Monte Carlo based 2D CAD Tool for Calculating Capacitance of a User-Parameterized Geometry and Validated against a Golden Reference

Project report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

Submitted by

Bhavith Dulipalla (1810110053)
and
S.Bhanoday (1810110194)

Under Supervision of

Dr.Venkatnarayan Hariharan
Associate Professor, Department of Electrical Engineering

Department of Electrical Engineering
School of Engineering
Shiv Nadar University

(Spring 2022)

Candidate Declaration

We hereby declare that the thesis entitled "Monte Carlo based 2D CAD Tool for Calculating Capacitance of a User-Parameterized Geometry and Validated against a Golden Reference" submitted for the B. Tech. degree program. This thesis has been written in our own words. We have adequately cited and referenced the original sources.

(Signature)
Bhavith Dulipalla
(1810110053)
Date: 28/04/2022

(Signature)
S.Bhanoday
(1810110194)
Date: 28/04/2022

Certificate

It is certified that the work contained in the project report titled "Monte Carlo based 2D CAD Tool for Calculating Capacitance of a User-Parameterized Geometry and Validated against a Golden Reference", by Bhavith Dulipalla and S.Bhanoday have been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Signature)
Dr.Venkatnarayan Hariharan
Dept. of Electrical Engineering
School of Engineering
Shiv Nadar University
Date: 28/04/2022

Abstract

We have developed a 2-dimensional CAD tool for accurately calculating the capacitance of a subset of user-specifiable arbitrary geometries, taking fringe capacitance into consideration. Its key advantage is that it is much faster than a regular electrostatic field-solver. We also validate its results by comparing with a commercial field-solver as a golden reference. In this report, we describe the capabilities of our tool, explain the methodology and the validation, and finally present the results.

This report starts off by understanding the importance of capacitance extraction followed by discussion of various numerical methods such as Boundary Element Method, Finite Element Method, Finite Difference Method and Monte Carlo Methods which are capable of extracting capacitance's, a brief literature survey is done for various Monte Carlo Methods such as Floating Random Walk Algorithm, Fixed Random Walk Algorithm and Exodus Method.

The later part of report focuses on extracting the capacitance between two rectangular Plates using the Monte Carlo Simulation Methods (Floating Random Walk Algorithm) and validating those results with a field solver [11]. Floating Random Walk Algorithm was implemented in Python programming (version 3.9.1). Sentaurus Visual is used to visualise Heat Maps. At the end of this project, a good understanding of theoretical and practical knowledge of Floating Random Walk Algorithm and a good hands-on experience of using the Raphael tool is obtained.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Importance of Capacitance Extraction

In this High-speed Era, all things around us are Digital. Recent trend is focused on development of the smallest and powerful computational Digital ICs. Modem digital ICs can contain millions of interconnects. Performance and efficiency of these digital ICs depends on accurately estimating and minimizing the parasitic capacitances amongst the various on-chip interconnects in a VLSI chip. An accurate estimation of these capacitances (referred to as parasitic extraction in the industry) helps improve the accuracy of the simulation of the gate- and transistor-level designs of sub-blocks of these ICs. Capacitance of simple geometries can be calculated analytically, but capacitance of arbitrary geometries can only be numerically calculated.

1.2 Laplace Equation

1.2.1 Importance of Laplace equation

One of the method for obtaining the capacitance characteristics for a particular cross-sectional geometry involves solving the Laplace's equation,

$$\nabla[\epsilon(x,y)\nabla V(x,y)] = 0 \qquad (1)$$

where the electric potential V(x,y) is prescribed on the boundary of a given domain. Traditionally, this problem has been solved deterministically, either analytically for simple geometries exhibiting spatial symmetry or numerically for more complicated geometries. Due to the complexity of typical integrated circuit cross-sections numerical techniques explained below are used instead of deterministic methods.

1.2.2 Role of Laplace equation in Capacitance Extraction

We have expressions for electric field and electric potential as integrals over the charge density ρ. These are not trivial problems especially for arbitrary geometries and/or where fringing fields play a significant role. So we shall find it useful to work with the differential equations for electric field and electric potential especially if we are interested in a solution in a limited region of space.

Laplace equation given below is used to solve the potential i.e.,

$$\nabla^2 V = 0 \qquad (2)$$

where V is the electric potential, related to the electric field by

$$E = -\nabla V \qquad (3)$$

Once electric potential is obtained from Laplace equation, we can find capacitance from it by calculating electric field and charge. The process mentioned here is clearly explained in Work Done section

1.3 Boundary Conditions

There are usually some types of boundary conditions which control the value of the solution on the boundary of the domain. The two most basic forms of boundary condition are called the Dirichlet boundary and the Neumann boundary.

1.3.1 Dirichlet Boundary Condition

Dirichlet boundary condition specifies the value that the unknown function needs to take on along the boundary of the domain.

Given, for example, the Laplace equation, the boundary value problem with the Dirichlet boundary condition is written as:

$$\Delta^2 \psi(x) = 0 \qquad \forall x \in \Omega \qquad (4)$$

$$\psi(x) = f(x) \qquad \forall x \in \partial\Omega \qquad (5)$$

where ψ is the unknown function, x is the independent variable (e.g. the spatial coordinates), Ω is the function domain, $\partial\Omega$ is the boundary of the domain, and f is a given scalar function defined on $\partial\Omega$.

1.3.2 Neumann Boundary Condition

The Neumann boundary condition exists when the derivative of the potential function is known. Generally speaking, the derivative is defined with respect to the outward unit normal at the boundary, which is written as:

$$\frac{\partial V(r)}{\partial n} = f'(x) \qquad r \in \Omega \qquad (6)$$

where n is the outward-pointing unit normal vector, and f' specifies the set of known derivatives. Unlike the Dirichlet condition, the Neumann does not offer a direct solution to the voltage potentials on a discrete, sampled grid. Rather, the boundary point must be expressed in terms of the surrounding points.

1.4 Different Methods For Capacitance Extractions

There are various numerical techniques like Boundary Element Method (BEM), Finite Element Method (FEM), Finite Difference Method (FDM), Monte Carlo Method (MCM), etc. which can compute these capacitance's at different levels of accuracy.

1.4.1 Boundary Element Method (BEM)

The BEM [1] relates the potential at the conductor boundaries to the charge distribution at those boundaries, BEM discretizes boundary and volume into elements, and writes electrostatic equations at each element. The resulting linear equations are solved by efficient techniques. Boundary element based field solvers can effectively be used for small structures but can not be used for large structures because of large grid requirements.

1.4.2 Finite Element Method (FEM)

The Finite element method (FEM) [2] is a numerical method for solving a differential or integral equation. In this technique the solution of Laplace's equation is approximated by a simple function over each element. The global solution is then obtained by combining the solutions for the many elements. If the solution of the potential equation is required at only one point (or some limited number of points as in a capacitance calculation), then a considerable amount of time is wasted in calculating potentials at points that are of little or no interest.

1.4.3 Finite Difference Method (FDM)

The Finite Difference Method (FDM) [2] has been widely used to solve variable electromagnetic problems, especially the extraction of distributed parameters. In the FDM technique Laplace's equation is replaced by its difference equation equivalent and then either direct (matrix inversion) or iterative (relaxation) methods are employed to solve for the potential.

Domain Discretization Method and Boundary Element Method (BEM) are two major methods for the field solver, especially useful for building capacitance library for LPE tools. However, its accuracy is sensitive to the Boundary Discretization. Compared with BEM, the Domain Discretization in FDM is much easier so that the FDM based capacitance solver is robust on accuracy and widely used in industry.

The Finite Difference techniques are based upon approximations which permit replacing differential equations by finite difference equations. These finite difference approximations are algebraic in form; they relate the value of the dependent variable at a point in the solution region to the values at some neighboring points.

Thus a Finite Difference solution procedure of Poisson's or Laplace's equations involves three steps:

1. Dividing the domain of interest (in which the potential is to be determined) into a suitable fine grid. Instead of a solution for ϕ(x,y), which provides its continuous

variation for a given charge distribution ρ(x,y), the FD solution will provide discrete values of ϕ at the "nodes" of the established grid.

2. Applying the difference equation at each node of the grid to obtain, for example, N equations in the N unknown node potentials.

3. Solving the difference equations subject to the prescribed boundary conditions and/or initial conditions.

The drawback of deterministic techniques such as FEM and FDM is that in order to find the potential at some local point, the solution must be obtained globally. This is due to the very nature of deterministic numerical techniques in that the entire domain of interest is discretized: a mesh for the FDM technique and sub-regions (elements) for the FEM technique.

1.4.4 Monte Carlo Method (MCM)

Monte Carlo method [4] is defined as a simulation process that generates probabilities using a mathematical model. The method provides a range of possible results based on the varying parameters that are measured in the analysis. This method is about generating situational results based on distributing factors that may influence the outcome of the process. It takes into account the maximum and minimum threshold of each parameter and randomly iterates the simulation with different values.

It is the one of the efficient methods for extraction of capacitance produced in interconnects of the VLSI chip or IC. Monte Carlo method computes capacitance by evaluating the charge integral by statistical sampling techniques.

1.5 Advantages of MCM over other methods

Monte Carlo method has the following advantages if compared with other deterministic methods:

1. Locality - It allows us to obtain the solution at a few local positions without solving the whole equation.

2. Stability of accuracy - Its accuracy is more stable and tunable, as its error is mainly stochastic error.

3. Geometric adaptability - Without geometry discretization, it reduces memory and avoids preprocessing.

4. Suitability for large problems - Without the generation of a linear equation system, it works for large problems.

5. Execution time - The Monte Carlo techniques takes less time to run than deterministic methods, especially for large problems (as encountered in dense interconnect meshes on a VLSI chip).

