

Contents

1	Introduction	2
1.1	Motvation	2
1.2	General Idea	2
2	Theory	2
3	Methodology	4
3.1	Finite Difference Method	4
3.1.1	Finite Difference Approximations	4
3.1.2	Local Truncation Error of Finite Difference Approximations	5
3.1.3	Reducing PDE to a discretised difference equation	7
3.2	Iterative methods to solve linear algebraic equations	7
3.2.1	Jacobi Method	8
3.2.2	Gauss-Seidel Method	8
3.2.3	Relaxation methods and Successive Over-Relaxation(SOR)	8
4	Analysis of Numerical Results	8
5	Appendix	8
5.1	Non-Dimensionalization	8
5.2	Programs	8
5.3	Contributions	8

1 Introduction

1.1 Motivation

We first encountered Laplace Equation during our course in electricity and magnetism in second semester and we were fascinated with how one can calculate 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 was able to solve in region having charges (or sources). When we were given the opportunity to choose a project in our computational physics charge 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 equation which is an elliptic 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

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 depicted 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 third axis which will be dropped) inside the capacitor after we disconnect the capacitor from sources, treating the interior plates as line charge distributions.

$$U_A = 5V$$

$$U_B = -5V$$

$$\text{Capacitance} = 0.1\mu F$$

$$\text{Distance between plates } (d) = 0.5\mu m$$

$$\text{Dimensions: } 4 \times 4.4\mu m$$

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,

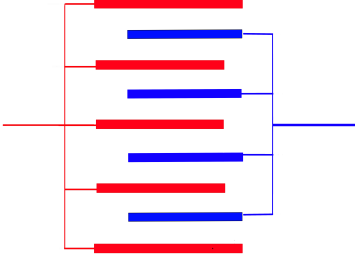
$$C_0 = \epsilon_0 \frac{A}{d} \implies A = \frac{C_0 d}{\epsilon_0} \quad (2.1)$$

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

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

Similarly on B_i ,

$$\rho_B = -2\epsilon_0 \times 10^5 C m^{-2} \quad (2.3)$$



(a) Diagram depicting the arrangement of plates in an 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,

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

where,

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

Now according to our given arrangement,

$$B_1 = \{(x^*, y^*) : x^* = 0.5\mu m ; 0.4\mu m \leq y^* \leq 4.4\mu m\} \quad (2.6)$$

$$B_2 = \{(x^*, y^*) : x^* = 1.5\mu m ; 0.4\mu m \leq y^* \leq 4.4\mu m\} \quad (2.7)$$

$$B_3 = \{(x^*, y^*) : x^* = 2.5\mu m ; 0.4\mu m \leq y^* \leq 4.4\mu m\} \quad (2.8)$$

$$B_4 = \{(x^*, y^*) : x^* = 3.5\mu m ; 0.4\mu m \leq y^* \leq 4.4\mu m\} \quad (2.9)$$

$$A_2 = \{(x^*, y^*) : x^* = 1\mu m ; 0\mu m \leq y^* \leq 4\mu m\} \quad (2.10)$$

$$A_3 = \{(x^*, y^*) : x^* = 2\mu m ; 0\mu m \leq y^* \leq 4\mu m\} \quad (2.11)$$

$$A_4 = \{(x^*, y^*) : x^* = 3\mu m ; 0\mu m \leq y^* \leq 4\mu m\} \quad (2.12)$$

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

$$U(0, y) = +5V ; U(4, y) = +5V \quad (2.13)$$

$$U_y(x, 0) = 0V/m ; U_y(x, 4.4) = 0V/m \quad (2.14)$$

Non-Dimensionalizing the variables,

Let new dimensionless variables,

$$x' = \frac{x}{s} \quad y' = \frac{y}{s} \quad U' = \frac{U}{\nu} \quad (2.15)$$

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

The B.V.P reduces to,

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

where,

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

Now according to our given arrangement,

$$B_1 = \{(x^*, y^*) : x^* = 0.5/s ; 0.4/s \leq y^* \leq 4.4/s\} \quad (2.18)$$

$$B_2 = \{(x^*, y^*) : x^* = 1.5/s ; 0.4/s \leq y^* \leq 4.4/s\} \quad (2.19)$$

$$B_3 = \{(x^*, y^*) : x^* = 2.5/s ; 0.4/s \leq y^* \leq 4.4/s\} \quad (2.20)$$

$$B_4 = \{(x^*, y^*) : x^* = 3.5/s ; 0.4/s \leq y^* \leq 4.4/s\} \quad (2.21)$$

$$A_2 = \{(x^*, y^*) : x^* = 1/s ; 0/s \leq y^* \leq 4/s\} \quad (2.22)$$

$$A_3 = \{(x^*, y^*) : x^* = 2/s ; 0/s \leq y^* \leq 4/s\} \quad (2.23)$$

$$A_4 = \{(x^*, y^*) : x^* = 3/s ; 0/s \leq y^* \leq 4/s\} \quad (2.24)$$

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

$$U(0, y) = +5/\nu ; U(4, y) = +5/\nu \quad (2.25)$$

$$U_y(x, 0) = 0 ; U_y(x, 4.4) = 0 \quad (2.26)$$

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 continuous 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 $(x_0, x_1, x_2, \dots, x_i, \dots, x_n)$ 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 $x_i \equiv x_0 + ih$.

