

# **MAGNETIC MIRROR EFFECT IN MAGNETRON PLASMA:MODELING OF PLASMA PARAMETERS**

## **A PROJECT REPORT**

*Submitted in partial fulfillment for the award of the degree of*

**Bachelor of Technology**

*in*

**Mechanical Engineering**

*by*

**SASHI KANT SHAH - 18BEM0145**

**KAUSHAL TIMILSINA-18BME2104**

**HRISHAV MISHRA-18BME2109**

**School of Mechanical Engineering**



**VIT<sup>®</sup>**

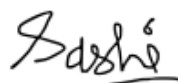
**Vellore Institute of Technology**


(Deemed to be University under section 3 of UGC Act, 1956)

April 2022

**DECLARATION BY THE CANDIDATE**

We hereby declare that the project report entitled “**MAGNETIC MIRROR EFFECT IN MAGNETRON PLASMA:MODELING OF PLASMA PARAMETERS**” submitted by me to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the degree of < **Bachelor of Technology** > in < **Mechanical Engineering** > is a record of bonafide project work carried out by me under the supervision of < **Professor Sitaram Dash** >. We declare that this report represents my concepts written in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We further declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed. Further we affirm that the contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

 18BEM0145

 18BME2109

 18BME2104

Place : Vellore

Signature of the Candidate(s)

Date: 15<sup>th</sup> February 2022



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## **School of Mechanical Engineering**

### **BONAFIDE CERTIFICATE**

This is to certify that the project report entitled “**MAGNETIC MIRROR EFFECT IN MAGNETRON PLASMA:MODELING OF PLASMA PARAMETERS**” submitted by **<SASHI KANT SHAH>(18BEM0145) , <KAUSHAL TIMILSINA>(18BME2104) &<HRISHAV MISHRA> (18BME2109)** to Vellore Institute of Technology University, Vellore, in partial fulfillment of the requirement for the award of the degree of **< Bachelor of Technology > in < Mechanical Engineering >** is a record of bona fide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

26/04/22

**Project Guide**

**Head of the Department**

**Internal Examiner**

**External Examiner**

## ACKNOWLEDGEMENT

We would like to express our special gratitude to our project guide Prof. Sitaram Dash for his personal support and advice to carry out the thesis and complete it. We would further like to extend our gratitude to the School of Mechanical Engineering and VIT for providing us with this opportunity to enhance our previous semester knowledge practically through this capstone project and strengthen our understanding through the thesis for the same. We have got to learn a lot more about this project regarding the understanding and improvement of practical applications of Magnetron sputtering which will be very helpful for us as we progress in our field of study. We are also grateful for having the opportunity of learning team work which has helped us in recognizing our potential both on an individual as well as a team level. Being able to understand how each and every one of us are able to contribute through our individual thinking and building the outcome through amalgamation of the same is this something we are truly thankful for.



Sashi Kant Shah



Hrishav Mishra



Kaushal Timilsina

Place : Vellore

<Signature of the Student>

(Name of the Student)

Date : 15<sup>th</sup> February 2022

## TABLE OF CONTENTS

| <b>CHAPTER NO.</b> | <b>TITLE</b>                                   | <b>PAGE NO.</b> |
|--------------------|--|-----------------|
|                    | <b>EXECUTIVE SUMMARY</b>                       | i               |
|                    | <b>LIST OF TABLES</b>                          | ii              |
|                    | <b>LIST OF FIGURES</b>                         | iii             |
|                    | <b>LIST OF SYMBOLS AND ABBREVIATIONS</b>       | iv              |
|                    |  |                 |
| <b>1</b>           | <b>INTRODUCTION AND LITERATURE REVIEW</b>      | 1               |
|                    | 1.1 Introduction                               | 1               |
|                    | 1.2 Literature Review                          | 2               |
|                    | 1.3 Knowledge gained from the literature       | 6               |
|                    | 1.4 Gaps identified                            | 7               |
|                    | 1.5 Objectives of the project                  | 8               |
|                    | 1.6 Motivation                                 | 9               |
| <b>2</b>           | <b>METHODOLOGY AND EXPERIMENTAL WORK</b>       |                 |
|                    | 2.1 Project Execution Stages                   | 9               |
|                    | 2.2 Technical Specifications                   | 12              |
|                    | 2.3 Design Approach Details                    | 15              |
|                    | 2.4 Realistic and Design Constraints addressed | 17              |
|                    | 2.5 Codes and Standards                        | 19              |
| <b>3</b>           | <b>RESULTS AND DISCUSSION</b>                  | 23              |
|                    | 3.1 Checks                                     | 23              |
|                    | 3.2 Study 1                                    | 28              |
|                    | 3.3 Study 2                                    | 31              |
|                    | 3.4 Study 3                                    | 35              |
| <b>4</b>           | <b>CONCLUSIONS</b>                             | 39              |
|                    | 4.1 Contributions to the literature            | 40              |

|  |                                     |    |
|--|-------------------------------------|----|
|  | 4.2 Scope for future work           | 41 |
|  | <b>REFERENCES</b>                   | 42 |
|  | <b>PUBLICATIONS</b>                 | 42 |
|  | <b>PLAGIARISM REPORT</b>            |    |
|  | <b>CONTRIBUTION STATEMENT</b>       | 43 |
|  | <b>PROGRAMME OUTCOME DATA SHEET</b> |    |

