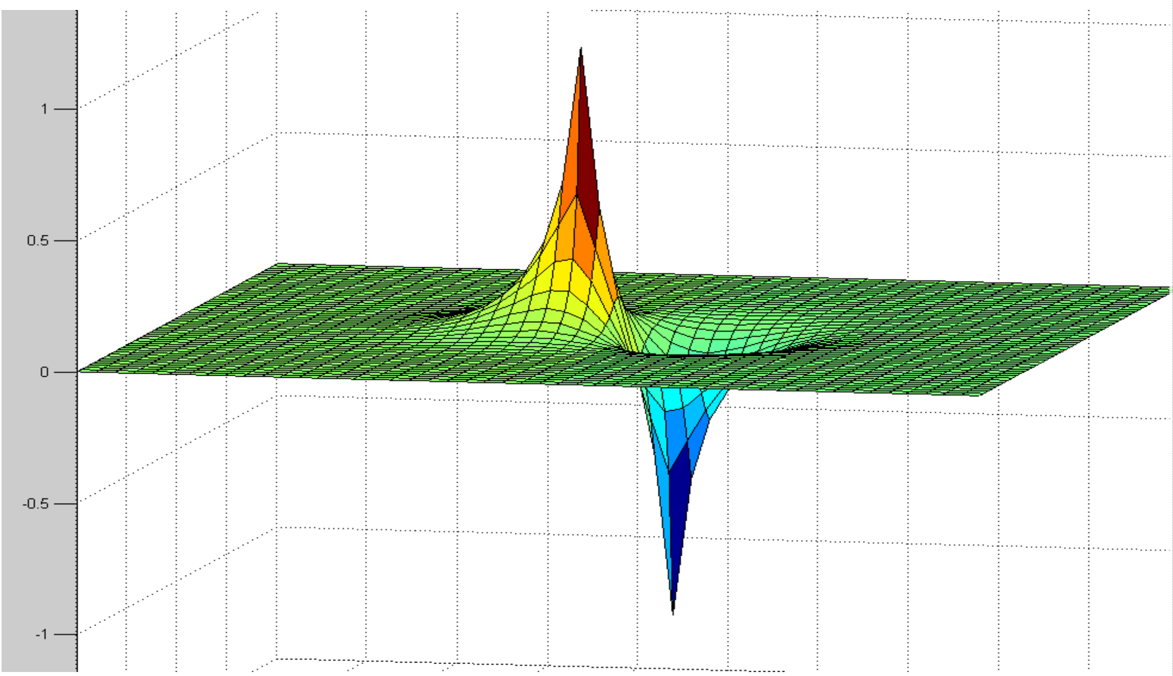
Electric Field for Stationary Charge Distribution

Solving Poisson's equation in 2D and 3D



Introduction

This report mainly presents the procedure of solving Poisson's equation to find and plot the electric field and voltage due to a given stationary charge distribution using finite difference method. Here the charges can be specified in different location or the charge density can be given as a function of space. And as the output we show the electric field and voltage created by those charges in 2D and 3D space. As an application we have shown the electric fields due a dipole and a parallel plate capacitor. We solve Poisson's equation to get the desired electric field.

The Poisson equation is a very powerful tool for modeling the behavior of electrostatic systems, but unfortunately may only be solved analytically for very simplified models. Consequently, numerical simulation must be utilized in order to model the behavior of complex geometries with practical value. Although there are several competing algorithms for achieving this goal, one of the simple stand more straightforward of these is called the *finite-difference method* (FDM). At its core , FDM is nothing more than a direct conversion of the Poisson equation from continuous functions and operators into their discretely-sampled counterparts. This converts the entire problem into a system of linear equations that may be readily solved via matrix inversion. The accuracy of such a method is therefore directly tied to the ability of a finite grid to approximate a continuous system, and errors may be arbitrarily reduced by simply increasing the number of samples.

Beginning with Maxwell's equations, the ultimate governing equation for any electrostatic system is Gauss's law. Expressed in point form, this may be written as

$$\nabla \cdot \mathbf{D}(\mathbf{r}) = \rho(\mathbf{r})$$

In this context, **r** is a position vector in space, ρ is the charge density function, and **D** is the electric flux density. Using the constitutive relation

$$\mathbf{D}(\mathbf{r}) = \epsilon_0 \epsilon(\mathbf{r}) \mathbf{E}(\mathbf{r})$$

Gauss's law maybe rewritten in terms of the electric field intensity **E** as

$$\nabla \cdot \Big[\epsilon(\mathbf{r})\ \mathbf{E}(\mathbf{r})\Big] = \frac{\rho(\mathbf{r})}{\epsilon_0}$$

Gauss's law may be further rewritten in terms of the *voltage potential function V* (**r**) by making the substitution $\mathbf{E}(\mathbf{r}) = -\nabla V(\mathbf{r})$