1.6 Fringing Fields

In the traditional analytical models, Fringing fields (curved field lines) from all edges are neglected and assumed that field lines are exactly parallel and straight as shown in the Fig 1.1.

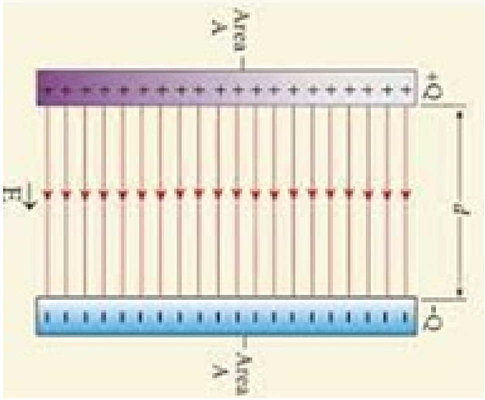


Figure 1.1: Electric field lines when fringing fields are neglected

In this case the capacitance value of the parallel plate capacitor is given by

$$C = \epsilon_o \frac{A}{d} \qquad (7)$$

But in our model we take the Fringing effect into consideration as shown in the Fig 1.2.

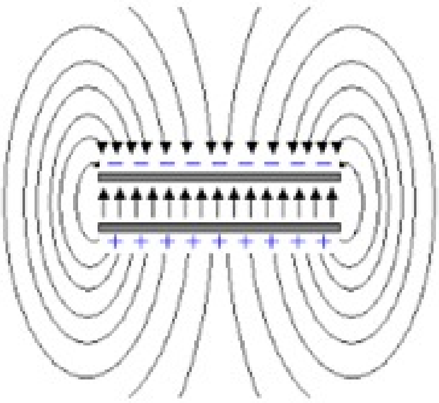


Figure 1.2: Electric field lines when fringing fields are not neglected

Fringing is the bending of the electric field lines near the edge of the parallel plate capacitors. Fringing is also known as " **Edge Effect** ". Normally the field lines inside the capacitor are uniform and parallel. But at the edges, the field lines are not straight and bend slightly upward due to the geometry. This is known as the Fringing effect and is dominant when the size of the plates are comparable to other dimensions such as space separating them, a common situation in VLSI interconnects . Monte Carlo methods takes the Fringing field into consideration when calculating the capacitance value.

So, considering all the above advantages we focused on Monte Carlo methods for capacitance extraction in our project.

1.7 Raphael Field Solver:

Raphael field solver [8] is the gold standard, 2D and 3D resistance, capacitance, and inductance extraction tool for optimizing on-chip parasitic for multi-level interconnect structures in small cells. As a reference field solver, Raphael provides the most accurate parasitic models in the industry. Trusted by major foundries, interconnect parasitics generated by Raphael are included as part of their design reference guides.

1.8 Benefits of Raphael:

1. Analyze complex on-chip interconnect structures and the influence of process variation.

2. Create a parasitic database for both foundries and designers to study the effect of design rule change.

3. Generate accurate capacitance rules for layout parameter extraction tools.

4. Interface with Sentaurus Structure Editor to create and analyze arbitrary and complex 3D shapes using standard CAD operations or process emulation steps.

5. Visualize output characteristics like potential distribution inside complex 3D shapes with state-of-the-art Sentaurus Visual tools.

Chapter 2

Literature Survey

Different types of random walk lead to different Monte Carlo methods. The most popular types are floating random walk, Fixed random walk and Exodus method.

2.1 Floating Random Walk Algorithm:

The extraction of the electrostatic capacitance is a well known problem that involves the solution of Laplace's equation for the electrical potential. Among the possible techniques for solving Laplace's equation, statistical methods based on the Floating Random Walk are particularly promising since they can efficiently handle very complex geometries.

The mathematical basis of the Floating Random Walk method [3],[4] is the **Mean Value Theorem** [12] of potential theory. For example let's consider a circle of radius r, centered at (x, y) , which lies wholly within region R. When the potential varies in two dimensions, V(x, y) is given by

$$V(x,y) = \frac{1}{2\pi r} \oint V(\rho')dl' \qquad (8)$$

It can be shown that the above equation follows from Laplace's equation. Also, Equation(8) can be written as

$$V(x,y) = \frac{1}{2\pi} \int_0^1 V(r,\phi)d\phi \qquad (9)$$

where ϕ represents the polar angle of a circle.

The Floating random walk method depends on the application of (8) and (9) in a statistical sense. Suppose that a random walking particle is at some point (x_j, y_j) after j steps in the i^{th} walk. The next $(j+1)^{th}$ step is taken as follows.

First, a circle is constructed with center at (x_j, y_j) and radius p_j which is equal to the shortest distance between (x_j, y_j) and the boundary. We ***randomly*** choose a polar angle ϕ that is uniformly distributed over (0, 2π), i.e., $\phi = 2\pi$r where 0<r<1. This angle specifies a new point (X_{j+1}, Y_{j+1}) on the circle circumference as shown in the Fig 2.1.

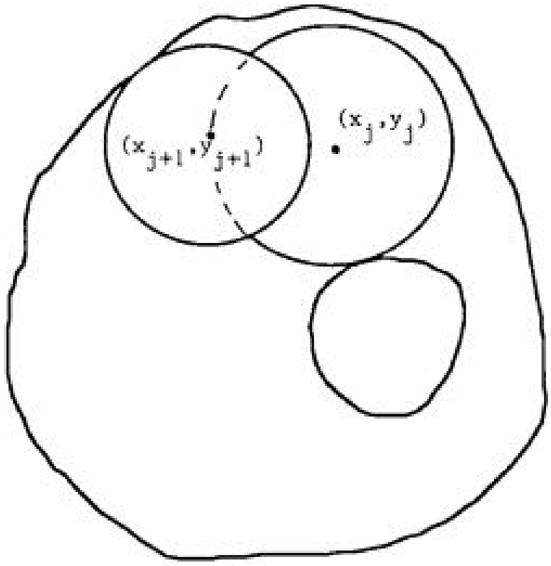


Figure 2.1: Configuration for Floating Random Walk

The next random walk is executed by constructing a circle centered at (X_{j+1}, Y_{j+1}) and of radius P_{j+1} which is the shortest distance between (X_{j+1}, Y_{j+1}) and the boundary as shown in Fig 2.1.

This procedure is repeated several times and the walk is terminated when the walk approaches some prescribed small distance δ of the boundary. Finally the potential $V_p(i)$ at the end of this i^{th} walk is recorded as in fixed random walk and the potential at (x, y) is eventually determined after N walks using

$$V(x,y) = \frac{1}{N}\sum_{i=1}^{N} V_p(i) \qquad (10)$$

where N, the total number of walks, is large.

It is evident that in the floating random walk MCM, neither the step sizes nor the directions of the walk are fixed in advance. The quantities may be regarded as “Floating,” and hence the name Floating Random Walk.

2.2 Fixed Random Walk Algorithm

There are some similarities and as well as some contrasts between Floating and Fixed Random Walk algorithms [6]. For example, consider a square/rectangular grid which has been divided into meshes of the same size as Δ shown in the Fig 2.2. To calculate the potential at (x, y) a random walking particle starts at that point. The particle proceeds to wander from node to node in the grid until it reaches the boundary. When it does, the walk is terminated and the prescribed potential V_p at that boundary point is recorded. Let the value of V_p at the end of the first walk is denoted by $V_p(1)$, as illustrated in Fig 2.2.

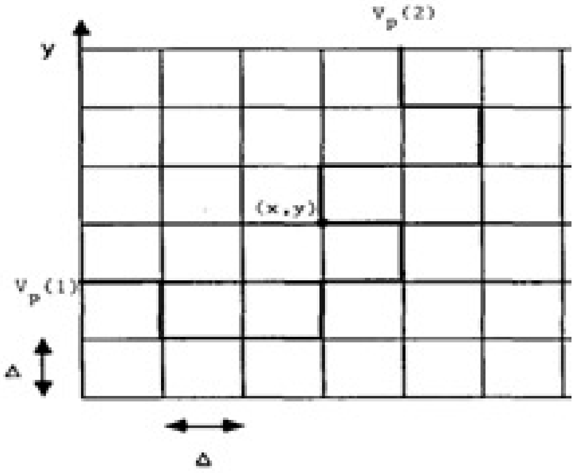


Figure 2.2: Configuration for Fixed Random Walk

Then a second particle is released from (x, y) and allowed to wander until it reaches a boundary point where the walk is terminated and the corresponding value of V_p, is denoted as $V_p(2)$. This procedure is repeated for the third, fourth, - - - , and Nth particle released from (x, y) and the corresponding prescribed potential $V_p(3)$, $V_p(4)$, ... , V_p(N) are noted. The expected value of V_p(l), $V_p(2)$, ..., V_p(N) is the solution of the potential at (x, y) i.e.,

$$V(x, y) = \frac{1}{N} \sum_{i=1}^{N} V_p(i) \qquad (11)$$

where N, the total number of walks, is large.

One of the main disadvantages of this method compared to the Floating Random Walk Method discussed above is that it is slow, and is therefore recommended for solving problems for which only a few potentials are required.

2.3 Exodus Method

The above discussed two methods had a common feature i.e., both use random numbers, whereas the Exodus Method doesn't require random numbers. To apply the Exodus Method [7] in finding the solution of a potential problem usually involves the following three steps:

1) We first obtain the random walk probabilities from the finite difference equivalent of the partial differential equation describing the problem.

2) The Exodus Method is used along with the random walk probabilities in calculating

the transition probabilities.

3) The potential at the point of interest is finally obtained using the transition probabilities and the boundary conditions.

One of the disadvantages of the Exodus Method is that it only permits calculating the potential at one point at a time. Another major disadvantage is that it is more difficult to program because two maps of the nodes are required, one just prior to the walks, and one just after the walks.