## EXECUTIVE SUMMARY

The aim of this work is to understand the magnetic mirror configuration that is ideal in a magnetic plasma trap chamber which can be used in Magnetron sputtering. Particles in a plasma have different speeds depending on the initial distribution which is based on parameters like the plasma temperature. The speeds of particles change depending on the fields. Based on the magnetic mirror effect, one can determine which particle (having certain velocities) can escape the magnetic trap and which of those are reflected. The less the particles escape the magnetic trap, the more of the flux is used in forming coatings and less of the ionized gas is wasted. This is very useful in understanding the required gas supply and rate of deposition.

The model used in the project is mostly similar to the single particle model, where in a collection of particles, each particle evolves under the influence of the fields set up by the apparatus; governed by the Lorentz force. However, we incorporate a little bit of the Kinetic theory in that we are interested in different initial velocity distributions for particles in the plasma and how the velocity distribution changes over time. To achieve this in a simulation, the Lorentz force equations are discretized and then solved using the Boris Algorithm; a standard algorithm for simulating charged particles in electric and magnetic fields.

**Keywords:** *Magnetic mirror, Magnetron sputtering , Lorentz force , Kinetic theory, Boris Algorithm Thermogravimetric analysis; emission; monitoring;*

**LIST OF TABLES**

| <b>Table No.</b> | <b>Title</b> | <b>Page no.</b> |
|------------------|--------------|-----------------|
|------------------|--------------|-----------------|

|                     |  |  |
|---------------------|--|--|
| There are no tables |  |  |
|---------------------|--|--|

## LIST OF FIGURES

| <b>Fig<br/>no.</b> | <b>Title</b>   | <b>Page<br/>No.</b> |
|--------------------|--|---------------------|
| Fig. 3.1           | Study 1 positions of a particle                          | 29                  |
| Fig. 3.2           | Study 1 positions of 10 particles                        | 29                  |
| Fig. 3.3           | Study 1 x-component of positions                         | 29                  |
| Fig. 3.4.          | Study 1 x-component of velocities                        | 29                  |
| Fig. 3.5.          | Study 1 y-component of positions                         | 30                  |
| Fig. 3.6.          | Study 1 y-component of velocities                        | 30                  |
| Fig. 3.7.          | Study 1 z-component of positions                         | 30                  |
| Fig. 3.8.          | Study 1 z-component of velocities                        | 30                  |
| Fig. 3.9.          | Study 2.1 positions of a particle                        | 31                  |
| Fig. 3.10.         | Study 2.1 positions of 10 particles                      | 31                  |
| Fig. 3.11.         | Study 2.1 x-component of positions                       | 32                  |
| Fig. 3.12.         | Study 2.1 x-component of velocities                      | 32                  |
| Fig. 3.13.         | Study 2.1 y-component of positions                       | 32                  |
| Fig. 3.14.         | Study 2.1 y-component of velocities                      | 32                  |
| Fig. 3.15.         | Study 2.1 z-component of positions                       | 32                  |
| Fig. 3.16.         | Study 2.1 z-component of velocities                      | 32                  |
| Fig. 3.17.         | Study 2.2 positions of a particle                        | 34                  |
| Fig. 3.18.         | Study 2.2 positions of 10 particles                      | 34                  |
| Fig. 3.19.         | Study 2.2 x-component of positions                       | 34                  |
| Fig. 3.20.         | Study 2.2 x-component of velocities                      | 34                  |
| Fig. 3.21.         | Study 2.2 y-component of positions                       | 34                  |
| Fig. 3.22.         | Study 2.2 y-component of velocities                      | 34                  |
| Fig. 3.23.         | Study 2.2 z-component of positions                       | 35                  |
| Fig. 3.24.         | Study 2.2 z-component of velocities                      | 35                  |
| Fig. 3.25.         | Histogram of Maxwellian sampled speeds<br>from study 2   | 36                  |
| Fig. 3.26.         | Histogram of Parabolic sampled speeds<br>from study 3 16 | 36                  |
| Fig. 3.27.         | Study 3.1 positions of 10 particles                      | 37                  |



## **APPENDIX -9**

### **LIST OF SYMBOLS AND ABBREVIATIONS**

|          |                        |
|----------|------------------------|
| m        | meter                  |
| cm       | Centimetre             |
| s        | second                 |
| mT       | milliTesla             |
| J        | Joule                  |
| V        | Volt                   |
| K        | Kelvin                 |
| g        | gram                   |
| mol      | mole                   |
| $\mu_0$  | permeability of vaccum |
|          |                        |
| <b>x</b> | position               |
| R        | Universal Gas Constant |
| $\pi$    | pi                     |
| T        | temperature            |
| M        | molar mass             |
| $\sigma$ | Standard deviation     |

# CHAPTER 1

## INTRODUCTION AND LITERATURE REVIEW

An attempt has been made to give an overview of a plasma simulation from the ground up. It is true that numerous works have been done in plasma simulations. However, these systems are often very complicated and intended for special purposes like astrophysical understanding or nuclear fusion. If we were to work on such a project, it might have taken us no less than a year or several; and it is doubtful how much we would be able to understand. A lot of the complexity comes from the fact that a simulation solving Maxwell's equations will take much more memory and time.

