## Abstract

In this project will try to solve poisson equation for interleaved capacitor computationally using the method of finite differences to convert them into a system of linear equation and then using iterative schemes like SOR, Gauss Seidel Method , Jacobi Method , SOR (Successive Over Relaxation ) Method. We would also like to develop intuiton for poisson equation related problems related to electrostatic potential and compare the different iterative schemes named above on the basis of number of iterations to find the most compuationally viable method.We would also like to discuss the topics such as truncation error and non dimensionalisation which are involved in coverting the physical poisson problem into computational problem.

# Contents

# 1 Introduction

## 1.1 Motvation

We first encountered Laplace Equation during our course in electricity and magnetism in second semester and we were fascinated with how one can calculatge the potential in a region just by knowing the boundary condition, ofcourse the region has to be charge free for applying Laplace Equation. After Laplace Equation , we were introduced to Poisson Equation which we were able to solve for region having charges( or sources ). When we were given the oppurtunity to choose a project in our computational physics this semester, it did not take us long to decide the topic for project.

## 1.2 General Idea

In our project we will try to tackle the Laplace and Poisson equaton which is an ellipitic linear partial differential equation having application in various fields of physics ranging from thermodynamics, electrostatics etc. We will solve the equation computationally using the method of finite differences in one and two dimensions for rectangular membrane. We will first convert the partial differential equation into system of linear differential equation and then use three different iteration schemes for solving those system of linear equation.

## 1.3 Poisson and Laplace Equation

The general form of Poisson's equation in Euclidean space is given here,

$$\vec{\nabla}^2 \varphi(\vec{r}) = f(\vec{r})$$

where, $f(\vec{r})$, $\varphi(\vec{r})$ are the functions of position vector $\vec{r}$. $f(\vec{r})$ is given and $\varphi(\vec{r})$ is sought.
In our case $\varphi(\vec{r})$ is electrostatic potential, $f(\vec{r})$ is the charge density of region in which solution is required. In a special case in which $f(\vec{r}) = 0$ we can apply same process for solving laplace equation for a region free of charge.

## 1.4 Plan of Report

In theory section we will formulate and explain the physical problem we chosen to solve using finite difference method. In methodology we will explain our methods and explain the algorithm followed for programming. In the Result and anlaysis section we analyse our results compared to what we expect to obtain and also compare different iterative methods based on our computation relating problem.And in computation we would like to discuss our experiences and results in brief.

## 2 Theory

Identical infinitely-long thin metal plates $A_1, A_2, A_3, A_4, A_5, B_1, B_2, B_3$ and $B_4$ are placed in an *interleaved* arrangement as dipicted in fig1.(a) Group of plates $A_i$ and $B_i$ are connected to the terminal $A$ and $B$ , respectively, by means of gold wires. The terminals are connected to different potential sources $U_A$ and $U_B$ respectively.
This is called an interleaved capacitor, Our goal will be to approximate the potential distribution (in two dimensions considering symmetry along the the third axis which will be dropped) inside the capacitor after we disconnect the capacitor from sources, treating the interior plates as line charge distributions.

$$\begin{aligned}
&U_A = 5V \\
&U_B = -5V \\
&\text{Capacitance } = 0.1\mu F \\
&\text{Distance between plates } (d) = 0.5\mu m \\
&\textit{Dimensions: } 4 \times 4.4\mu m
\end{aligned}$$

This arrangement can be seen as a number of parallel plate capacitors connected in parallel to each other as seen in fig1(b), if $C_0$ is the capacitance of each capacitor in parallel and $C$ is the capacitance of the entire arrangement then,

$$C_0 = C/8$$

Also we know that for parallel plate capacitors with cross-section area $A$ and distance $d$ between the plates,

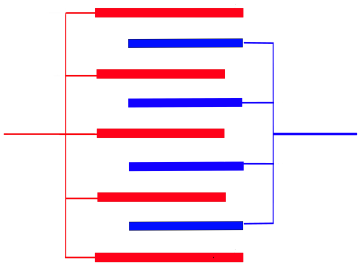
$$C_0 = \epsilon_0 \frac{A}{d} \implies A = \frac{C_0 d}{\epsilon_0} \tag{2.1}$$

Therefore the charge distribution on any plate $A_i$ is given by,

$$\rho_A = \frac{(C \times V)}{5A} \implies \rho_A = 2\epsilon_0 \times 10^5 Cm^{-2} \tag{2.2}$$

Similarly on $B_i$,

$$\rho_B = -2\epsilon_0 \times 10^5 Cm^{-2} \tag{2.3}$$



(a) Diagram dipciting the arrangement of plates in a interleaved fashion.