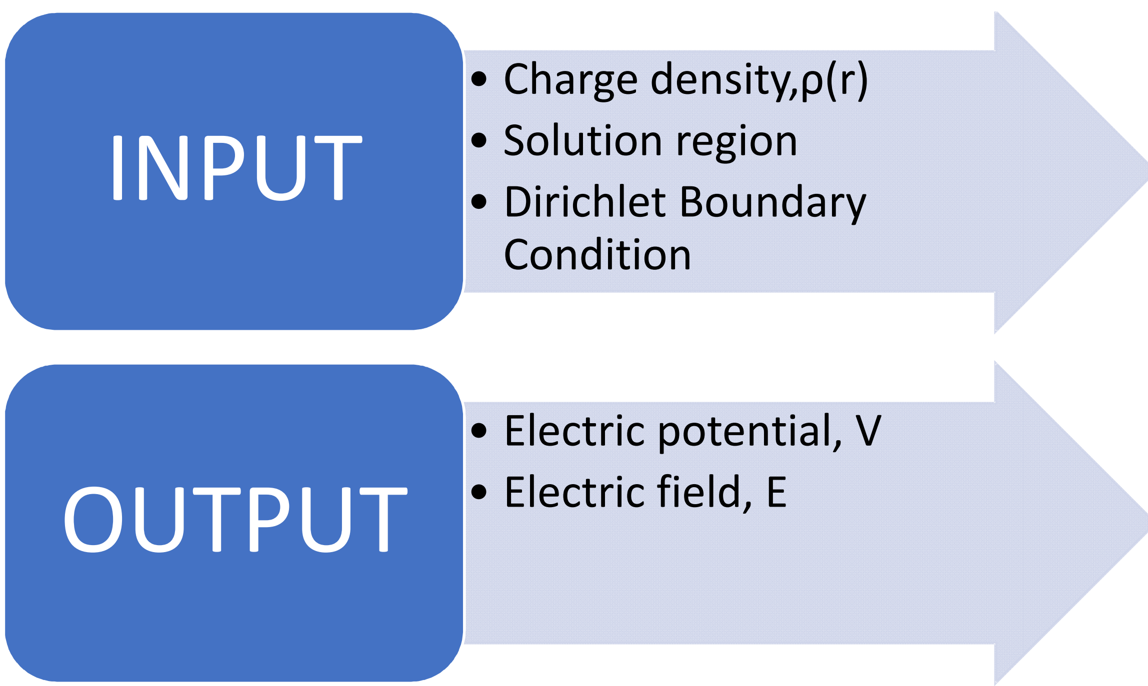
$$\nabla \cdot \Big[\epsilon(\mathbf{r})\ \nabla V(\mathbf{r})\Big] = -\frac{\rho(\mathbf{r})}{\epsilon_0}\ .$$

Although it is not commonly discussed in the literature, this is really nothing more than a generalized form of the *Poisson equation*, and is the expression we shall be most interested in throughout this report. But the final classical form of poisson's equation is

$$\nabla^2 V(\mathbf{r}) = -\frac{\rho(\mathbf{r})}{\epsilon_0 \epsilon_r}\ ,$$

Problem Statement

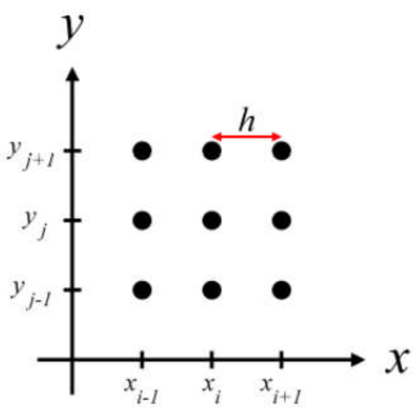
We have to compute electric field and electric potential for any arbitrary charge distribution in space. So mathematically we are given a charge density function $\rho(r)$ which represents how charge is distributed in space as input. Then the region for which we have to solve for electric field is given. For 2D, a square region and for 3D, a cubic region is given as solution region. Now we can solve Poisson's equation to compute electric potential if we are given proper boundary condition. We take Dirichlet Boundary Condition where electric potential at boundary points are given.