### 1.1 INTRODUCTION

One of the team members- 18BME2104 Kaushal studied the class MEE4005 Surface Engineering taught by Professor Sitaram Dash- our internal guide, during the Fall Semester 2021. Many interesting plasma based surface engineering techniques were studied during the class, one such technique being Magnetron Sputtering. This inspired the study of plasma in this project.

1.1 Plasma One comes across many definitions of plasma including: fourth state of matter, ionized gas, a non equilibrium state of matter with dynamical characteristics due to electrodynamics, etc. However, it is best to describe plasma with some characteristic parameters, when one attempts to describe a plasma quantitatively. Some quantities that help define a plasma are:

#### 1. Number density, $n$

Number density of a plasma describes the number of particles per unit volume. Plasma contains charged particles or ionized species. However, a plasma might at the same time also contain neutral atoms and molecules but also particles of different species- charged and neutral. If multiple species are contained in a plasma system, number densities of each species could be used to describe the system. For example, a plasma may contain electrons, charged ions and neutral atoms and molecules. Mass density is defined as  $\rho := mn$  and is often used alongside number density, where  $m$  is the mass of the species.

#### 2. Ionization, $\alpha$

Defined as  $\alpha := \frac{n_{charged}}{n_{charged} + n_{neutral}}$ , the ionization of a plasma describes the fraction of charged particles, with  $\alpha = 1$  meaning that all the particles are charged and  $\alpha = 0$  meaning that all the particles are neutral

### 3. Temperature, $T$

The temperature of a plasma describes the average kinetic energy in the plasma. When a gas is ionized to form a plasma, the ionization  $\alpha$  can depend on the temperature of the plasma.

### 4. Mean free path, $\lambda_{mf p}$

The gas-like behavior of a plasma is characterized by large mean free path. The mean free path and the thermal velocity of the particles as described by the temperature, are related by the timescale of collisions as  $\lambda_{mf p} := v_{th} \tau$  where  $\tau$  is the timescale of collisions

### 5. Debye Length, $\lambda_D$

In a plasma, electrostatic Coulomb interactions between charged particles compete with random thermal speed of the particles described by the temperature of the plasma. The Debye sphere is an imaginary sphere around a charged particle, where oppositely charged particles are attracted and in doing so screen the charge of the central particle from the outer plasma so that the electrostatic influence of a particle is limited to the Debye sphere surrounding it. This is why plasma's are often said to be Quasi-neutral as charge screening leads to a neutral behavior electrostatically on a scale much larger than the Debye length. The Debye length is defined as the radius of the Debye sphere

### 6. Plasma beta parameter, $\beta$

The beta parameter defined as  $\beta := 8\pi nT/B^2$  describes the ratio of the thermal and magnetic energies of the plasma, as particles in random thermal motions compete with the Lorentz force. Many other parameters like the Larmor radius, quantities like frequencies of different waves describing the dynamics; are also important in describing a plasma. However, as we shall see later; particles in the plasma are more important to this project than many interesting parameters describing the plasma.

## 1.2 LITERATURE REVIEW

A large part of the literature review for the project consisted of studying basic plasma physics in order to understand how we could formulate the project and proceed ahead.

## Plasma as a system

Various models are used to describe Plasma as a system. Some of the common approaches are:

### 1. Single particle description

This model is used to describe the motion of a charged particle.

$$\frac{d\mathbf{v}}{dt} = \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad \boxed{\dots(1)}$$

Describing many particles evolving under the influence of Lorentz force, it is easy to describe a plasma assuming particles hardly ever collide. This is to say that in the simplest case, under this model particles evolve but the particles do not produce any electric and magnetic fields of their own or affect the external applied fields and hence also do not interact with other particles.

### 2. Kinetic theory

The kinetic theory describes the plasma as collection of particles whose state (position and velocity) is treated as a random variable with a density function  $f(x, y, z, v_x, v_y, v_z, t)$  which describes the number of particles at position  $(x, y, z)$  at time  $t$  with velocities between  $v_x$  and  $v_x + dv_x$ ,  $v_y$  and  $v_y + dv_y$ ,  $v_z$  and  $v_z + dv_z$  in directions  $x, y$  and  $z$  respectively. The expression

$$\int_{all} dv_x \int_{all} dv_y \int_{all} dv_z f(\mathbf{x}, \mathbf{v}, t) \quad \boxed{\dots(2)}$$

gives the number of particles at position  $\mathbf{x}$ , at time  $t$ . For a simple choice of notation, it is often written as

$$\int_{all} d^3v f(\mathbf{x}, \mathbf{v}, t) \quad or \quad \int_{all} d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t) \quad \boxed{\dots(3)}$$

A density function is said to be normalized if

$$\int_{all} d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t) = 1 \quad \boxed{\dots(4)}$$

Such a density function is also denoted with a hat as  $\hat{f}(\mathbf{x}, \mathbf{v}, t)$ .