Mathematical Formulation:
Let $U(x, y)$ and $\vec{E}(x, y)$ be the potential and Electric field distribution defined in the region of our arrangement $(x, y) = [0, 4\mu m] \times [0, 4.4\mu m]$ .
We know from maxwell's laws that for static electric fields $\vec{\nabla} \cdot \vec{E} = \rho/\epsilon_0$, $\vec{\nabla} \times \vec{E} = 0$
From the later we get $E = -\vec{\nabla} U$, substituting this back into the former we get,

$$\vec{\nabla}^2 U = \frac{-\rho}{\epsilon_0}$$

In two dimensions with euclidean coordinate system the equation reduces to,

$$\frac{\partial^2 U(x,y)}{\partial x^2} + \frac{\partial^2 U(x,y)}{\partial y^2} = -\frac{\rho(x,y)}{\epsilon_0} \tag{2.4}$$

where,

$$\rho(x,y) = \begin{cases} -2\epsilon_0 \times 10^5 Cm^{-2} & : \text{if} \quad (x,y) \in B_i \quad \text{where} \quad i = 1,2,3,4 \\ 2\epsilon_0 \times 10^5 Cm^{-2} & : \text{if} \quad (x,y) \in A_i \quad \text{where} \quad i = 2,3,4 \\ 0Cm^{-2} & : \quad \text{elsewhere} \end{cases} \tag{2.5}$$

Now according to our given arrangement,

$$B_1 = \{(x^*, y^*) \quad : \quad x^* = 0.5\mu m \quad ; \quad 0.4\mu m \leq y^* \leq 4.4\mu m\} \tag{2.6}$$
$$B_2 = \{(x^*, y^*) \quad : \quad x^* = 1.5\mu m \quad ; \quad 0.4\mu m \leq y^* \leq 4.4\mu m\} \tag{2.7}$$
$$B_3 = \{(x^*, y^*) \quad : \quad x^* = 2.5\mu m \quad ; \quad 0.4\mu m \leq y^* \leq 4.4\mu m\} \tag{2.8}$$
$$B_4 = \{(x^*, y^*) \quad : \quad x^* = 3.5\mu m \quad ; \quad 0.4\mu m \leq y^* \leq 4.4\mu m\} \tag{2.9}$$
$$A_2 = \{(x^*, y^*) \quad : \quad x^* = 1\mu m \quad ; \quad 0\mu m \leq y^* \leq 4\mu m\} \tag{2.10}$$
$$A_3 = \{(x^*, y^*) \quad : \quad x^* = 2\mu m \quad ; \quad 0\mu m \leq y^* \leq 4\mu m\} \tag{2.11}$$
$$A_4 = \{(x^*, y^*) \quad : \quad x^* = 3\mu m \quad ; \quad 0\mu m \leq y^* \leq 4\mu m\} \tag{2.12}$$

We get the following boundary conditions for the above boundary value problem,

$$U(0,y) = +5V \quad ; \quad U(4,y) = +5V \tag{2.13}$$
$$U_y(x,0) = 0V/m \quad ; \quad U_y(x,4.4) = 0V/m \tag{2.14}$$

Non-Dimensionalizing the variables,
Let new dimensionless variables,

$$x' = \frac{x}{s} \qquad y' = \frac{y}{s} \qquad U' = \frac{U}{\nu} \tag{2.15}$$

where $s$ and $\nu$ are known constant scaling factors having dimensions of length and electric potential respectively.