Solving algorithm

1. **Discretizing space and selecting the solution domain**

The first step in applying FDM is to define a *mesh*, which is simply a uniform grid of spatial points at which the voltage function will be sampled. Letting *h* be the distance between each sample, the points that lie on the mesh may be defined by
$x_i = ih$
$y_j = jh$



where *i* and *j* are integers. In practice, *i* and *j* will eventually be used as indices for a matrix of voltage samples, so it helps to use a short-hand notation that bears this in mind. We shall therefore replace the spatial coordinates with simple indices by assuming the following convention:

$$V(i, j) = V(x_i, y_j) .$$

In a similar fashion, we may also define the charge density samples along the same mesh by using the $\rho(i, j)$ notation. The next step is to expand the Poisson equation by explicitly showing the partial derivatives in space:

$$\frac{\partial^2 V(i, j)}{\partial x^2} + \frac{\partial^2 V(i, j)}{\partial y^2} = -\frac{\rho(i, j)}{\epsilon_0} .$$

The reason for doing this is so that we may approximate the derivative operators through the use of finite-differences. The easiest way to do this is through the three-point approximation for the second-derivative, which is given as

$$\frac{\partial^2}{\partial x^2}V(i,j) \approx \frac{V(i-1,j) - 2V(i,j) + V(i+1,j)}{h^2},$$

with a similar expression for the partial derivative with respect to *y*.

We solved the equation in a **square domain** in XY plane (2D) and **cubic domain** in XYZ space. The co-ordinate of the corner of the region is given as **input**. Then number of points in each direction-> 'N' is taken. After that, we divide the domain into N*N (N*N*N for 3D) grids. These discretized space is then represented by X, Y (X, Y, Z) matrix .

2. Setting charge distribution

A charge density matrix **rho** of N*N*N order is set up. Charge density function is given as input. Then the rho values at the sample points are calculated. But sometimes the charges can be given as discrete points instead of a continuous function.

3. Setting the boundary condition

We used **Dirichlet Boundary condition,** which means we have specified electrical potential values which we are going to solve at the boundary points. Having the boundary conditions we can start solving for the poisson's equation.

4. Solving the finite difference equation by iteration

Finite difference equation for V is ->

For 2D; $$V(i,j) = \frac{1}{4}\left(V(i-1,j) + V(i+1,j) + V(i,j-1) + V(i,j+1) + \frac{h^2\rho}{\epsilon}\right)$$

For 3D; $$V(i,j,k) = \frac{1}{6}\left(V(i-1,j,k) + V(i+1,j,k) + V(i,j-1,k) + V(i,j+1,k) + V(i,j,k-1) + V(i,j,k+1) + \frac{h^2\rho}{\epsilon}\right)$$

We set the potential matrix **V** to have all zero entry. Then We iteratively solve for V(I,j,k) using the previous values of V. The iteration is guaranteed to converge.

5. Solving electric field from scalar potential and normalising electric field

We know that, electric field is the negative gradient of potential. After solving for potential at every point, we take partial derivatives and get electric field component Ex, Ey, Ez. Then we normalize E field by diving each component by the electric field value.

6. Solution visualization

For 2D, we can visualize E-field by vector field diagram and potential by surface plot and contour curves.
For 3D, we can visualize E-field by vector field diagram and potential by contour surface.

Analysis of results

It is difficult to intuitively understand the behavior of electric field for any arbitrary charge distribution. So we have simulated some common situations like electric dipole, parallel plate capacitor.

Charge density function is modified to create a dipole. We get a electric field like figure 1.