The average velocity  $\bar{v}$  for a density function  $\hat{f}(\mathbf{x}, \mathbf{v}, t)$  is calculated as

$$\int_{all} d\mathbf{v} \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) \quad \dots(5)$$

Various other features of the distribution are: the Root Mean Square velocity  $v_{rms}$ , the average absolute velocity  $|\mathbf{v}|$ , the average velocity in z direction  $|v_z|$ , etc. The evolution of the particles is described as the changing of the density function. Boltzmann equation is often used to describe this dynamics:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \partial_{\mathbf{v}} f = \left( \frac{\partial f}{\partial t} \right)_c \quad \dots(6)$$

While it may look complicated at first, it is actually just a short form for:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} v_x + \frac{\partial f}{\partial v_x} a_x + \frac{\partial f}{\partial y} v_y + \frac{\partial f}{\partial v_y} a_y + \frac{\partial f}{\partial z} v_z + \frac{\partial f}{\partial v_z} a_z \quad \dots(7)$$

For simplicity, let us look at the 1-dimensional (2-phase dimensional) version of this expression

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial v_x} \frac{dv_x}{dt} \quad \dots(8)$$

If the right hand side of the Boltzmann equation were zero, in 1-dimension (2-phase dimensions), it would read:

$$\frac{df}{dt} = 0 \quad \dots(9)$$

This would mean that the distribution function is unchanging (in its own frame). If particles were not interacting and there was no external disturbance, this is exactly what would happen. This is why with the expression is set to zero, the Boltzmann equation is said to be collision less. Adding a non-zero expression would account for interactions between the particles or the effect of external disturbances.

### 3.Fluid model and Magnetohydrodynamics

The fluid model and the MHD approach, describe plasma by coupling Maxwell's equations to the fluid equations. This can also be obtained by taking moments of the Boltzmann equation. Charged particles in motion generate currents and therefore themselves create electric and magnetic fields which have an effect on other charged particles.

. In situations where the fields created by the plasma is 2 much more stronger than or competes with externally applied fields; like in astrophysical plasma or nuclear fusion, the MHD approach is very important to study Such a procedure gives rise to:

The continuity equation:

$$\frac{\partial \varrho}{\partial t} + \nabla \cdot (\varrho \mathbf{v}) = 0 \quad \boxed{\dots(10)}$$

and the Cauchy momentum equation:

$$\varrho \left( \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \right) \mathbf{v} = \mathbf{J} \times \mathbf{B} - \nabla p \quad \boxed{\dots(11)}$$

Higher moments yield other equations for quantities like the entropy. The fluid equations average over the velocity distribution of the particles to obtain the macroscopic variables like Pressure and Temperature, as discussed earlier- by taking moments of the Kinetic equations like the Boltzmann equation or the Vlasov equation.

#### 4. The model used in the project

In the context of plasma processes like Magnetron Sputtering where particles have a considerably large mean free path, we have decided not to use the fluid model and the MHD descriptions of a plasma. In such processes, it is almost always the case that the strength applied by the apparatus is far stronger than those generated by charged particles in a plasma. This approximation allows us to avoid solving Maxwell's equations and use only the Lorentz force. The model used in the project is mostly similar to the single particle model, where in a collection of particles, each particle evolves under the influence of the electric and magnetic fields set up by the apparatus; governed by the Lorentz force. However, we incorporate a little bit of the Kinetic theory in that we are interested in different initial velocity distributions for particles in the plasma and how the velocity distribution changes over time; as it is important to understand the velocity distribution to characterize the reflection and loss of particles in a magnetic-mirror like trap that may be setup in the plasma chamber

### 1.3 KNOWLEDGE GAINED FROM LITERATURE

The knowledge we gained from the literature can be summarized in the following points:

1. Understanding the plasma behavior in surface engineering techniques can help better use and operate plasma devices. Using simulation studies allows one to make various studies while not having to run a plasma device a lot of times.
2. Magnetic mirror effect is caused by a certain configuration of the magnetic field that allows particles to be trapped under the influence of the magnetic field. We weren't able to use the configuration used in a magnetic mirror, however we demonstrated that plasma can be trapped by controlling the magnetic field; especially in studies 2 and 3.
3. Plasma is usually studied with 3 approaches: single particle model, kinetic theory and, fluid model and MHD.
4. Most plasma simulations use Particle In Cell methods to run the simulation.
5. Boris Algorithm uses only the Lorentz force; following the single particle approach, and hence is easy to setup while being suitable for simple studies.
6. Plasma simulation in research are often aimed at studying special plasma systems like Astrophysical plasma or plasma in a Nuclear Fusion chamber, which require one to solve Maxwell's equations coupled with either the Lorentz force or the fluid equations; which is computationally very difficult to setup and run.
7. Using the Boris Algorithm provides easy way to set up a simple simulation that can help perform interesting studies.
8. A simulation of plasma as particles consists of defining particles and updating their positions and velocities. To update the positions and the velocities, one can define electric and magnetic fields that cause charged particles to accelerate. This can be done with the Lorentz force, using the Boris Algorithm.
9. Many simple studies set the initial speeds and positions of the particles in a simulation to zero. However, if this can be done with a procedure like sampling from statistical distributions, properties of the plasma like its temperature can mimic the corresponding properties of a real plasma system.
10. If the strength of the externally applied fields is much greater than those produced by the plasma, it is a good approximation to assume that the particles do not interact. This saves one from having to solve Maxwell's equations for a simple simulation.

## 1.4 GAPS IDENTIFIED

It would take a number of courses on plasma physics to fully understand the current research methods in plasma physics. In the interest of time, we have studied some simple setups; so that we can setup well functioning plasma configuration during the course of the project and study some aspects that we are interested in. The forefront of research in plasma physics concerns complicated devices like Magnetic Confinement Fusion, or Quantum Optic systems. Most research in surface engineering focuses on the properties of coatings obtained and parameters of plasma used; in processes that use plasma. In our project, we would like to set up a simple plasma simulation that can help us study smaller devices like the one available in the School of Mechanical Engineering; where we can control a small flux of particles by tuning the electric and magnetic fields. This section describes, how we construct a simple easy to use, plasma simulation system; which is yet to be functional.

### Particle in Cell Methods

Particle in a cell methods are used to simulate the kinetic theory of plasma. A simple strategy used for particle in cell plasma simulation based on strategies as outlined in the paper [6] and the slides [7] involves the following steps:

#### Sampling and Initialization

The initial positions and velocities of particles in the plasma are sampled from a distribution, or based on some strategy.

#### Action of fields on the particles

The particles move as described by the Lorentz force as stated in the equation (1).

#### Particle deposition

In this step, charged particles are deposited on the grid defined by the mesh, and the charge density  $\rho_i$  and the current density  $j_i$  generated by the deposited particles is computed. One strategy outlined in the paper [6] defines charge deposition as following.  $\mathbf{x}_i = (\mathbf{i} + 1/2) \Delta \mathbf{x}$ ,  $\mathbf{i} \in \mathbb{Z}^D$  define the grid. A second order deposition can be achieved by:

$$\rho_i = \sum_p \left( \frac{q_p}{V_i} \right) W_2 \left( \frac{\mathbf{x}_i - \mathbf{x}_p}{\Delta \mathbf{x}} \right) \quad \dots(12)$$

where  $\mathbf{V}_i = \Delta \mathbf{x}^D$  is the volume of the cell  $i$  and  $\mathbf{W}_2(\mathbf{x})$  is a  $D$ -dimensional interpolating function defined in [6]. In simple models, the current density  $j_i$  is often not used.

### **Fields generated by particles**

In this step, charge density and current density are computed. The paper [6] uses Poisson equation to compute the electric field generated by the charge distribution and neglects the magnetic field generated. However, in high performance simulations like that outlined in [7], the full set of Maxwell's equations are used to compute the electric and magnetic fields generated by the particles.

### **Force on particles**

In this step, the force on the particles due to the fields are computed. Most simulations like the one outlined in the paper [6]; because they compute the fields generated by the deposited particles, describe the interaction of particle with the fields generated by other particles in the plasma, and hence capture the particle-particle dynamics.

### **Action of the force on particles**

Step 2 is repeated to move the particles under the influence of the electric and magnetic fields.

## **1.5 OBJECTIVE**

The objectives of the project define the structure of our algorithm and are as follows:

### **1. Single Particle Method**

To simulate charged particles that evolve under fields; as governed by the Lorentz force.

### **2. Field Configurations**

To simulate a few fields- some describing apparatus like coils; some describing analytic expressions for fields and to study the different evolution of particles.

### **3. Kinetic Theory**

To study different initial velocity distributions and how the velocity distribution of particles changes as the particles dynamically evolve. Parameters like the Plasma temperature are to be studied under this topic.

### **4. Analysis**

To analyze different batches or collections of particles, subjected to different field configurations.

## **1.6 MOTIVATION**

Currently large number of researches going on to use of plasma in daily practical life, and as the theme of our project lies on using magnetic field and electric field to study the efficiency of magnetic sputtering, it gave us the motivation to choose the topic for our thesis.

## **2.1 Project Execution Stages**

### **Expected Timeline**

#### **A. Schedule and Tasks:**

##### **1. Weeks 1-2 Study: plasma models, kinetic theory**

We first studied basic plasma physics, and a bit more on the kinetic theory of plasma as that seemed relevant to our project.

##### **2. Weeks 2-3 Study: PIC, Boris Algorithm**

We then studied particle in cell methods and then the Boris Algorithm.