The B.V.P reduces to,

$$\frac{\partial^2 U'(x',y')}{\partial x'^2} + \frac{\partial^2 U'(x',y')}{\partial y'^2} = -\frac{\rho'(x',y')s^2}{\epsilon_0 \nu} \tag{2.16}$$

where,

$$\rho'(x',y') = \begin{cases} -2\epsilon_0 \times 10^5 & : \text{if} \quad (x',y') \in B_i \quad \text{where} \quad i = 1,2,3,4 \\ 2\epsilon_0 \times 10^5 & : \text{if} \quad (x',y') \in A_i \quad \text{where} \quad i = 2,3,4 \\ 0 & : \quad \text{elsewhere} \end{cases} \tag{2.17}$$

Now according to our given arrangement,

$$B_1 = \{(x^*, y^*) \quad : \quad x^* = 0.5/s \quad ; \quad 0.4/s \leq y^* \leq 4.4/s\} \tag{2.18}$$
$$B_2 = \{(x^*, y^*) \quad : \quad x^* = 1.5/s \quad ; \quad 0.4/s \leq y^* \leq 4.4/s\} \tag{2.19}$$
$$B_3 = \{(x^*, y^*) \quad : \quad x^* = 2.5/s \quad ; \quad 0.4/s \leq y^* \leq 4.4/s\} \tag{2.20}$$
$$B_4 = \{(x^*, y^*) \quad : \quad x^* = 3.5/s \quad ; \quad 0.4/s \leq y^* \leq 4.4/s\} \tag{2.21}$$
$$A_2 = \{(x^*, y^*) \quad : \quad x^* = 1/s \quad ; \quad 0/s \leq y^* \leq 4/s\} \tag{2.22}$$
$$A_3 = \{(x^*, y^*) \quad : \quad x^* = 2/s \quad ; \quad 0/s \leq y^* \leq 4/s\} \tag{2.23}$$
$$A_4 = \{(x^*, y^*) \quad : \quad x^* = 3/s \quad ; \quad 0/s \leq y^* \leq 4/s\} \tag{2.24}$$

We get the following boundary conditions for the above boundary value problem,

$$U(0,y) = +5/\nu \quad ; \quad U(4,y) = +5/\nu \tag{2.25}$$
$$U_y(x,0) = 0 \quad ; \quad U_y(x,4.4) = 0 \tag{2.26}$$

# 3 Methodology

## 3.1 Finite Difference Method

Finite Difference Methods(FDM) are used for approximating the solution of partial differential equations over a set of finite points, arranged in a geometrical structure called a **mesh**[1], in the continous domain of solution. The methods involve the idea of reducing the given PDE, by means of truncated taylor series approximation of the derivatives, to a difference equation which is much easier to digest numerically.

### 3.1.1 Finite Difference Approximations

The quality of the solution depends on the quality of approximations made to the derivatives. Consider this one-dimensional structured mesh of nodes $(x_0, x_1, x_2, .., x_i, .., x_n)$ at which the solution $U(x_i)$ is to be found, such that the difference $h = x_{i+1} - x_i$ is constant throughout the mesh and $x_i \equiv x_0 + ih$.

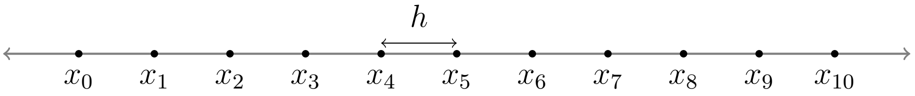
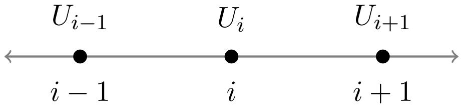


Figure 2: 1D mesh with 11 nodes and a meshsize h



Let $U_i$ represent the solution at the $i$-th node and

$$\left.\frac{\partial U}{\partial x}\right|_{x_i} = U_x(x_0 + ih) \equiv U_x|_i$$

$$\left.\frac{\partial^2 U}{\partial x^2}\right|_{x_i} = U_{xx}(x_0 + ih) \equiv U_{xx}|_i$$

The first order derivative can be defined as,

$$U_x|_i = \lim_{h\to 0} \frac{U_{i+1} - U_i}{h}$$

$$\text{or,} \qquad U_x|_i = \lim_{h\to 0} \frac{U_i - U_{i-1}}{h}$$

$$\text{or,} \qquad U_x|_i = \lim_{h\to 0} \frac{U_{i+1} - U_{i-1}}{2h}$$

Finite difference approximations are obtained by dropping the limit and can be written as,

$$\text{Forward Difference} \qquad U_x|_i \approx \frac{U_{i+1} - U_i}{h} \equiv \delta_x^+ U_i$$

$$\text{Backward Difference} \qquad U_x|_i \approx \frac{U_i - U_{i-1}}{h} \equiv \delta_x^- U_i$$

