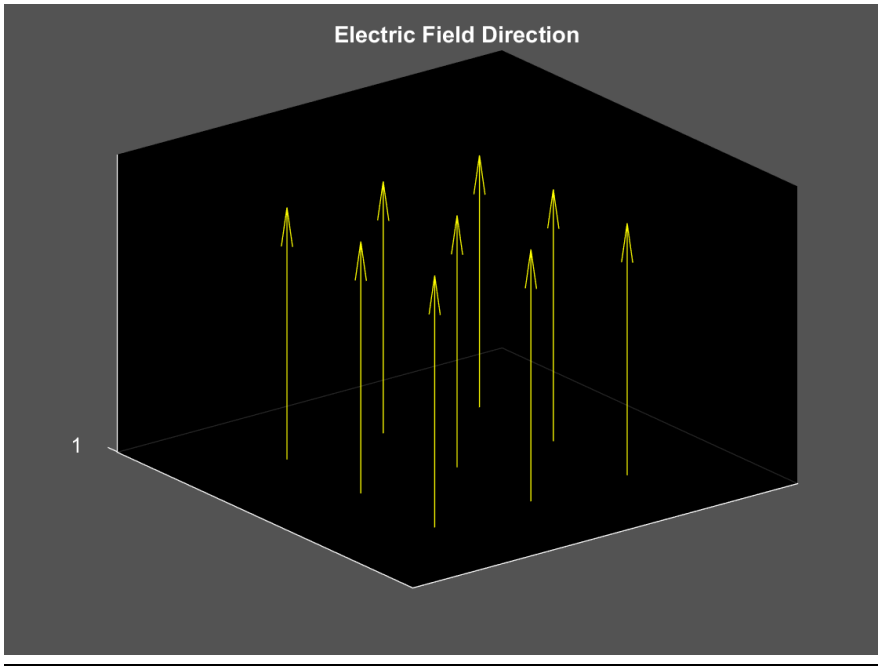

Figure (5.2): Visualization of the electric field vectors inside the box, coming from the positive side to the negative on the top.