##### **3. Weeks 3-4 Algorithm: idea, constants**

With some idea of how to proceed with the project, we started working on the algorithm. We started by with work on the constants.ipynb notebook, defining the constants that might be needed for the project.

##### **4. Weeks 4-5 Algorithm: particle, field, step**

We then worked on the notebook particle.ipynb where we defined a particle and how to update it. In the field.ipynb file, we worked to define some basic field configurations. Then we worked on step.ipynb to define steps of evolutions of the particles in a field configuration. This is the basic work on single particle dynamics.

##### **5. Weeks 5-6 Algorithm: sampling, Review preparation**

We then worked on some kinetic theory aspects of the project, defining some sampling strategies based on the Uniform velocity and Maxwellian distribution. We then started preparing for review 1

### **Review 2**

##### **1. Weeks 7-8 Algorithm: run, plot**

We plan to work next on run.ipynb to define running of batches of particle evolution and on plot.ipynb to define plot functionalities.

## 2. Weeks 8-9 **Algorithm: changes and fixes**

We anticipate that there may be errors in the programs that we will need some time to debug.

## 3. Weeks 9-10 **First plasma simulation**

We then expect to have the first plasma simulation running.

## 4. Weeks 10-11 **First results and study**

Running some plasma simulations, we intend to study the first set of results. 5. Weeks 11-12

## **Review preparation**

We plan to prepare for review 2 based on some preliminary results and understanding.

Review 3

## 1. Weeks 13-14 **Definitions: fields, density functions**

We then intend to make some new definitions like sampling based on the parabolic density functions, some new field configurations and possibly some new functionality handling particle states.

## 2. Weeks 14-15 **Fixes and study**

We expect to spend some time making final fixes and changes on the algorithm. We then turn to studying the results.

## 3. Weeks 15-16 **Study and result analysis**

We intend to spend some time studying the results and trying to understand the plasma model in the simulation.

## 4. Weeks 16-18 **Report and review preparation**

We then expect to start preparing for review 3, and preparation of the report and other documents.

## **B. Milestones:**

### **Milestone 1 : Algorithm start *Week 3 - Achieved***

Around week 3 we had some understanding of basic plasma physics and so started working on how to formulate the problem as an algorithm. Before review 1

### **Milestone 2 :Simulation running *Week 9 - Expected***

We expect to have the simulation running around week 9 of the project. This will allow us to have some preliminary results and understanding of our model, by review 2. Before review 2

### **Milestone 3: Study and understanding *Week 15 - Expected***

We intend to perform study on plasma using our algorithm and improve our understanding of a plasma and how to control and study a plasma system before the end of the project. Before review 3

## **Revised Timeline:**

### **A. Schedule and Tasks:**

#### **1. Weeks 1-2 Study: plasma models, kinetic theory**

We first studied basic plasma physics, and a bit more on the kinetic theory of plasma as that seemed relevant to our project.

#### **2. Weeks 2-3 Study: PIC, Boris Algorithm**

We then studied particle in cell methods and then the Boris Algorithm.

#### **3. Weeks 3-4 Algorithm: idea, constants**

With some idea of how to proceed with the project, we started working on the algorithm. We started by with work on the constants.ipynb notebook, defining the constants that might be needed for the project.

#### **4. Weeks 4-5 Algorithm: particle, field, step**

We then worked on the notebook particle.ipynb where we defined a particle and how to update it. In the field.ipynb file, we worked to define some basic field configurations. Then we worked on step.ipynb to define steps of evolutions of the particles in a field configuration. This is the basic work on single particle dynamics.

#### **5. Weeks 5-6 Algorithm: sampling, Review preparation**

We then worked on some kinetic theory aspects of the project, defining some sampling strategies based on the Uniform velocity and Maxwellian distribution. We then started preparing for review 1

## **Review 2**

#### **1. Weeks 7-8 Algorithm: run, changes and fixes**

We plan to work next on run.ipynb to define running of batches of particle evolution and on plot.ipynb to define plot functionalities.

#### **2. Weeks 8-9 Algorithm: changes and fixes**

We anticipate that there may be errors in the programs that we will need some time to debug.

#### **3. Weeks 9-10 First plasma simulation and study 1**

After the first plasma simulation was running we were able to start working on Study 1.

#### **4. Weeks 10-11 Study 1 and review preparation**

We prepared everything required to complete Study 1.

## 5. Weeks 11-12 **Review preparation**

We prepared for review 2 based on Study 1 results and understanding.

### **Review 3**

#### 1. Weeks 13-14 **Study 2**

We were able to complete all the due work for study 2 .

#### 2. Weeks 14-15 **Study 3**

We successfully completed all the assignments of Study 3 for the project.

#### 3. Weeks 15-17 **Review 3 preparation**

We prepared some documents for the final review

### **B. Milestones:**

#### **Milestone 1 : Algorithm start *Week 3 - Achieved***

Around week 3 we had some understanding of basic plasma physics and so started working on how to formulate the problem as an algorithm. Before review 1

#### **Milestone 2 :Simulation running *Week 9 - Expected***

We had the simulation running around week 9 of the project. This allowed us to have some preliminary results and understanding of our model, by review 2.