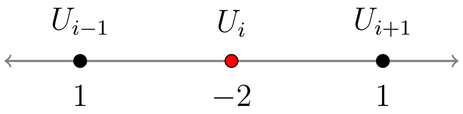
$$\text{Central Difference} \qquad U_x|_i \approx \frac{U_{i+1} - U_{i-1}}{2h} \equiv \delta_{2x} U_i$$

Where $\delta_x^+, \delta_x^-, \delta_{2x}$ are called the **finite diference operators** for approximating **first-order derivatives** and their expansion is called the **finite difference quotient**, each representing

[1] An object which consists of points which are spaced in a specific geometrical pattern is referred to as a **mesh** and each point in this mesh is called a **node**. The distance between any two adjacent nodes in a mesh with uniform spacing is called its **meshsize**

forward,backward and centered respectively. Second and Higher order finite difference Quotients can also be obtained,

$$\begin{aligned}
U_{xx}|_i &= \lim_{h\to 0} \frac{U_x(x_i+\frac{h}{2}) - U_x(x_i-\frac{h}{2})}{h} \\
&= \lim_{h\to 0} \frac{1}{h}\left[\frac{U(x+h)-U(x)}{h} - \frac{(U(x)-U(x-h))}{h}\right] \\
&= \lim_{h\to 0} \frac{U_{i+1}-2U_i+U_{i-1}}{h^2} \\
&\approx \boxed{\delta_x^2 U_i \equiv \frac{1}{h^2}(U_{i+1}-2U_i+U_{i-1})} \qquad \text{[Central second-order Difference]}
\end{aligned}$$



The vector of coefficients of the function values at various nodes forms what is called the **stencil** of the finite difference operator and it uniquely idetifies the operator. The combination $(1,-2,1)$ is called a **three point stencil** as it combines function values from three different points on the mesh. It is fairly obvious to notice that any finite difference operator for any derivative at any node is just a linear combination of the function values at various neighbourhood nodes.

### 3.1.2 Local Truncation Error of Finite Difference Approximations

The *'error'* that accompanies *'approximations'* in the method must be acccounted for. In this section, the truncation error in the derivative approximations is ascertained which will later help us deduce the error in PDE's solved using these approximations.

**The local truncation error for derivative approximations** is defined here as the difference between the exact value of the derivate and the approximated value at node $i$, it can be calculated using Taylor series expansions about $i$,

For Forward difference operator,

$$\begin{aligned}
\tau &\equiv \delta_x^+ U_i - U_x|_i \\
&= \frac{1}{\Delta x}\left(U_{i+1}-U_i\right) - U_x|_i \\
&= \frac{1}{\Delta x}\left[\left(U_i + \Delta x U_x|_i + \frac{1}{2}\Delta x^2 U_{xx}|_i + \mathcal{O}(\Delta x^3)\right) - U_i\right] - U_x|_i \\
&= \frac{1}{2}\Delta x U_{xx}|_i + \mathcal{O}(\Delta x^2) = \mathcal{O}(\Delta x)
\end{aligned}$$

For Backward difference operator,

$$\begin{aligned}
\tau &\equiv \delta_x^- U_i - U_x|_i \\
&= \frac{1}{\Delta x}\left(U_i - U_{i-1}\right) - U_x|_i \\
&= \frac{1}{\Delta x}\left[U_i - \left(U_i - \Delta x U_x|_i + \frac{1}{2}\Delta x^2 U_{xx}|_i + \mathcal{O}(\Delta x^3)\right)\right] - U_x|_i \\
&= -\frac{1}{2}\Delta x U_{xx}|_i + \mathcal{O}(\Delta x^2) = \mathcal{O}(\Delta x)
\end{aligned}$$