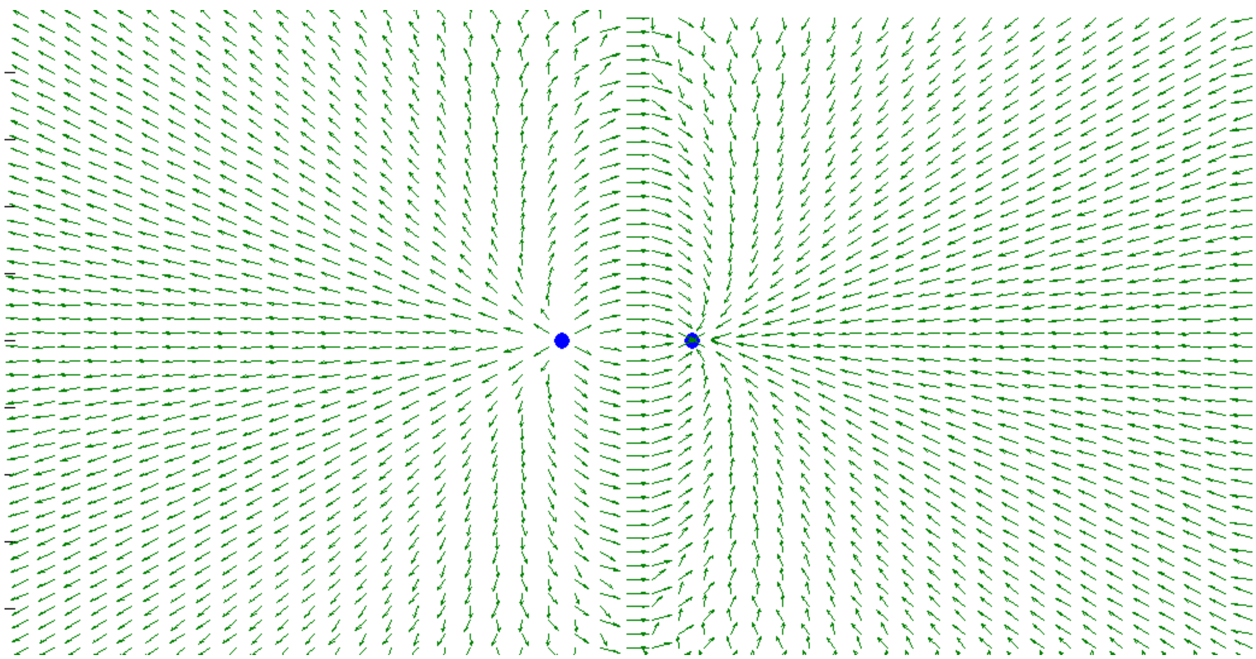


Figure 1

Electric field lines are coming off radially outward from the positive charge and converging to the negative charge which is expected. Next the electric potential is plotted as 3D surface (figure 2).

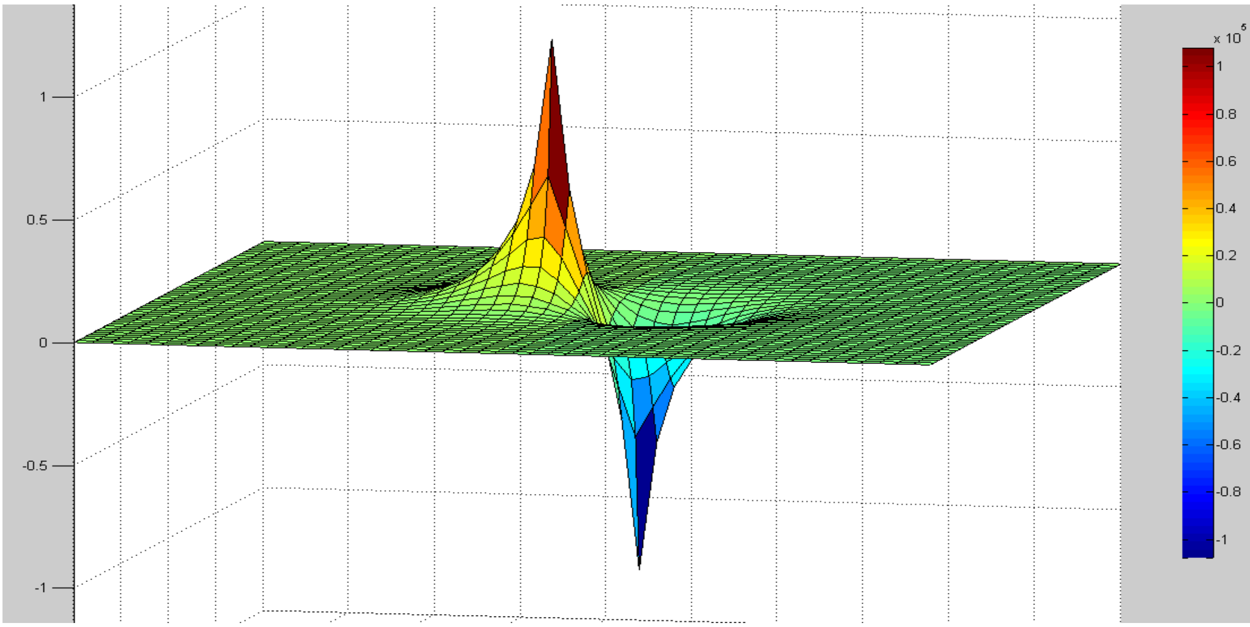


Figure 2

Electric potential near the positive charge is positive and for the negative charge it is negative. Potential at distant places are close to zero.

Next we model parallel plate capacitor with a lots of point charge along capacitor plate. The capacitor electric field comes out to be like figure 3 and 4.