2.4 Advantages of FRWA over other methods

1. It is evident that in the Floating Random Walk, neither the step sizes nor the directions of the walk are fixed in advance like we do in the Fixed Random Walk Method.

2. A Floating Random Walk bypasses many intermediate steps of a Fixed Random Walk in favor of a long jump.

3. Fewer steps are needed to reach the boundary, and so computation is much more rapid than in Fixed Random walk.

Taking all these factors into consideration, we've decided that implementation of the Floating Random Walk algorithm suits better for this project.

2.5 Raphael Field Solver:

Raphael [9] provides different solvers for different simulation domains

1. Two-dimensional electrical parameters are best analyzed using RC2 and RC2-BEM.

2. RC3, RC3-BEM, and RCV solve 3D thermal and electrical problems.

3. The calculation of inductance's in 3D structures formed by bars and planes can be analyzed using RI3.

4. RCX analyzes capacitance, resistance, and thermal resistance.

For this project, RC2 field solver is used to calculate the capacitance of two-dimensional geometries.

2.5.1 RC2 Field Solver:

The RC2 solver [10] can be used for the entire range of 2D problems. RC2 interprets all structures as infinitely long in the direction normal to the simulation domain. Resistance, Capacitance, and Inductance are presented as the corresponding electrical units per unit length. For example, capacitance is given as farad/meter.

The combination of the automatic adjustment of mesh and the speed of the linear equation solvers makes RC2 versatile and easy to use. RC2 uses dynamic memory allocation, so the complexity of the structure to be analyzed is limited only by the available memory.

RC2 solves the Laplace equation (section 1.2) and the Poisson equation using the Finite Difference Method (section 1.4.3), with automatic gridding and regridding .

Chapter 3

Work Done

3.1 Implementing FRWA for Parallel Plate Capacitor

As discussed in the literature survey, we'll use the same procedure of Floating Random Walk Algorithm to calculate the potential and then capacitance value of parallel plate capacitor.

Consider a parallel plate as shown in the Fig 3.1. We assume a homogeneous medium with dielectric constant ϵ_o. The upper plate is given a voltage of 1V and lower plate is given a voltage of 0V.

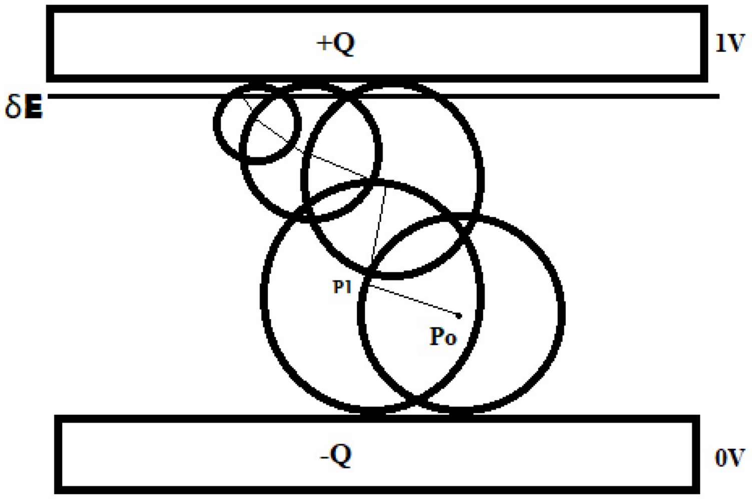


Figure 3.1: Floating random walks on parallel plate capacitor

We will calculate the potential at some point p_o. As discussed earlier, we take point p_o as the center and draw the largest possible circle of radius r_o which is equal to the shortest distance between p_o and the boundary. Then we select new point p_1 on the circle circumference by choosing random value of ϕ that is uniformly distributed on $[0,2\pi]$. This same process is repeated several times until the point lies within some small distance δE (one thousands times smallest dimension) to the boundary. At this point the Random Walk terminates and we then take the potential of that boundary, in this case 1V as shown in the Fig 3.1.

The above procedure is repeated N times, the potential at the point p_o will be the average reward i.e. sum of all the rewards divided by N as shown from Equation(5),

$$V(p_o) = \frac{1}{N}\sum_{i=1}^{N} V(s_i) \qquad (12)$$

Thus we get the required potential at point p_o using FRWA.

3.2 Calculation of Capacitance using Potential value obtained by using FRWA

The idea to calculate capacitance in this case is to find the charge on the plates. Once we find the charge then we can get capacitance by using Equation(13)

$$C = \frac{Q}{V} \qquad (13)$$

To find the charge we first need to find the normal component of the electric field across all the boundary of the upper plate. To find electric field we take the potential difference between two points separated by a distance one hundredth times of smallest dimension near the upper plate divided by the distance between them.

For example, to find the electric field at a particular point say (ϕ_1), we first find the potential at this point(ϕ_1) and we find the potential at a point very close to it say (ϕ_2). let the small separation between these two points be ‘r’. Then the electric field is simply given by Equation(14)

$$E = \frac{\phi_1 - \phi_2}{r} \qquad (14)$$

Now the calculation of potential at these points is done by Monte Carlo simulation i.e. using Floating Random Walk algorithm. Once after finding the electric field across all the boundary of upper plate, we then use Gauss law to add all the electric field across boundary by using Equation(15).

$$\oint_s E.ds = \frac{Q}{\epsilon_o} \qquad (15)$$

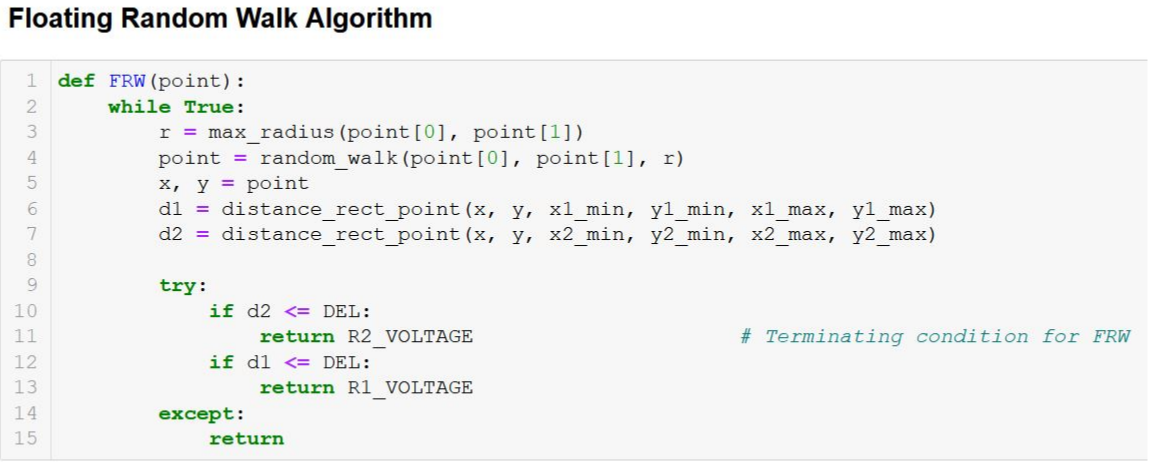
From this we get charge on upper plate by Equation(16).

$$Q = \oint_s \epsilon_o E.ds \qquad (16)$$

So, we have an electric field from Equation(14) and from Equation(16) we get charge. Now after applying the charge in Equation(13), we finally get the required capacitance.

3.3 Monte Carlo Code Snippets

Fig 3.2 shows the implementation of Floating Random Walk Algorithm which we discussed in Section 3.1 in detail.



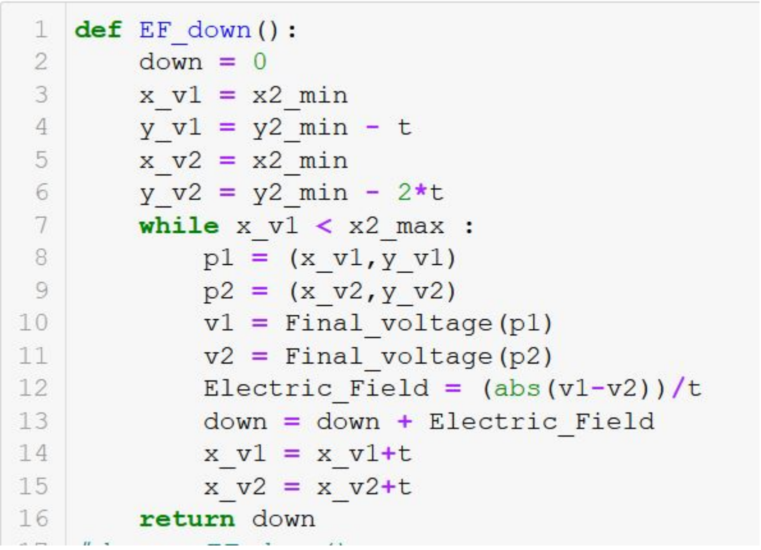
Floating Random Walk Algorithm

```
def FRW(point):
    while True:
        r = max_radius(point[0], point[1])
        point = random_walk(point[0], point[1], r)
        x, y = point
        d1 = distance_rect_point(x, y, x1_min, y1_min, x1_max, y1_max)
        d2 = distance_rect_point(x, y, x2_min, y2_min, x2_max, y2_max)

        try:
            if d2 <= DEL:
                return R2_VOLTAGE                          # Terminating condition for FRW
            if d1 <= DEL:
                return R1_VOLTAGE
        except:
            return
```