For Central difference operator,

$$
\begin{aligned}
\tau &\equiv \delta_{2x} U_i - U_x|_i \\
&= \frac{1}{2\Delta x}\left(U_{i+1} - U_{i-1}\right) - U_x|_i \\
&= \frac{1}{2\Delta x}\Bigg[\left(U_i + \triangle x U_x|_i + \frac{1}{2}\triangle x^2 U_{xx}|_i + \frac{1}{6}\triangle x^3 U_{xxx}|_i + \frac{1}{12}\triangle x^4 U_{xxxx}|_i + \mathcal{O}(\triangle x^5)\right) \\
&\quad - \left(U_i - \triangle x U_x|_i + \frac{1}{2}\triangle x^2 U_{xx}|_i - \frac{1}{6}\triangle x^3 U_{xxx}|_i + \frac{1}{12}\triangle x^4 U_{xxxx}|_i + \mathcal{O}(\triangle x^5)\right)\Bigg] - U_x|_i \\
&= -\frac{1}{6}\Delta x^2 U_{xxx_i} + \mathcal{O}(\Delta x^4) = \mathcal{O}((\Delta x)^2)
\end{aligned}
$$

where in the above expressions we assume that the Higher order derivatives of $U$ at $i$ are well defined. For a fairly small $\Delta x$ (less than 1) we can confidently say that $\mathcal{O}(\Delta x^2)$ is samller than $\mathcal{O}(\Delta x)$[1]. Thus we note that the centered difference approximation (second-order accurate) approximates the derivative more accurately than either of the *one-sided diferences* which are first-order accurate.[2]

Similarly, Approximation of second-order derivative,

$$
\begin{aligned}
\tau &\equiv \delta_x^2 U_i - U_{xx}|_i \\
&= \frac{1}{(\Delta x)^2}(U_{i+1} - 2U_i + U_{i-1}) - U_{xx}|_i \\
&= \frac{1}{(\Delta x)^2}\Bigg[\left(U_i + \triangle x U_x|_i + \frac{1}{2}\triangle x^2 U_{xx}|_i + \frac{1}{6}\triangle x^3 U_{xxx}|_i + \frac{1}{12}\triangle x^4 U_{xxxx}|_i + \mathcal{O}((\triangle x)^5)\right) - 2U_i \\
&+ \left(U_i - \triangle x U_x|_i + \frac{1}{2}\triangle x^2 U_{xx}|_i - \frac{1}{6}\triangle x^3 U_{xxx}|_i + \frac{1}{12}\triangle x^4 U_{xxxx}|_i + \mathcal{O}((\triangle x)^5)\right)\Bigg] - U_{xx}|_i \\
&= \mathcal{O}((\triangle x)^2)
\end{aligned}
$$

Thus, the second-order derivative approximator is also second order accurate.

### 3.1.3 Reducing PDE to a discretised difference equation

First we decompose our continous domain $\Omega := [0,4] \times [0,4.4]$ of $U'(x,y)$ to a discretised one by overlaying it with a uniformly structured rectangular mesh of meshsize $\triangle x = \triangle y = h$ and working only on the nodes of the mesh.

Number of nodes along x axis, $N_x = \frac{4}{h} + 1$

Number of nodes along y axis, $N_y = \frac{4.4}{h} + 1$

Therefore we have, $U'_{i,j} = U'(x_i, y_j)$ and $\rho'_{i,j} = \rho'(x_i, y_j) \quad \forall \quad i \in 0, 1, \ldots, N_x - 1; \quad j \in 0, 1, \ldots, N_y - 1$.