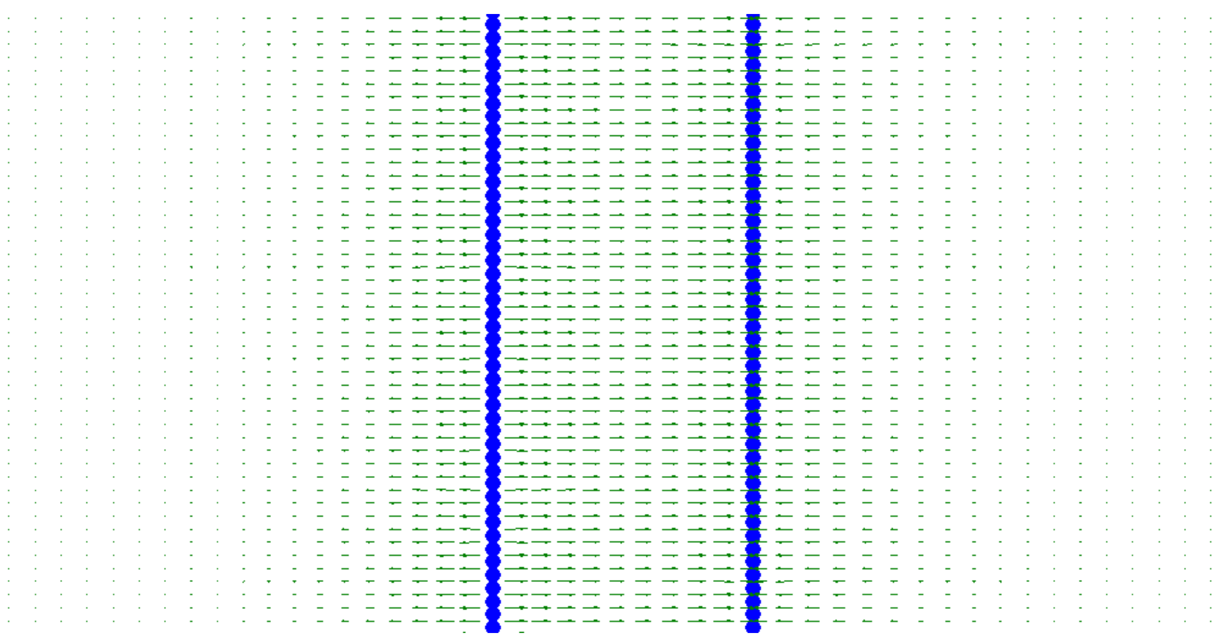


Figure 3

Electric field inside the capacitor is uniform. And electric field outside the capacitor is dying out as we see in figure

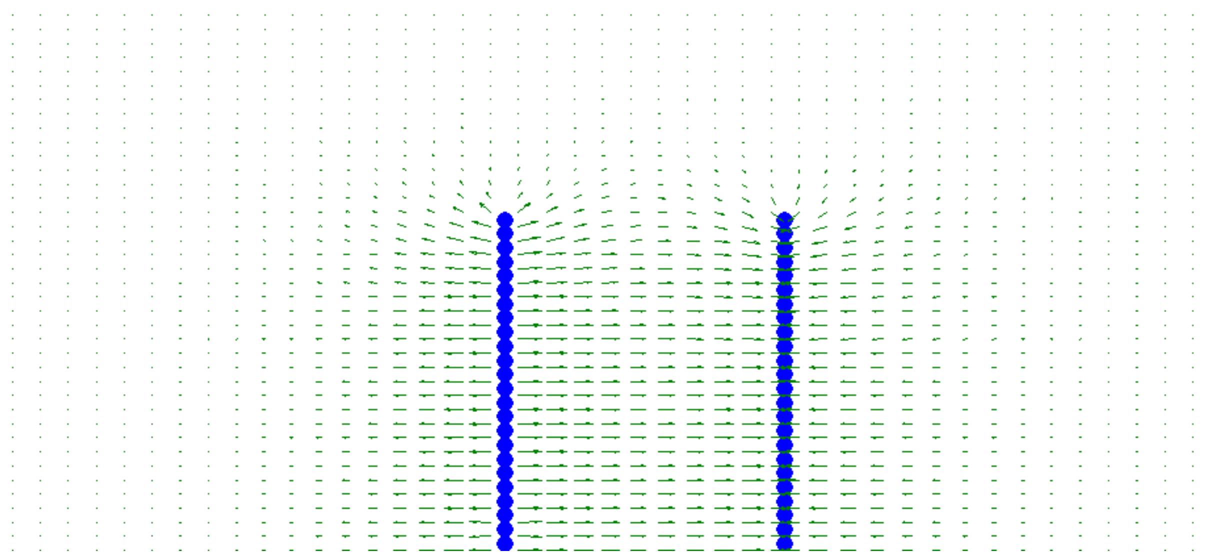


Figure 4

Electric field lines near the edges are not truly parallel and a little of fringing occurs. Figure 4 illustrates the fact. Next we plot the potential of the capacitor.