Figure 3.2: Floating Random Walk Algorithm

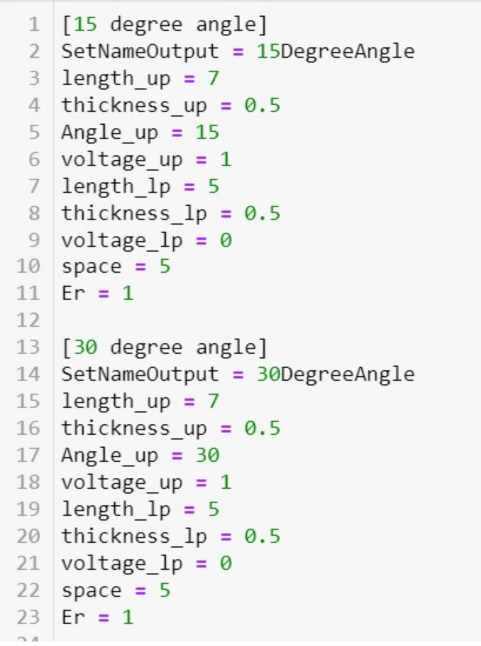
Fig 3.3 shows the implementation of calculation of Electric field by the process discussed in Section 3.2.



```
def EF_down():
    down = 0
    x_v1 = x2_min
    y_v1 = y2_min - t
    x_v2 = x2_min
    y_v2 = y2_min - 2*t
    while x_v1 < x2_max :
        p1 = (x_v1,y_v1)
        p2 = (x_v2,y_v2)
        v1 = Final_voltage(p1)
        v2 = Final_voltage(p2)
        Electric_Field = (abs(v1-v2))/t
        down = down + Electric_Field
        x_v1 = x_v1+t
        x_v2 = x_v2+t
    return down
```

Figure 3.3: Calculation of Electric field

Fig 3.4 shows the human readable configuration file with multiple sections of inputs. The configuration file consists of sections, each led by a [section] header. Between square brackets, we can put the section's name. Section is followed by key/value entries separated by "=" character. For each section of the configuration file, a separate log file with the name "SetNameOutput" is created which stores the output capacitance value for that particular section of inputs.

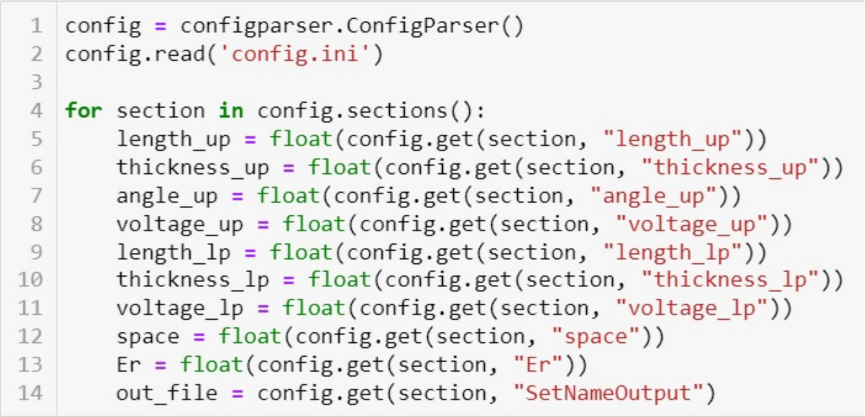


```
[15 degree angle]
SetNameOutput = 15DegreeAngle
length_up = 7
thickness_up = 0.5
Angle_up = 15
voltage_up = 1
length_lp = 5
thickness_lp = 0.5
voltage_lp = 0
space = 5
Er = 1

[30 degree angle]
SetNameOutput = 30DegreeAngle
length_up = 7
thickness_up = 0.5
Angle_up = 30
voltage_up = 1
length_lp = 5
thickness_lp = 0.5
voltage_lp = 0
space = 5
Er = 1
```

Figure 3.4: Configuration File

The configparser module from the Fig 3.5 defines functionality for reading and writing the configuration files. The configparser module has a ConfigParser class, which is responsible for parsing a list of configuration files.

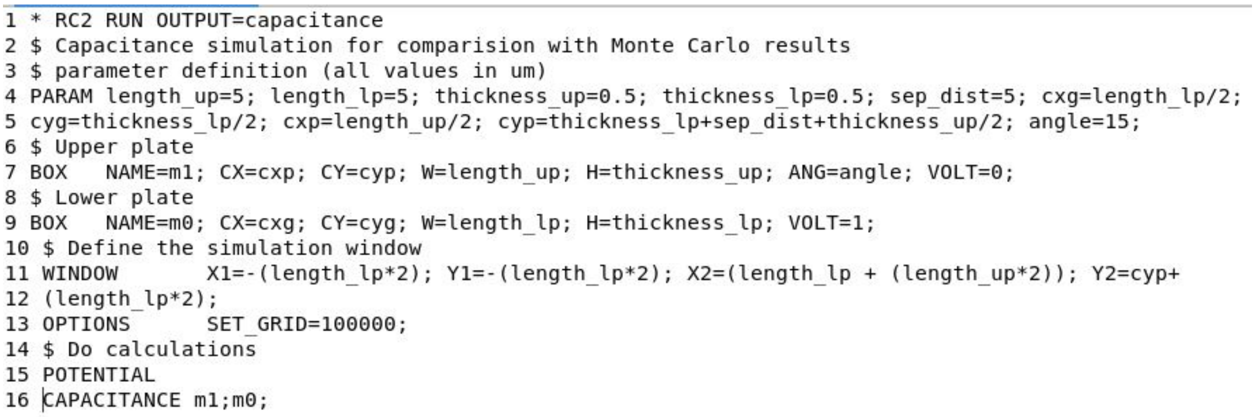


```python
config = configparser.ConfigParser()
config.read('config.ini')

for section in config.sections():
    length_up = float(config.get(section, "length_up"))
    thickness_up = float(config.get(section, "thickness_up"))
    angle_up = float(config.get(section, "angle_up"))
    voltage_up = float(config.get(section, "voltage_up"))
    length_lp = float(config.get(section, "length_lp"))
    thickness_lp = float(config.get(section, "thickness_lp"))
    voltage_lp = float(config.get(section, "voltage_lp"))
    space = float(config.get(section, "space"))
    Er = float(config.get(section, "Er"))
    out_file = config.get(section, "SetNameOutput")
```

Figure 3.5: Reading Configuration File

3.4 Raphael:



```
1 * RC2 RUN OUTPUT=capacitance
2 $ Capacitance simulation for comparision with Monte Carlo results
3 $ parameter definition (all values in um)
4 PARAM length_up=5; length_lp=5; thickness_up=0.5; thickness_lp=0.5; sep_dist=5; cxg=length_lp/2;
5 cyg=thickness_lp/2; cxp=length_up/2; cyp=thickness_lp+sep_dist+thickness_up/2; angle=15;
6 $ Upper plate
7 BOX   NAME=m1; CX=cxp; CY=cyp; W=length_up; H=thickness_up; ANG=angle; VOLT=0;
8 $ Lower plate
9 BOX   NAME=m0; CX=cxg; CY=cyg; W=length_lp; H=thickness_lp; VOLT=1;
10 $ Define the simulation window
11 WINDOW       X1=-(length_lp*2); Y1=-(length_lp*2); X2=(length_lp + (length_up*2)); Y2=cyp+
12 (length_lp*2);
13 OPTIONS      SET_GRID=100000;
14 $ Do calculations
15 POTENTIAL
16 CAPACITANCE m1;m0;
```

Figure 3.6: Raphael Code Snippet

By default, all geometric dimensions are in micrometers.

The code shown in Fig 3.6 simulates a structure containing parallel plates separated by a distance of 5um, length of 5um, thickness of 0.5um and the upper plate is rotated at an angle of 15 degrees. The upper plate is given 1V and the lower plate is grounded.

The **PARAM** command in line 4 parametrically defines the characteristic dimensions of the structure and eases the generation of several input files to compare the calculated capacitance with Monte Carlo Simulator results.

The **BOX** commands in line 7 and 9 define the Upper plate and Lower plate structures respectively.

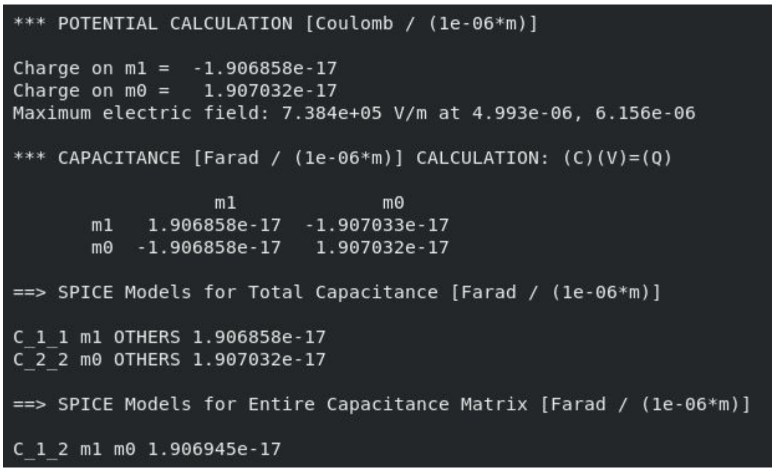
The **WINDOW** command in line 11 sets the size of the simulation window and its physical properties. Any object outside the rectangle defined by the coordinate limits X1, Y1, X2, Y2 is ignored. In our case, the size of the window is taken as twice the value of largest dimension of parameters specified.

A Neumann (or reflective) boundary condition is applied to the four sides of the simulation window. By using the Neumann boundary condition, an input structure can be reduced to one-half or one-quarter of the whole structure. This symmetry can be exploited when both geometry and bias are symmetric around the same axis.

The **SET_GRID** in line 13 generates 100000 grid points automatically allocated for the simulation.

The **POTENTIAL** command in line 15 calculates the potential distribution and the amount of charge on each electrode for the given bias. It also creates the potential file .pot and .tdf, if specified. The file contains the potential and electric field distribution data that helps to visualize in the form of heat maps.