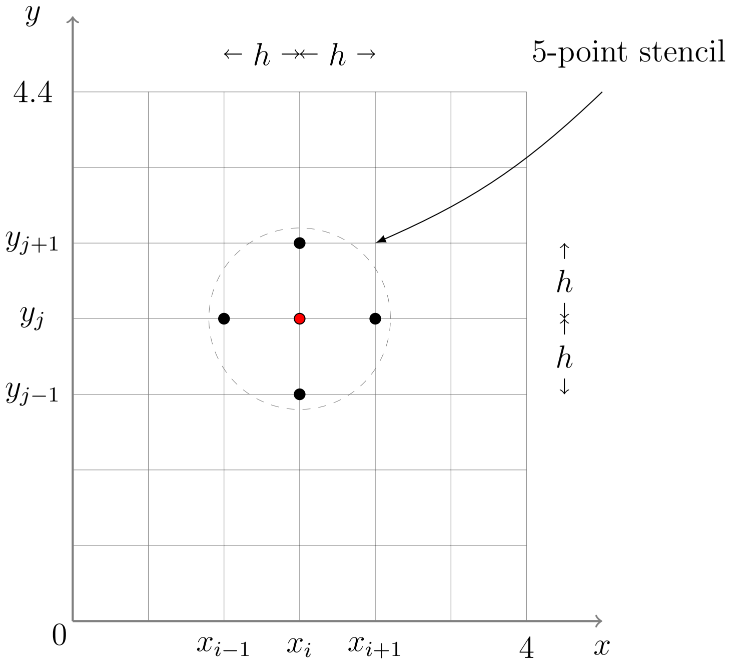
We replace the second-order derivatives in partial differential equation (2.16) with central difference operators,

$$\delta_x^2 U'_{i,j} + \delta_y^2 U'_{i,j} = -\frac{\rho'_{i,j} s^2}{\epsilon_0 \nu} \tag{3.1}$$

$$\frac{1}{h^2}(U_{i+1,j} + U_{i-1,j} - 4U_{i,j} + U_{i,j+1} + U_{i,j+1}) = -\frac{\rho'_{i,j} s^2}{\epsilon_0 \nu} \tag{3.2}$$

[1] The definition of the "big $\mathcal{O}$" notation says that if for given functions $f(x)$ and $g(x)$ for $x \in S$ where S is some subset of $\mathbf{R}$, there exists a positive constant A such that $|f(x)| \leq A|g(x)| \; \forall \; x \in S$, we say that $f(x)$ is the "big $\mathcal{O}$" of $g(x)$ or that $f(x)$ is of order of $g(x)$, mathematically given by $f(x) = \mathcal{O}(g(x))$

[2] Forward and Backward differences are also called one-sided differences



After rearranging we obtain the useful relation,

$$U_{i,j} = \frac{1}{4}\left[U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j+1} + h^2\frac{\rho'_{i,j}s^2}{\epsilon_0\nu}\right] \tag{3.3}$$

## 3.2 Iterative methods to solve linear algebraic equations

In the last section we have discussed how to reduce a PDE to a linear combination of function values at various nodes by means of the method of finite differences. If we let the function value at any node as an unknown variable then the stencil when applied to all interior nodes gives rise to a system of linear algebraic equations, which may be very large. A two-dimensional problem like ours may lead to a system of several thousand unknowns, and three-dimensional problems involving several hundred thousand unknowns are common in real engineering situations. The solution of such a system is a major problem in itself as traditional methods like Gaussian-elimination result in large computation times, we are therefore forced to employ faster methods. As we have seen above, the system of equations produced by a discretisation has many special features and an efficient solution procedure must exploit these. The most obvious property of the system is that it is extremely sparse. Even when there are many thousand unknowns, each equation will involve one unknown and the unknowns at its immediate neighbourhood. In particular, if we write the equations in the conventional notation,

$$A\vec{x} = \vec{b}$$

where A is an N × N matrix, b the given data vector and x the vector of N unknown interior mesh values, there is an implied one-dimensional ordering of these values which is somewhat unnatural and obscures the important property that only immediate neighbours are involved. Each row of the matrix A involves only a very small number of non-zero elements, commonly five or seven; moreover in many problems a suitable ordering of the unknowns will lead to a matrix in which these non-zero elements occur in a regular pattern. In devising efficient methods for the solution of the system these structural properties will be important,