#### **Milestone 3: Study and understanding *Week 15 - Expected***

We performed study on plasma using our algorithm and improve our understanding of a plasma and how to control and study a plasma system before the end of the project. Before review 3

## **2.2 Technical Specifications**

The simulation is programmed in jupyter notebook files (.ipynb extensions) that run python and markdown. The notebooks are available in the following Github repository, in the notebooks folder.

### **Github Repository**

<https://github.com/18BME2104/MagneticMirror>

### **Algorithm Outline**

Different aspects of the program are discussed based on the different notebooks (ipynb files) that describe them.

Required functionalities and Files:

1. Constants - **constants.ipynb**
2. Particle - **particle.ipynb**
3. Electric and Magnetic fields - **field.ipynb**
4. Particle initialization - **sampling.ipynb**
5. Updating the particles - **step.ipynb**
6. Batches of updates - **run.ipynb**
7. Plotting - **plot.ipynb**

### Constants - constants.ipynb

The **constants.ipynb** notebook describes some constants useful in the program. Some useful constants are  $e$  (electron charge)  $m_e$  (electron mass), charges and masses of ions in the plasma, a.m.u (atomic mass unit),  $N_A$  (Avogadro's number),  $\epsilon_0$  (permittivity of vacuum),  $\mu_0$  (permeability of vacuum),  $K$  or  $k_B$  (Boltzmann's constant), etc.

### Particle - particle.ipynb

The **particle.ipynb** notebook describes the state of a particle; its position, velocity, mass, charge, name, and optionally acceleration (which is set to 0 as default if it is not required to track the acceleration of a particle) as of now.

Currently the Boris algorithm as discussed earlier, is an update strategy defined to update the state of a particle. Other strategies could be defined in new functions in the class. However, Boris algorithm is good enough for us to get started.

### Electric and Magnetic fields - field.ipynb

Electric and Magnetic field configurations described in **field.ipynb**.

Currently **Uniform Electric field** and the **Radial Electric field** (the field depends on the particle's position) created by an electrode are available to set up electric fields. Uniform Magnetic field and Magnetic field created by a Helmholtz coil and that by two Helmholtz coils are available. Other Electric and Magnetic fields can be defined as functions in this class. These fields could be based on modeling of apparatus used to create electric or magnetic fields such as coils, or based on analytic expressions

## Particle initialization - `sampling.ipynb`

The initial positions and velocities of the particles evolution of their state under the influence of electric and magnetic fields. The initial distribution of positions and velocities define where the particles start in the setup (or lab apparatus); for example where they are injected into a sputtering chamber through valves, and what velocities they start with; for example what potential they are accelerated through or what parameters were used for the pumps used to pump the particles in.

The tracking of initial distribution is also important in determining how the magnetic mirror effect is observed in the plasma. The Sampler class samples initial positions and velocities for a given number of particles based on some available schemes.

Currently positions can be sampled such that all particles start at the **same position** (like for example if injected through a port), **at the same distance** but randomly distributed in angular positions relative to some point(origin as of now). Velocities can be sampled such that particles have the **same speed**, **same velocity** or **Maxwellian distributed speeds** or **Maxwellian distributed velocities**. The initial positions and velocities could be passed around in lists but to be able to reuse some generated samples, and avoid sampling all the time, it is convenient to write sampled positions and velocities to files (csv format seems convenient). This functionality is described as:

## Updating the particles - `step.ipynb`

The Step class is concerned with **initializing particles**, **defining the fields**, and **updating the states of the particles** (positions and velocities) .

To use sampled positions and velocities that have been saved in csv files, some reading functionality is useful to load these positions and velocities to be used during initialization.

## Batches of updates - `run.ipynb`

The Run class is concerned with running the steps defined by the Step class in `step.ipynb`. This includes creating batches of particles, updating them and removing them if they move out of the region of interest or for example are absorbed during a coating process. New batches of particles can be created to model the flow of particles as in a plasma chamber setup. Functionality to change the fields; for example changing the Voltages in electrodes or Currents in the coils, are also defined. This better models the control of plasma supply and electric and magnetic field control apparatus as in a laboratory.

## 2.3 Design Approach Details

### Design Approach / Materials & Methods

As discussed earlier, we evolve individual particles in the plasma with the Lorentz force. To achieve this in a simulation, the Lorentz force equations are discretized and then solved using the Boris Algorithm; a standard algorithm for simulating particles .

#### Lorentz Force

The equations of motion for a charged particle is described by the Lorentz Force in the S.I. units as

$$\frac{d\mathbf{v}}{dt} = \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad \dots(13)$$

as discussed earlier, along with the expression for the velocity.