Finally, the **CAPACITANCE** command in line 16 calculates the capacitance between the plates.



```
*** POTENTIAL CALCULATION [Coulomb / (1e-06*m)]

Charge on m1 =  -1.906858e-17
Charge on m0 =   1.907032e-17
Maximum electric field: 7.384e+05 V/m at 4.993e-06, 6.156e-06

*** CAPACITANCE [Farad / (1e-06*m)] CALCULATION: (C)(V)=(Q)

                  m1             m0
       m1   1.906858e-17  -1.907033e-17
       m0  -1.906858e-17   1.907032e-17

==> SPICE Models for Total Capacitance [Farad / (1e-06*m)]

C_1_1 m1 OTHERS 1.906858e-17
C_2_2 m0 OTHERS 1.907032e-17

==> SPICE Models for Entire Capacitance Matrix [Farad / (1e-06*m)]

C_1_2 m1 m0 1.906945e-17
```

Figure 3.7: Raphael Output Snippet

Fig 3.7 corresponds to the output of the code shown in fig 3.6. capacitance value obtained from simulation by giving above mentioned parameters as input is 19.06 aF/um.

Chapter 4

Results

4.1 Raphael Field Solver vs Monte Carlo Simulator

Monte Carlo simulation results obtained by using Floating Random Walk algorithm were validated against the results obtained from Raphael RC2 field solver.

Length of each plate = 5um, Thickness of each plate = 0.5um, Separation distance between the plates = 1um, Voltage of the Upper plate = 1V and Lower plate is grounded (i.e. 0V).

In this project we are finding the capacitance value per unit width. So all the below stated values are capacitance per unit width.

4.1.1 Capacitance value for different separation distances without Fringing Effect

Table 4.1: Capacitance values when separation distance between two plates is changed

Separation Distance (um)	**Raphael Field Solver (aF/um)**	**Monte Carlo Simulator (aF/um)**	**Error** %
0.5	88.5	90.5	2.26
1	44.2	45.5	2.94
2	22.1	22.9	3.62

The Table 4.1 shows the value of capacitance's for different Separation distances without taking Fringing Effect into consideration by keeping all other variables as constant.

4.1.2 Capacitance value for different separation distances with Fringing Effect

Table 4.2: Capacitance values when separation distance between two plates is changed

Separation Distance (um)	Raphael Field Solver (aF/um)	Monte Carlo Simulator (aF/um)	Error %
0.5	107.5	101.3	-5.77
1	60.2	59.8	-0.66
2	35.3	33.9	-3.97

The Table 4.2 shows the value of capacitance's for different Separation distances by keeping all other variables as constant. We can see that the value of capacitance decreases with the increase of separation distance between the plates.

4.1.3 Capacitance value for different lengths of the upper plate

Table 4.3: Capacitance values when length of the upper plate changed

Length of the upper plate (um)	Raphael Field Solver (aF/um)	Monte Carlo Simulator (aF/um)	Error %
3	46.1	44.6	-3.25
5	60.2	59.8	-0.66
7	64.5	63.4	-1.7

The Table 4.3 shows the value of capacitance's for different lengths of the upper plate by keeping all other variables as constant. We can see that the value of capacitance increases with the increase of upper plates length.

4.1.4 Capacitance value for different thickness of the upper plate

Table 4.4: Capacitance values when thickness of the upper plate changed

Thickness of the upper plate (um)	Raphael Field Solver (aF/um)	Monte Carlo Simulator (aF/um)	Error %
0.25	59.6	58.9	-1.17
0.5	60.2	59.8	-0.66
0.75	60.6	60.4	-0.33

The Table 4.4 shows the value of capacitance's for different thickness values of the upper plate by keeping all other variables as constant. We can see that the value of capacitance increases with the increase of the upper plates thickness.

4.1.5 Capacitance value for different positions of the plates

Table 4.5: Capacitance values when position of the plates is changed

Position	**Raphael Field Solver (aF/um)**	**Monte Carlo Simulator (aF/um)**	**Error %**
Perfectly Overlapped	60.2	59.8	-0.66
Partially Overlapped	47.6	48.6	2.1
No Overlapping	28.1	30.1	7.11

The Table 4.5 shows the value of capacitance's for different position of the plates by keeping all other variables as constant. We can see that the value of capacitance decreases with the decrease in the common area between the plates.

4.1.6 Capacitance value when Upper plate is rotated at a certain angle

Length of each plate = 5um, Thickness of each plate = 0.5um, Separation distance between the plates(d) = 5um, Voltage of the Upper plate = 1V and Lower plate is grounded (i.e. 0V). Upper plate is rotated at an angle θ from its center.

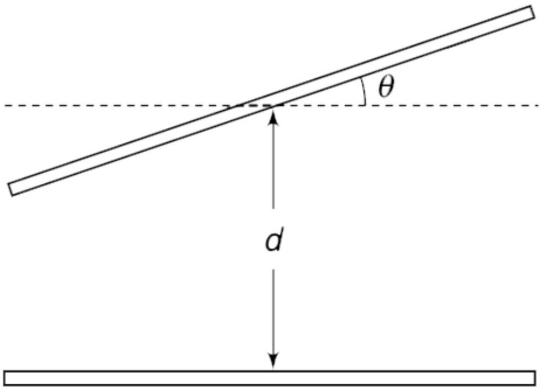


Figure 4.1: Upper plate is inclined at an angle θ

The Table 4.6 shows the value of capacitance's for different angles of the upper plate by keeping all other variables as constant. We can see that the value of capacitance increases with the increase in the angle (θ).

Table 4.6: Capacitance values when the angle of the upper plate is changed

Angle = θ (degrees)	**Raphael Field Solver (aF/um)**	**Monte Carlo Simulator (aF/um)**	**Error %**
15	19	19.5	2.63
30	19.4	19.8	2.06
45	19.9	20.4	2.51
60	20.5	21.3	3.9
75	21	22	4.76

In the above case, when the upper plate is rotated at a certain angle (θ) the edges of the plates aren't perfectly aligned. So, in this case we extended the length of the upper plate such that the edges of the plates are perfectly aligned as shown in Fig 4.2.

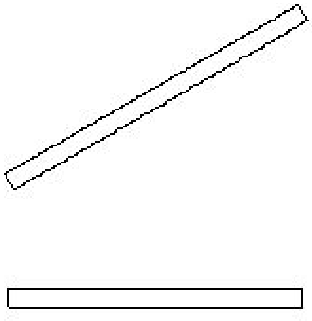


Figure 4.2: Edges are perfectly aligned after rotation

Table 4.7: Capacitance values for different angles of the upper plate when edges of the plates are perfectly aligned

Angle = θ (degrees)	**Raphael Field Solver (aF/um)**	**Monte Carlo Simulator (aF/um)**	**Error %**
15	19.26	20.3	5.12
30	20.37	21.26	4.18
45	23.15	24.03	3.66
60	35.43	36.36	2.55

The Table 4.7 shows the value of capacitance's for different angles of the upper plate when edges of the plates are perfectly aligned. We can see that the value of capacitance's in Table 4.7 are increased slightly compared to the previous Table 4.6, where the edges of plates are not perfectly aligned.

4.2 Heatmaps of Potential Distribution:

Heatmaps of Potential Distribution for different positions of upper plate were visualised using Sentaurus Visual.

4.2.1 Upper plate is shifted along X-axis

The Fig 4.3 - 4.5 shows the potential distribution around the plates for three different positions of upper plate. The points in the region closer to the Upper plate has potential value close to 1V and the points in the region closer to the Lower plate has potential value close to 0V.