### 3.2.1 Jacobi Method

### 3.2.2 Gauss-Seidel Method

### 3.2.3 Relaxation methods and Successive Over-Relaxation(SOR)

```
Data: INPUT -: An $k \times m$ matrix of initial values, value of step size $h$ and also a matrix
      containing the initial charge configuration
Result: OUTPUT -: An $k \times m$ matrix containing the values of potential on all $x$ and $y$ values
for $f$ = 0, 1, 2, 3, 4....N do
    make new array of size $k \times m$ ;          /* initialising a new array for solution */
    for $i$ =0, 1, 2, 3 ... $k$ do ;                                  /* taking a x value */

        for  $j$ =1, 2, 3 ... (m-1) do ;             /* taking all y value for a x value */

            /* now defining the required quantites for stencil                          */
            $left = a_{i,j-1}$ ;
            $right = a_{i,j+1}$ ;
            if $i = k - 1$ then
                $up = a_{i-1,j}$;
                else
                    up = $a_{i+1,j}$;
                end
            end
            if  $i = 0$ then
                down = $a_{i+1,j}$;
                else
                    down = $a_{i-1,j}$ ;
                end
            end
            new $a_{i,j}$ = (up + down + left + right + $h^2 * p_{i,j}$ )/4 + new $a_{i,j}$ ;  /* new value the
            grid point */
        end
        max relative error = max(new x - x)/ $x$;
        if max relative error < tolerance then ;                 /* checking for tolerance */

            break;
            else ;       /* if tolerance not reached then the iteration continues */

                new x = x
            end
        end
    end
end
```

**Algorithm 1:** Jacobi Method

# 4 Numerical Analysis

In this section we will try to analyse results thoroughly. In this section we will analyse the Jacobi Method, SOR and Gauss Seidel Method

## 4.1 Main Problem

our problem consists of 4 capacitor which were charged at the beginning and then disconnnected after they were charged to a potential of $\pm 5volts$. So in our problem the first plate of capacitor at the begining (i.e. $A_1$ in figure 1) of system and last plate (i.e. $A_5$) of the last capacitor would act as the boundary condition. We are solving our problem by considering it only in two direction and not taking the third direction due to the symmetry of problem.

### 4.1.1 Expectation

Now since in our problem there are 4 capacitor and all of them have identical condition in the begining so we expect to get symmetrical result with high potential near the positively chacged plates (i.e. the plates denoted by A in the figure1) of capacitor and low potential near the negatively charged plates but we are considerig them in two dimension so they will behave as the "line charges " rather than the capacitors.Since we are ensuring the value at the boundary reamins constant at $+5volts$ so the distribution should contain the positive spike in value of potential at the location of " line charges " having positive potential at the begining and negative spike at places of negative potential.

Now if we anlayse the our set up from ones side then, at the top of arrangement we have boundary in form of line charge whose potential is 5 volts and then we line charge $B_1$ which was charged to a potential of $-5volts$ so we should expect a decrease in potential as we move from $A_1$ to $B_1$ in x direction with spike at $B_1$ and since the potential is not maintained in Y direction so we should expect a decrease in potential from one side of Y to other, now after $B_1$ we move to $A_2$ so we should expect increase in potential with positive spike at $A_2$ in x direction and since the $A_2$ line charge extend till only $4\mu m$ and its predecessor and successor which are both negatively charged exist for $4.4\mu m$ so we should expect to see aggragation of negative charge for higher values of Y so low potential for higher values of Y and low value potential for values of Y closer to zero. Now if we move $A_2$ to $B_2$ the potential should decrease from $A_2$ to $B_3$ in x direction and the variation in potential should be similar to previous case.

Nwo due to symmetry, we can just divide the region as going from positve to negative or negative to positive line charge and then it's variation of potential can be explained as $A_2$ to $B_2$ or $B_1$ to $A_2$ respectively.

So our expected solution should be similar to figure below -:

### 4.1.2 Results

In this section we will analyse the computational results obatined using three different iterative methods after converting the problem into system of linear equation by using the method of finite differences.

**Jacobi Method** The results obtaied from this method agree with the expectation of form of solution. The solution obtained from this method is shown below in form of heat map and surfaceplot.

As we can see from the gaph that there is gradual decrease in value of potential in x direction till we reach in middle of x and then it again increases till the end of x direction which is as expcted. This runtime for this method is $\Bigg($ ) for a mesh size of with a step size of afer running for number of iterations which is not bad for iterative method running in python.

**Gauss Seidel Mehtod** The result obtained from this method are shown in below figure

After looking at the heat map can say that the results are in agreement with our expectation i.e. there is more negative potential for higher values of Y and decrease in potential in X direction til middle and then increase til boundary. The number of iterations for acheiving the tolearnce of *dashdash* is

**Successive Over Relaxation (SOR)** In this section we will first analyse the results obtained for different values of relaxation factor. Then we will analyse the results obtained for the chosen relaxation factor.