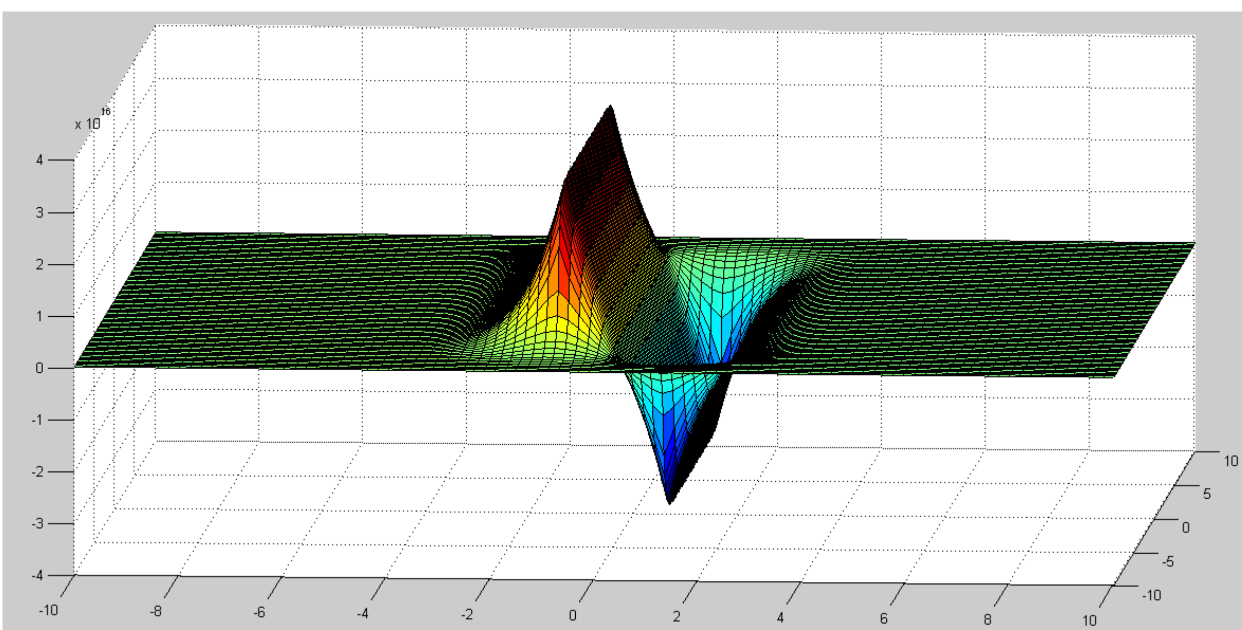


Figure 5

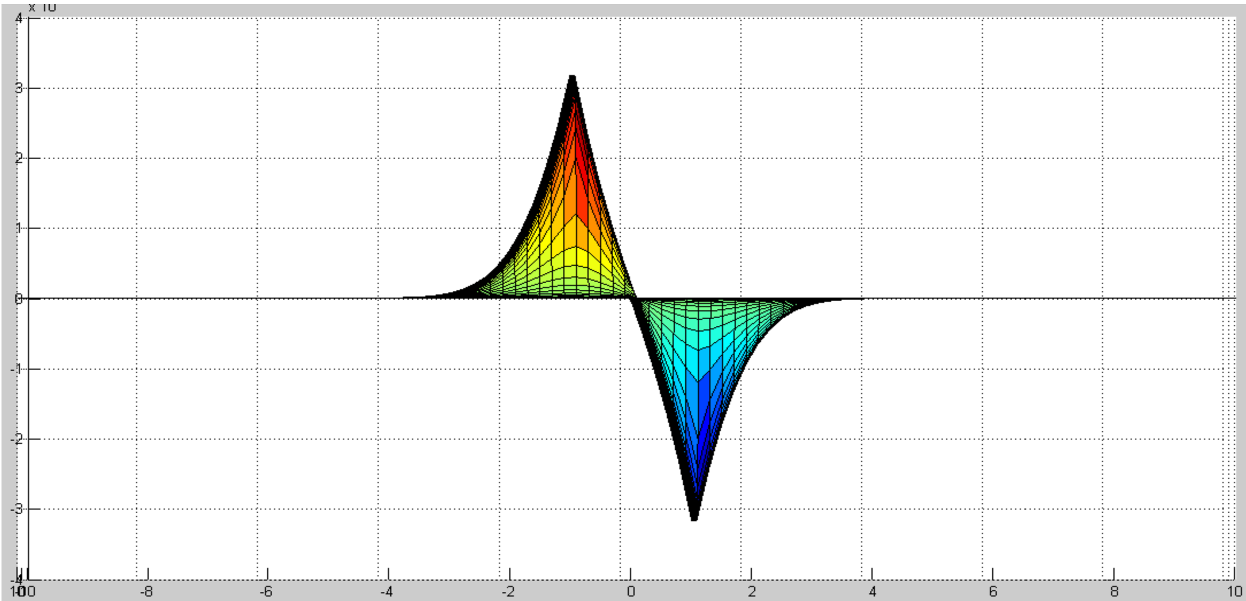


Figure 6

Figure 6 is the front view of figure 5. Figure 6 shows that electric potential inside capacitor changes linearly with distance. And it is true as electric field inside is uniform.