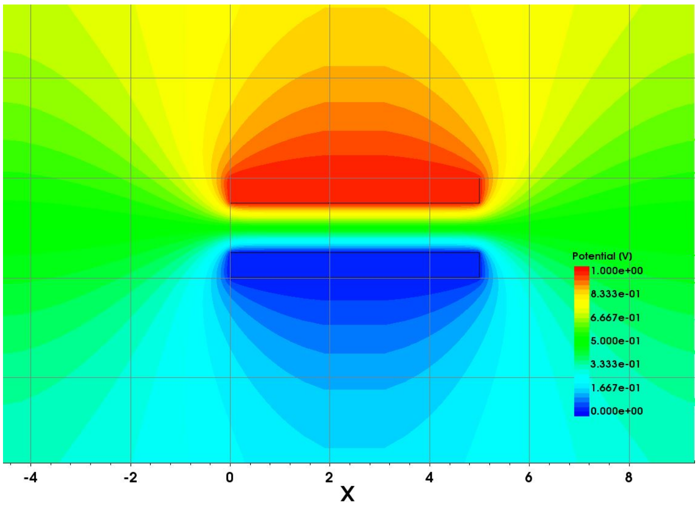


Figure 4.3: Perfectly Overlapped

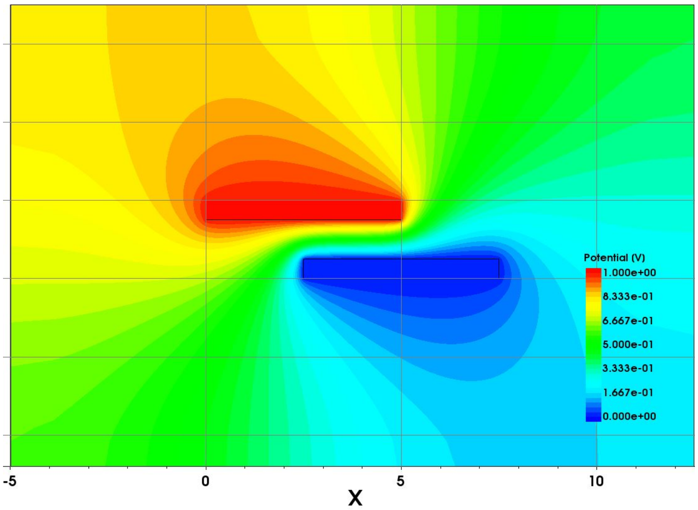


Figure 4.4: Partially Overlapped

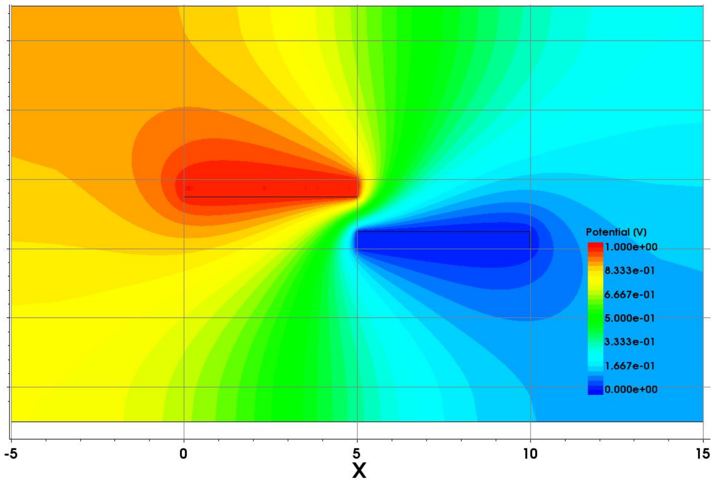


Figure 4.5: No Overlapping

4.2.2 Upper plate rotated at an angle θ

The Fig 4.6 - 4.9 shows the potential distribution around the plates for various angles at which upper plate is rotated. The points in the region closer to the Upper plate has potential value close to 1V and the points in the region closer to the Lower plate has potential value close to 0V.

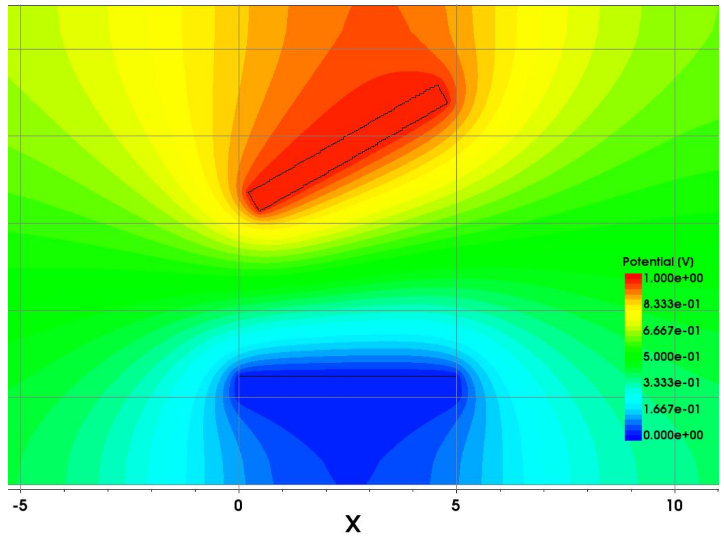


Figure 4.6: Upper plate rotated at an angle 30 degrees

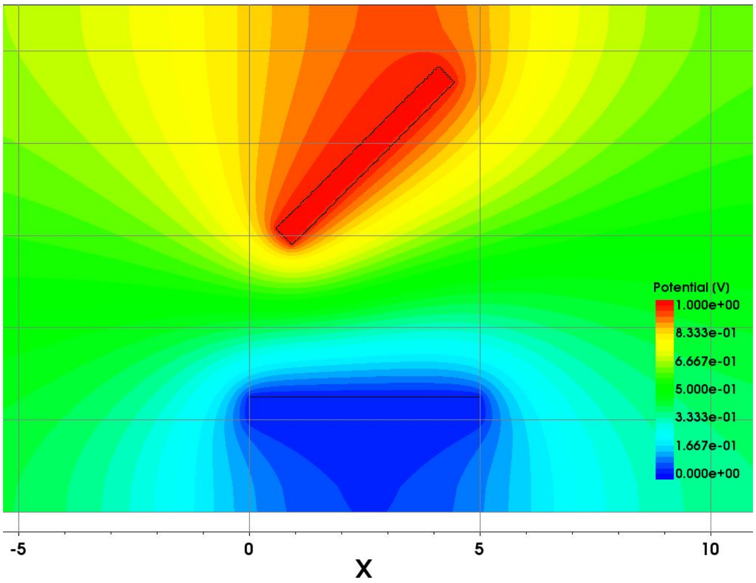


Figure 4.7: Upper plate rotated at an angle 45 degrees

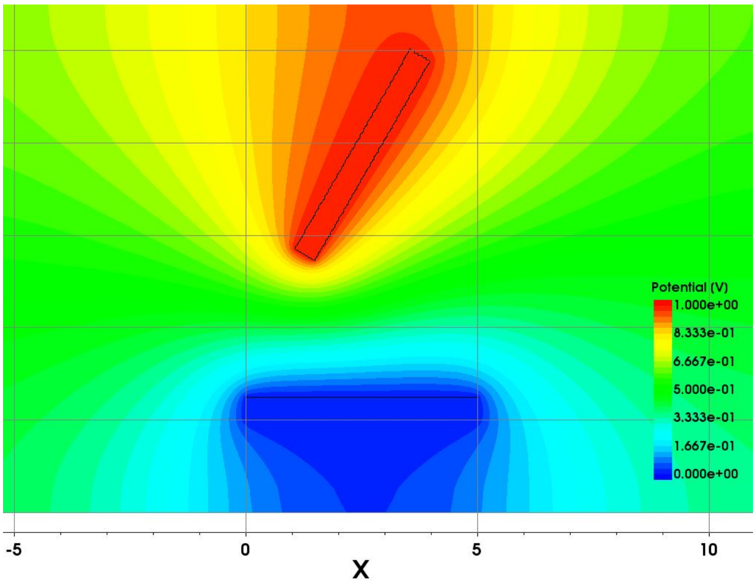


Figure 4.8: Upper plate rotated at an angle 60 degrees

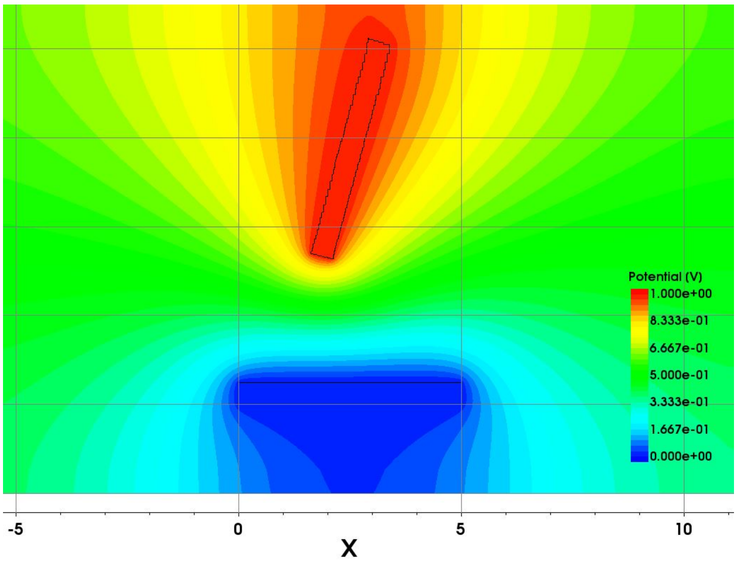


Figure 4.9: Upper plate rotated at an angle 75 degrees

4.2.3 Upper plate rotated at an angle θ and edges of the plates are perfectly aligned

The Fig 4.10 - 4.13 shows the potential distribution around the plates for various angles at which upper plate is rotated when edges of the plates are perfectly aligned. The points in the region closer to the Upper plate has potential value close to 1V and the points in the region closer to the Lower plate has potential value close to 0V.

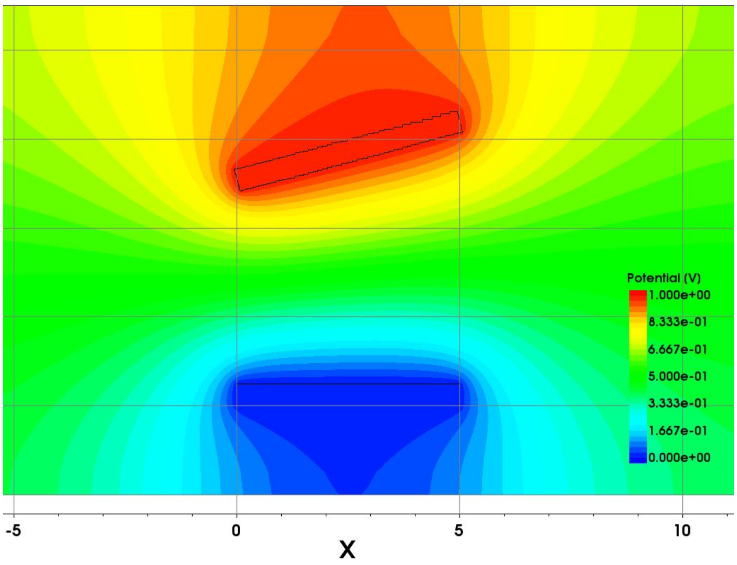


Figure 4.10: Upper plate rotated at an angle 15 degrees and edges of the plates are perfectly aligned