**Optimum value of Relaxation Factor** In Successive Over Relaxation we have to choose the value of a relaxation factor which is responsible for the rate of convergence of solution. It's value can be chosen anywhere between 1 to 2 i.e. relaxation factor or $\omega \in [1, 2]$. So we used different values of $\omega$ to see which one is best for computation in our case by varying the value of $\omega$ between 1 and 2 with a step size of 0.01 and stored the value of number of iterations required to reach the tolerance. The following graph represents the graph between number of itereation and value of $\omega$.

**Result** From the graph we can see that the most optimum value of $\omega$ is 1.89 according to number of iterations required to reach the solution.Now we will analyse the plot, heat map and compare it with the expectation. The below figure shows the heat map and the 3d plot of solution.
Now first analysing the y direction of graph, in genral we expected to see negative potential for higher values of Y and positive potential for lower values of Y which is exactly what is happening in the heat map. For X direction we expected to see increase or decrease in potential as depending upon going from a postively charged plate to negatively charged plate or vice-versa and overall change to be first decreasing till middle and then again increasing due to the fixed values of potential at boundary which is also agreeing with the heat map of solution.

**Comparison of Method** Since all the iterative methods we used have given us almost identical results so we will only compare them on the basis of number of iteration required so as to determine which method is more compuationally cheap and viable

**Jacobi and Gauss Seidel Method** Jacobi Method required *dashdash* number of iteration for giving us a result with a tolearnce of *dashdash* while Gauss Seidel Method required *dashdash* number of iteration for acheiving the same tolerance. Hence we can say that Gauss Method is better choice than Jacobi Method compuatationally.

**SOR and Jacobi** When we compared different values of relaxation factor for finding the optimum value of relaxation factor we found that for $\omega =$ , the number of iteration it required was only dash dash in comparison of Jacobi Method so clearly the SOR method is more computationally cheap and hence naturally better option as compared to Jacobi Method.

**SOR and Gauss Seidel Method** Both of these iterative schemes are dependent on $\omega$ while for Gauss Seidel $\omega$ is always 1 , the value of $\omega$ can vary from $(1, 2)$ after performing the computation for a mesh of same size and same step size, the number of iteration for SOR were and for Gauss Seidel were . So here also SOR is more computationally cheap and better option as compared to its counterpart.

So comparing all the method for computation under identical condition on the basis of number of iteration required for acheiving the required tolerance we can say $\boldsymbol{SORMethod}$ is best method among all three iterative schemes used.

**Numba Library** The function for all the iterative method were run under a *jitcover* of *numbalibrary* which when used in python converts the function of python into machine code which is more faster for performing computation in python.

# 5 Conclusion

## 5.1 Result

We tried to solve the problem of interleaved capacitor using the finite difference method for solving poisson equation. We have solved the problem using three different iterative schemes. After completing this project we can say that the method of SOR is best for solving the system of linear equation after using the finite differences method. Using SOR method we were able to solve mesh of size $(480times480)$ in just $xxxxxx$ number of iteration in just $yyyy$ seconds. Also we gained good insight and intuition after solving the problem of interleaved capacitor.

## 5.2 Experience

We learnt a lot of new things during this project. We have gained the knowledge on how to solve a physical problem computationally and the various process involved in it such as non-dimensionalisation, the concept of convergence etc. Our python, latex and gnu skills have also increase significantly and our fascination with power of poisson equation and computation method have only gone uphill as compared to start of project of how one can solve such complex physical problem just by using some standard method and understand the "physical aspect " of such problems easily. We also spend a good portion of time studying about theoretical aspects of different computational methods and trying to understand the concepts about which we didn't pay a lot of attention to earlier such as the truncation error , round off error etc. This project has been an incredible journey for us as it has not only increased our theoretical , physical and compuational knowledge but it has also taught us about the importance of perseverance and patience as there were many topic or subtopics that we didn't understand easily just by studying about it from one or two place or things that were not easily available in comprehendable nature for us due to advance nature of partial differential equation and sometimes we have to spend a lot of time just searching about it. But in the end we are very grateful that we had chosen such a topic that has taught us so much.