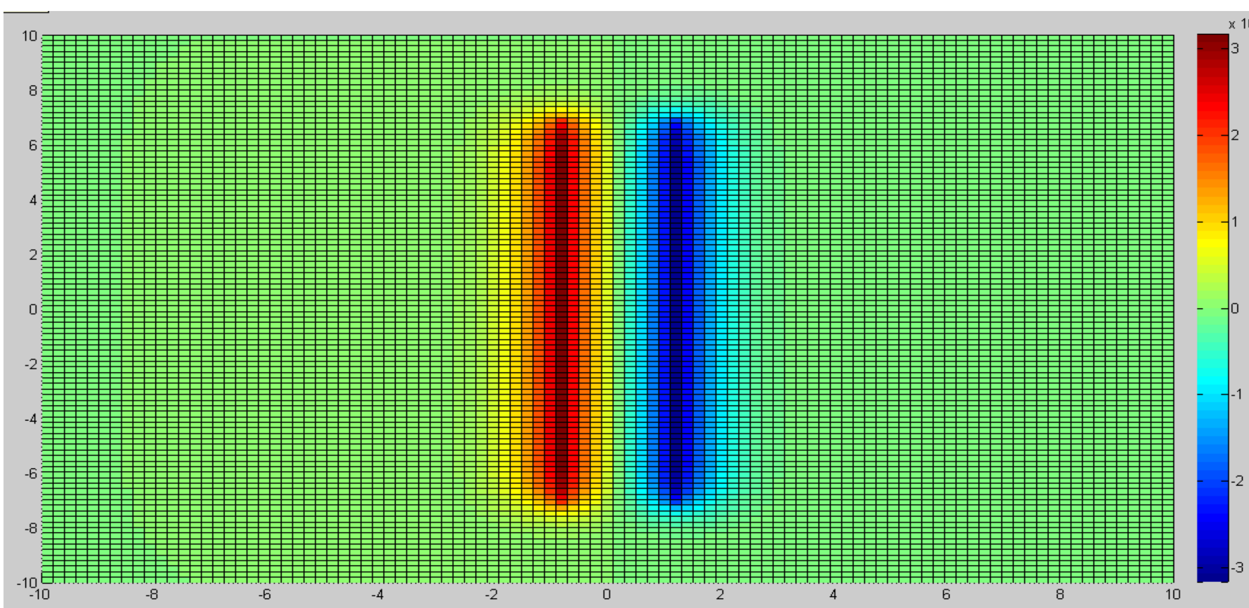


Figure 7

Figure 7 is the top view of figure 5. We see from figure 7 that potential across the capacitor plates are constant. All of these conform to the actual results.

Conclusion

We have solved Poisson's equation for any arbitrary charge distribution. We also have simulated some special situations like electric dipole, parallel plate capacitor and the results are satisfactory. But there are things which can be improved in future. We can allow Newmann Boundary condition. Also the boundary values need not be constant, they can be function of boundary points. Graphical representations can be improved especially in case of plotting the solution of 3D Poisson's equation. But overall, the main objectives of this project are achieved.

References

1. David J. Griffiths, *Introduction to Electrodynamics,* Pearson, Fourth Edition.
2. https://www.rsmas.miami.edu/users/miskandarani/Courses/MSC321/Projects/prjpoisson.pdf
3. http://www.univ-tebessa.dz/fichiers/ouargla/IJOPCM%28vol.5.4.17.D.12%29.pdf
4. https://www.mathworks.com