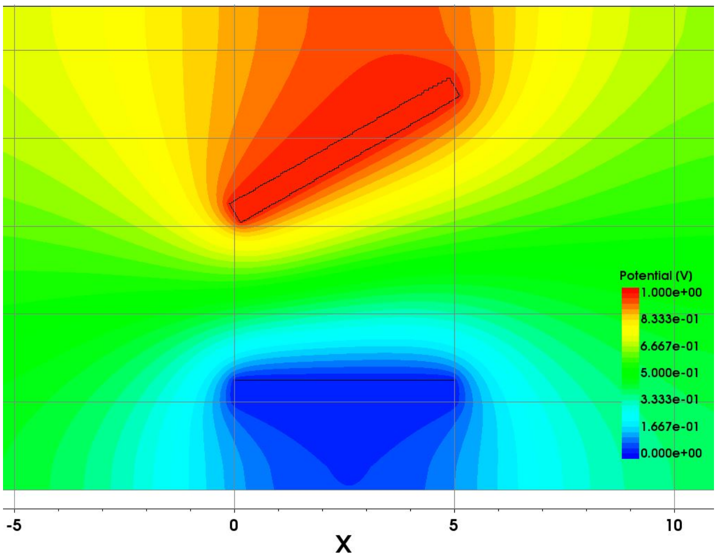


Figure 4.11: Upper plate rotated at an angle 30 degrees and edges of the plates are perfectly aligned

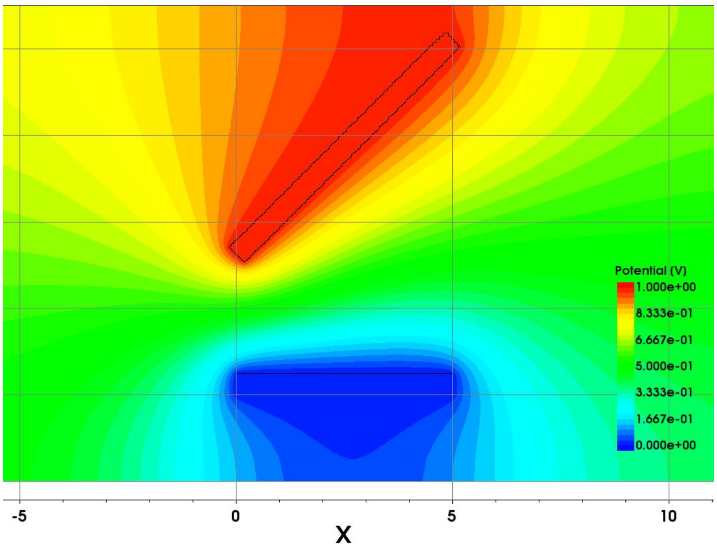


Figure 4.12: Upper plate rotated at an angle 45 degrees and edges of the plates are perfectly aligned

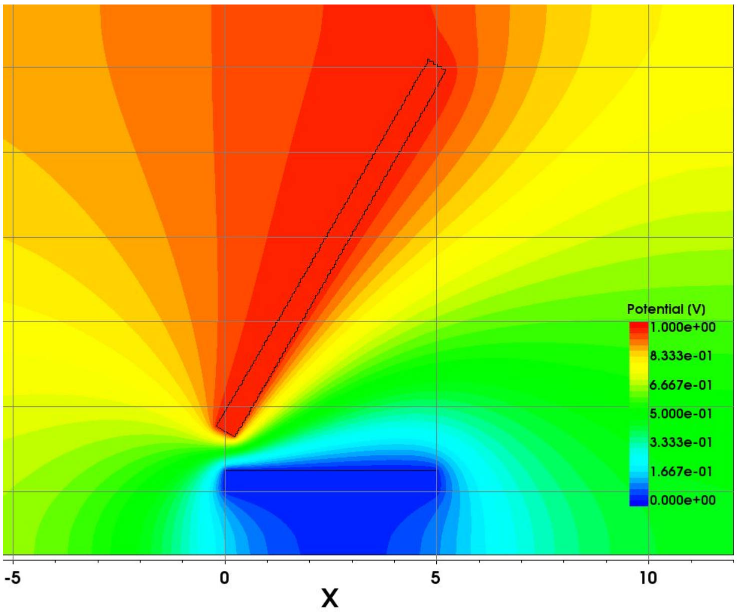


Figure 4.13: Upper plate rotated at an angle 60 degrees and edges of the plates are perfectly aligned

4.3 Raphael vs Monte Carlo comparison plots

In the Fig 4.14 - 4.19 Dotted lines shows the trend of the capacitance values obtained from Raphael Field Solver and the Solid lines shows the trend of the capacitance values obtained from Monte Carlo simulator.

The Fig 4.14 shows the family of curves representing the capacitance values obtained from Raphael field solver and Monte Carlo simulator for different overlapping cases versus length of the upper plate.

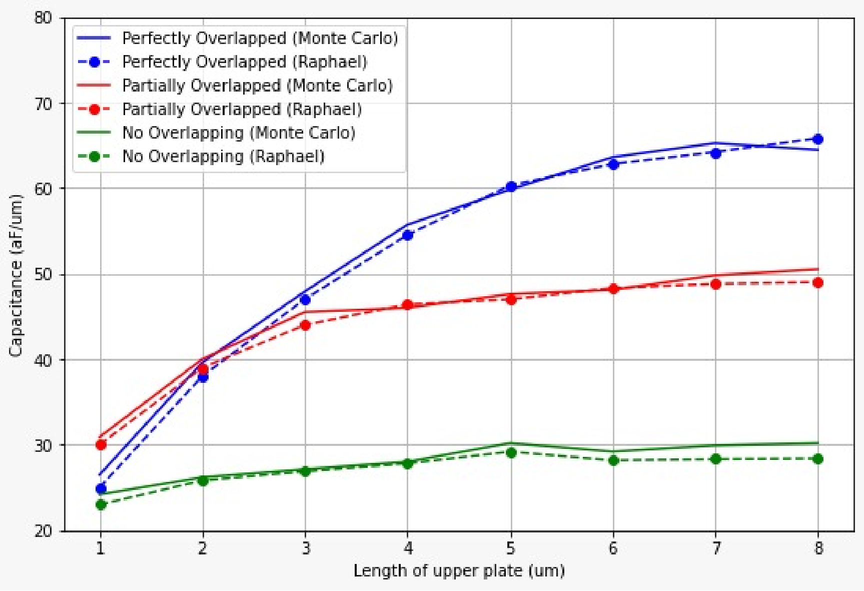


Figure 4.14: Capacitance vs Length of the upper plate

The Fig 4.15 shows the family of curves representing the capacitance values obtained from Raphael field solver and Monte Carlo simulator for different overlapping cases Vs thickness of upper plate.

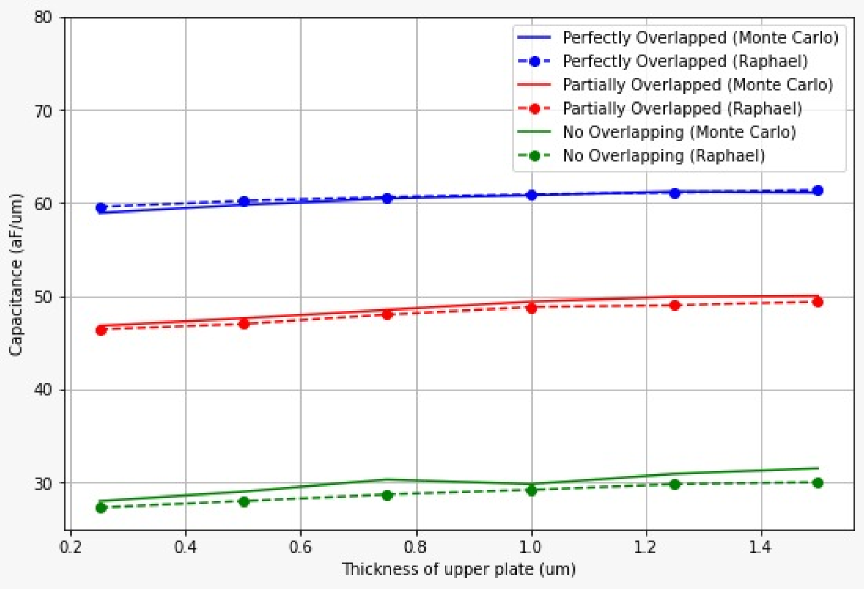


Figure 4.15: Capacitance vs Thickness of the upper plate

The Fig 4.16 shows the family of curves representing the capacitance values obtained from Raphael field solver and Monte Carlo simulator for different overlapping cases Vs separation distance between two plates.

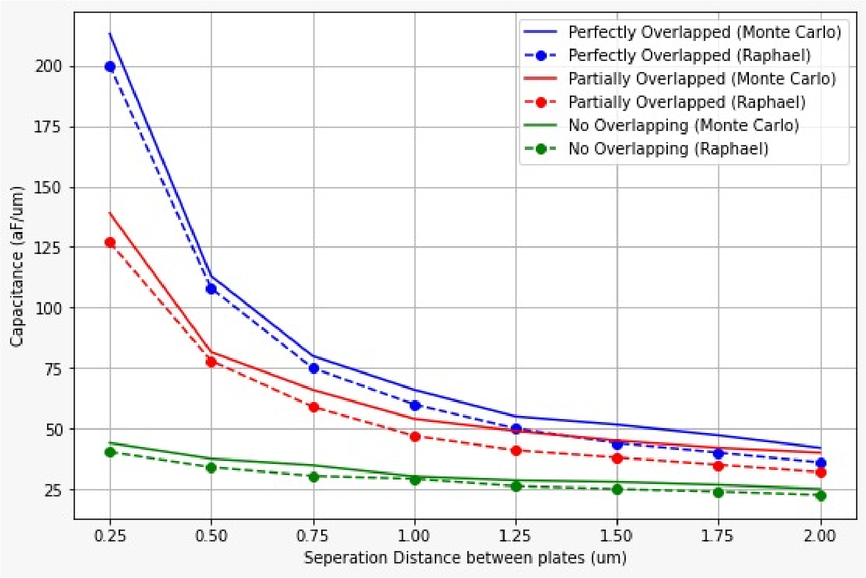


Figure 4.16: Capacitance vs Separation distance between plates

The Fig 4.17 shows the family of curves representing the capacitance values obtained from Raphael field solver and Monte Carlo simulator for different values of length of upper plate Vs Angle of the upper plate.