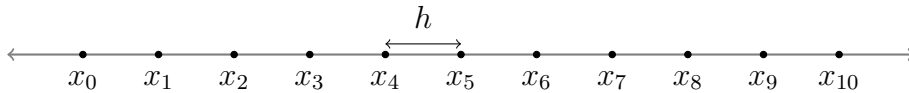
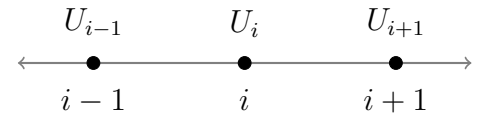


Figure 2: 1D mesh with 11 nodes and a meshsize h



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

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

¹An 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**

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

The first order derivative can be defined as,

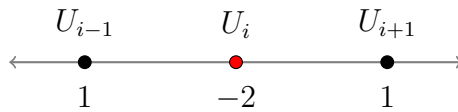
$$\begin{aligned} U_x|_i &= \lim_{h \rightarrow 0} \frac{U_{i+1} - U_i}{h} \\ \text{or, } U_x|_i &= \lim_{h \rightarrow 0} \frac{U_i - U_{i-1}}{h} \\ \text{or, } U_x|_i &= \lim_{h \rightarrow 0} \frac{U_{i+1} - U_{i-1}}{2h} \end{aligned}$$

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

$$\begin{aligned} \text{Forward Difference} \quad U_x|_i &\approx \frac{U_{i+1} - U_i}{h} \equiv \delta_x^+ U_i \\ \text{Backward Difference} \quad U_x|_i &\approx \frac{U_i - U_{i-1}}{h} \equiv \delta_x^- U_i \\ \text{Central Difference} \quad U_x|_i &\approx \frac{U_{i+1} - U_{i-1}}{2h} \equiv \delta_{2x} U_i \end{aligned}$$

Where $\delta_x^+, \delta_x^-, \delta_{2x}$ are called the **finite difference operators** for approximating **first-order derivatives** and their expansion is called the **finite difference quotient**, each representing forward, backward and centered respectively. Second and Higher order finite difference Quotients can also be obtained,

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



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 identifies 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 accounted 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 derivative and the approximated value at node i , it can be calculated using Taylor series expansions about i ,

For Forward difference operator,

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

For Backward difference operator,

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

For Central difference operator,

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

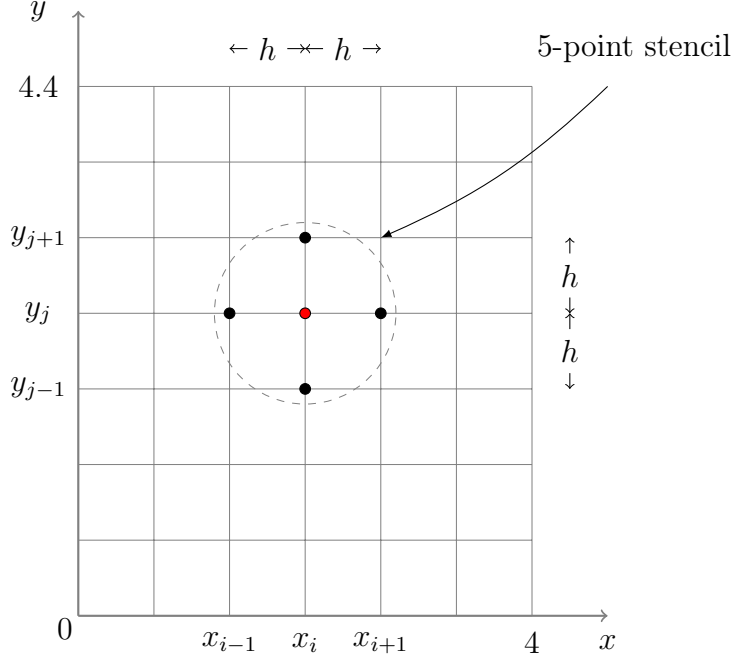
where in the above expressions we assume that the Higher order derivatives of U at i are well defined. For a fairly small Δx (less than 1) we can confidently say that $\mathcal{O}(\Delta x^2)$ is smaller than $\mathcal{O}(\Delta x)^1$. Thus we note that the centered difference approximation (second-order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

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

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

¹The 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 $f(x) = \mathcal{O}(g(x))$

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



3.1.3 Reducing PDE to a discretised difference equation

First we decompose our continuous 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 $\Delta x = \Delta y = h$ 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 $\rho'_{i,j} = \rho'(x_i, y_j) \quad \forall \quad i \in 0, 1, \dots, N_x - 1; \quad j \in 0, 1, \dots, N_y - 1$.

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

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

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

After rearranging we obtain the useful relation,

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

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 $N \times N$ 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)

4 Analysis of Numerical Results

5 Appendix

5.1 Non-Dimensionalization

5.2 Programs

5.3 Contributions

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 $f = 0, 1, 2, 3, 4 \dots N$ **do**

 make new array of size $k \times m$; */* initialising a new array for solution */*

for $i = 0, 1, 2, 3 \dots k$ **do** ; */* taking a x value */*

for $j = 1, 2, 3 \dots (m-1)$ **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