Matlab Code

1. 2D Poisson's Equation

```
%%Simulation region
% (a,b) -> co-ordinate of lower left corner of the square region
% (c,d) -> co-ordinate of upper right corner of the square region
% (c-a)==(d-b)
% N -> Number of points in any direction   % h -> grid spacing
clc;
clear all;
close all;
a=-10   ; b=-10   ;
c=10   ; d=10   ;
N=80 ;
h=(c-a)/N;
eps=8.854E-12;

x = linspace(a,c,N);   % grid points x including boundaries
y = linspace(b,d,N);   % grid points y including boundaries

[X,Y] = meshgrid(x,y);      % 2d arrays of x,y values
Y=-Y;

%%charge distribution

rho=zeros(N,N);
f=@(x,y) (1/((x)^2+(y)^2));
for i=1:N
    for j=1:N
        rho(i,j)=f(X(i,j),Y(i,j));
    end
end


%%Boundary Conditions (dirichlet condition) ->|v <-|^
% V -> Electric Potential
v=zeros(N,N);
v(1, 1:N) = 1000;
v(1:N, N)=  1000;
v(N, 1:N)=  1000;
v(1:N, 1)=  1000;
 vi=rho.*(1/4)*(1/eps)*h*h;
%%FDM to solve for potential
for iter=1:50
     vi=rho.*(1/4)*(1/eps)*h*h;
for i=2:N-1
    for j=2:N-1
        vi(i,j)=vi(i,j)+(1/4)*(v(i-1,j)+v(i+1,j)+v(i,j-1)+v(i,j+1));
    end
end
v=vi;
end
```

```matlab
%%Determinint the Electric field intensity

ex=zeros(N,N);
ey=zeros(N,N);

for i=2:N-1
    for j=2:N-1
        ey(i,j)=-(v(i,j)-v(i+1,j))/h;
        ex(i,j)=(v(i,j)-v(i,j+1))/h;
    end
end

for i=2:N-1
    for j=2:N-1
        ey(i,j)=(ey(i,j+1)+ey(i,j))/2;
        ex(i,j)=(ex(i+1,j)+ex(i,j))/2;
    end
end



%Normalising the Electric field
e=zeros(N,N);
for i=2:N-1
    for j=2:N-1
        e(i,j)=(ex(i,j))^2 + (ey(i,j))^2;
        e(i,j)=sqrt(e(i,j));
    end
end

for i=2:N-1
    for j=2:N-1
        if (e(i,j)~=0)
            ex(i,j)=ex(i,j)/e(i,j);
            ey(i,j)=ey(i,j)/e(i,j);
        end
    end
end

%%Graphical representation
contourf(X,Y,rho);
hold on
quiver(X,Y,ex,ey,'autoscalefactor',0.6);
figure
surf(X,Y,v)
hold on
```

2. 3D Poisson's equation

```
%%Simulation region
% (a,b) -> range of x values
% (c,d) -> range of y values
% (e,f) -> range of z values
% N -> Number of points in any direction   % h -> grid spacing
clc;
clear all;
close all;
a=-5   ; b=5   ;
c=-5   ; d=5   ;
e=-5; f=5;
N=20 ;
h=(b-a)/N;
eps=8.854E-12;

x = linspace(a,b,N);   % grid points x including boundaries
y = linspace(c,d,N);   % grid points y including boundaries
z=linspace(e,f,N);     % grid points z including boundaries
[X,Y,Z] = meshgrid(x,y,z);       % 2d arrays of x,y values

Y=-Y;
Z=-Z;

%%charge distribution

rho=zeros(N,N,N);
f=@(x,y,z) (1/(((x)^2+(y)^2+(z)^2)));
for i=1:N
    for j=1:N
        for k=1:N
        rho(i,j,k)=f(X(i,j,k),Y(i,j,k),Z(i,j,k));
        end
    end
end

%%Boundary Conditions (dirichlet condition) ->|v <-|^
% V -> Electric Potential
v=zeros(N,N,N);
 %boundary conditions
v(1, 1:N)  = 0;
v(1:N, N)=  0;
v(N, 1:N)=  0;
v(1:N, 1)=  0;

%%FDM to solve for potential
for iter=1:20
    vi=rho.*(1/6)*(1/eps)*h*h;
for i=2:N-1
    for j=2:N-1
        for k=2:N-1
         vi(i,j,k)=vi(i,j,k)+(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));
```

```
        end
    end
end
v=vi;
end

%%Determininint the Electric field intensity

ex=zeros(N,N,N);
ey=zeros(N,N,N);
ez=zeros(N,N,N);
 for i=2:N-1
    for j=2:N-1
        for k=2:N-1
        ey(i,j,k)=-(v(i,j,k)-v(i+1,j,k))/h;
        ex(i,j,k)=(v(i,j,k)-v(i,j+1,k))/h;
        ez(i,j,k)=-(v(i,j,k)-v(i,j,k+1))/h;
        end
    end
end

 for i=2:N-1
    for j=2:N-1
        for k=2:N-1
        ey(i,j,k)=(ey(i,j+1,k)+ey(i,j,k))/2;
        ex(i,j,k)=(ex(i+1,j,k)+ex(i,j,k))/2;
        ez(i,j,k)=(ez(i,j,k+1)+ez(i,j,k))/2;
        end
    end
end

%Normalising the Electric field
e=zeros(N,N,N);
for i=2:N-1
    for j=2:N-1
        for k=2:N-1
        e(i,j,k)=(ex(i,j,k))^2 + (ey(i,j,k))^2+(ez(i,j,k))^2;
        e(i,j,k)=sqrt(e(i,j,k));
        end
    end
end

for i=2:N-1
    for j=2:N-1
        for j=2:N-1
        if (e(i,j,k)~=0)
            ex(i,j,k)=ex(i,j,k)/e(i,j,k);
            ey(i,j,k)=ey(i,j,k)/e(i,j,k);
            ez(i,j,k)=ez(i,j,k)/e(i,j,k);
        end
        end
    end
end

%%Graphical representation
quiver3(X,Y,Z,ex,ey,ez,'autoscalefactor',3);
```