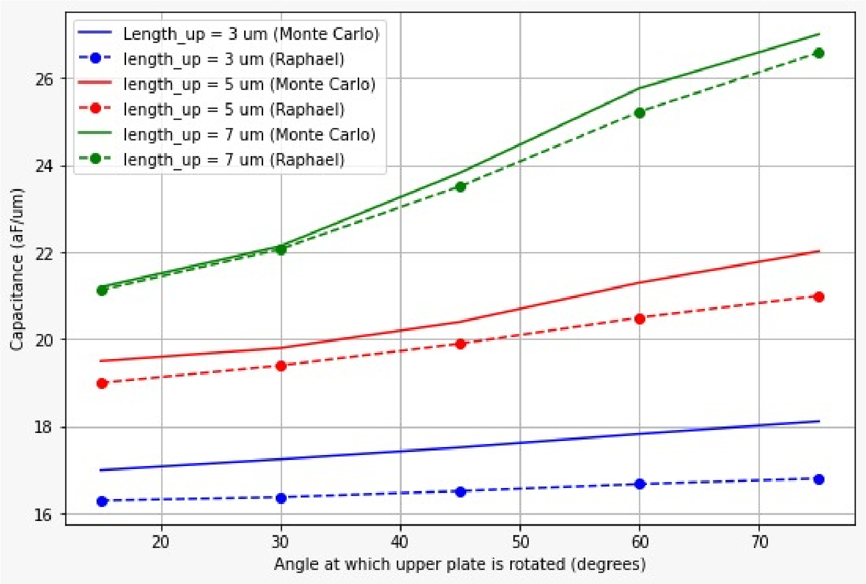


Figure 4.17: Capacitance vs Angle of upper plate for different lengths of upper plate

The Fig 4.18 shows the family of curves representing the capacitance values obtained from Raphael field solver and Monte Carlo simulator for different values of thickness of upper plate Vs Angle of the upper plate.

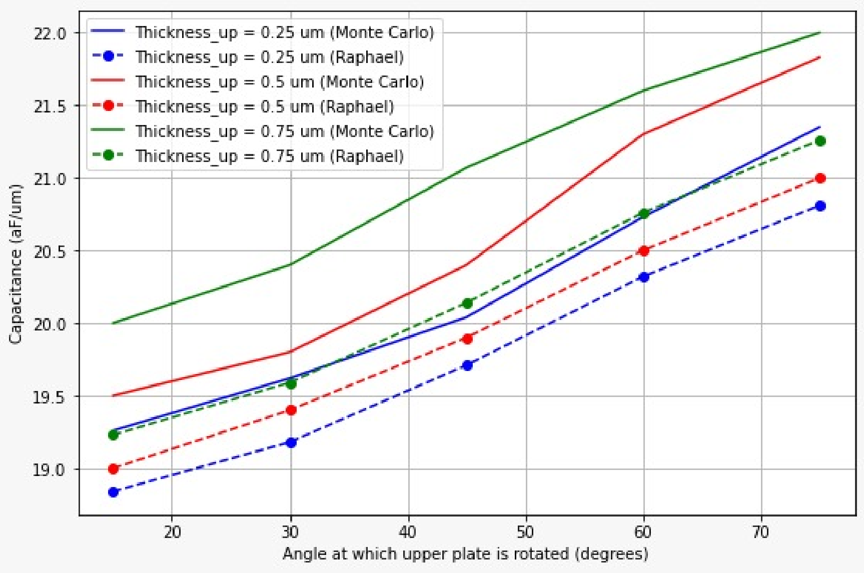


Figure 4.18: Capacitance vs Angle of upper plate for different thickness of upper plate

The Fig 4.19 shows the family of curves representing the capacitance values obtained from Raphael field solver and Monte Carlo simulator for different values of separation distance between two plates Vs Angle of the upper plate.

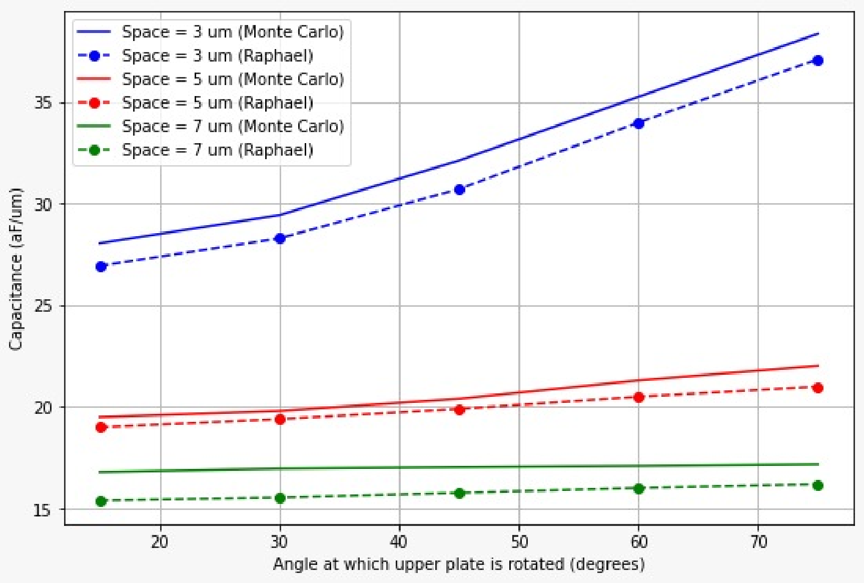


Figure 4.19: Capacitance vs Angle of upper plate for different separation distances between plates

Chapter 5

GitHub Link

The full python code and configuration files are available in the GitHub link provided below.

Link

github.com/bhanu006/Monte-Carlo-based-2D-CAD-Tool-for-Calculating-Capacitance-of-a-User-Parameterized-Geometry

Chapter 6

Conclusions

1. Finally, after completing the literature review of the different methods, it is evident that in the Floating Random Walk, neither the step sizes nor the directions of the walk are fixed in advance like we do in the Fixed Random Walk Algorithm. A Floating Random Walk bypasses many intermediate steps of a Fixed Random Walk in favor of a long jump.

2. Fewer steps are needed for Floating Random Walk algorithm to reach the boundary, and so computation is much more rapid than in fixed random walk.

3. By using Floating Random Walk Algorithm we can find potential at every point where ever we want.

4. Accuracy of output increases with increase in number of iterations.As we can see in the results, error percentage decreases with increase in the number of iterations.

5. As the separation distance between the parallel plates increases the value of the capacitance decreases and vice versa.

6. Moreover, when fringing effect is neglected the value of the capacitance obtained from the simulation is almost similar to the capacitance value that is calculated theoretically from equation(7). Therefore this validates that the code is working perfectly.

7. With decrease in overlapping area of parallel plate capacitor, the capacitance value also decreases.

Chapter 7

Future Work

1. In this project, we mainly concentrated on extracting the capacitance between two rectangular plates for better understanding and computational purposes but we can explore, study and perform calculations for many arbitrary shapes with symmetry and without symmetry.

2. We can extend the Monte Carlo simulator and Raphael Field Solver to take different kind of geometries as input.

3. In this project Monte Carlo Simulator works only on single dielectric case, but in general this can also be improved to multi-dielectric cases.

4. We can implement this project further into 3D domain, which will be more realistic and can be used in real life.

5. We can improve execution time and space of the Monte Carlo simulator by using advanced level coding.

Chapter 8

References

1. E. Aykut Dengi and Ronald A. Rohrer, "Boundary Element Method Macromodels for 2-D Hierarchical Capacitance Extraction", Design Automation Conference, pp.218-223, June 1998.

2. Genlong Chen, Hengliang Zhu, Tao Cui, Zhiming Chen, Xuan Zeng, wie Cai, "Par AFEM Cap : A Parallel Adaptive Finite-Element Method for 3-D VLSI Interconnect Capacitance Extraction" IEEE Transactions on Microwave Theory and Techniques, February 2012.

3. X. Ferrieres, J.-P. Parmantier, S. Bertuol and A.R. Ruddle, "Application of a hybrid finite difference/finite volume method to solve an automotive EMC problem", IEEE Transactions on Electromagnetic Compatibility, vol. 46, no. 4, pp. 624-634, Nov. 2004.

4. W. Yu, "Applications of Monte Carlo method to 3-D capacitance calculation and large matrix decomposition," 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2016, pp. 227-230, doi: 10.1109/ICSICT.2016.7998884.

5. J. N. Jere and Y. L. Le Coz, "An improved floating-random-walk algorithm for solving the multi-dielectric Dirichlet problem," in IEEE Transactions on Microwave Theory and Techniques, vol. 41, no. 2, pp. 325-329, Feb. 1993, doi: 10.1109/22.216475.

6. M. N. O. Sadiku, "Monte Carlo methods in an introductory electromagnetic course," in IEEE Transactions on Education, vol. 33, no. 1, pp. 73-80, Feb. 1990, doi: 10.1109/13.53630.

7. M. N. O. Sadiku and D. T. Hunt, "Solution of Dirichlet problems by the Exodus method," in IEEE Transactions on Microwave Theory and Techniques, vol. 40, no. 1, pp. 89-95, Jan. 1992, doi: 10.1109/22.108327.

8. www.synopsys.com/silicon/tcad/framework.html.

9. Raphael™ Reference Manual Version S-2021.06, June 2021.

10. RaphaelTM Interconnect Analysis Program Tutorial Version Z-2006.12, December 2006.

11. www.synopsys.com/silicon/tcad/interconnect-simulation/raphael.html.

12. en.wikipedia.org/wiki/Harmonic_function#The_mean_value_property.