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \quad \dots(14)$$

These equations are discretized to obtain

$$\frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{\Delta t} = \frac{q}{m} \left[ \mathbf{E}_k + \frac{(\mathbf{v}_{k+1} + \mathbf{v}_k)}{2} \times \mathbf{B}_k \right] \quad \dots(15)$$

from the Lorentz Force equation, and

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} = \mathbf{v}_{k+1} \quad \dots(16)$$

from the expression for velocity, where the subscript k denotes the k<sup>th</sup> time step.

#### Boris Algorithm

The discretized Lorentz equations may be solved using the Boris Algorithm, as we do in this project. The Boris Algorithm splits the discretized Lorentz equation into three equations.

$$\frac{\mathbf{v}^- - \mathbf{v}_k}{(\Delta t/2)} = \frac{q}{m} \mathbf{E}_k \quad \text{or} \quad \frac{\mathbf{v}^- - \mathbf{v}_k}{\Delta t} = \frac{1}{2} \frac{q}{m} \mathbf{E}_k \quad \dots(17)$$

which is often called the first half of the electric pulse.

$$\frac{\mathbf{v}^+ - \mathbf{v}^-}{\Delta t} = \frac{q}{m} \left( \frac{\mathbf{v}^+ + \mathbf{v}^-}{2} \right) \mathbf{B}_k \quad \boxed{\dots(18)}$$

which is often called rotation by the magnetic field.

$$\frac{\mathbf{v}_{k+1} - \mathbf{v}^+}{(\Delta t/2)} = \frac{q}{m} \mathbf{E}_k \quad or \quad \frac{\mathbf{v}_{k+1} - \mathbf{v}^+}{\Delta t} = \frac{1}{2} \frac{q}{m} \mathbf{E}_k \quad \boxed{\dots(19)}$$

which is almost the discretized Lorentz equation except that  $(\mathbf{v}^+ + \mathbf{v}^-)$  is substituted for  $(\mathbf{v}_{k+1} + \mathbf{v}_k)$ . However, subtracting the first electric pulse equation from equation gives  $(\mathbf{v}^+ + \mathbf{v}^-) = (\mathbf{v}_{k+1} + \mathbf{v}_k)$ , giving the discretized Lorentz equation. This means that the Boris algorithm is equivalent to the discretized Lorentz equation.

The equations of the Boris Algorithm can be written slightly different as

along with discretized position equation where  $q' = \frac{q}{m} \frac{\Delta t}{2}$ . These equations are used to update the velocity of the particle under the influence of the Lorentz force under the Boris update strategy.

### Particle Sampling- kinetic theory

As discussed earlier, kinetic theory of plasma uses density function and its evolution to describe a plasma. We pointed out that we will be using some aspects of kinetic theory in our project; in that we will initialize the batches (collections) of particles based on certain distributions. This will allow us to study parameters like the plasma temperature, in our project

### Maxwellian distribution

One important density function often used in the kinetic theory of plasma is the Maxwell-Boltzmann distribution often called the Maxwellian which has the density function

$$\widehat{f}_M := \hat{f}(\mathbf{x}, \mathbf{v}, t) = \left( \frac{m}{2\pi KT} \right)^{\frac{3}{2}} \exp \left( -\frac{v^2}{v_{th}^2} \right) \quad \boxed{\dots(20)}$$

Where

$$v_{th}^2 = \frac{2KT}{m} \quad \dots(21)$$

Some features of the Maxwellian are:

$$v_{rms} = \sqrt{\frac{3KT}{m}}, |\bar{v}| = 2\sqrt{\frac{2KT}{\pi m}}, |\bar{v}_z| = \sqrt{\frac{2KT}{\pi m}}, \bar{v}_z = 0 \quad \dots(22)$$

## 2.4 Realistic and Design Constraints addressed

Some constraints of the project can be described in terms of the limitations and decisions based on those limitations:

1. Solving Maxwell's equations: We did not solve the Maxwell's equations to calculate the fields generated by the particles in the plasma and that the applied fields are much stronger than the fields produced by the plasma, it is reasonable to assume that the particles generate no electric and magnetic fields and hence do not interact. However, for example if one is interested in studying a plasma over a longer period of time, or to study the behavior of plasma when the fields are turned off; this would be important.
2. Particle interactions: As in the previous point, particles are assumed not to interact. However, one could use several approximate terms to account for particle interactions without having to solve Maxwell's equations. Including this aspect would require a bit more time of studying such models and the implementing them.
3. Computational study: Due to the limitation of time, we weren't able to perform computational studies. The first thing one could study is how the plasma temperature and hence the initial velocities would affect the behavior of the plasma.
4. Running the simulation versus understanding the behavior: If we were more confident with how a plasma works mathematically; for example with a greater knowledge of solving the equations involved, we could have spent more time running the simulation and less time having to understand the plots and the behavior of the plasma. Even though it might be easier to run a plasma simulation, it takes time to understand the results.

5. Precision and theoretical calculations: The calculations done in the plasma are done with over 10 digits of precision. However, these purely theoretical calculations are not able to completely capture the dynamics of a plasma system until more precise models that solve Maxwell's equations are used.

6. Big picture of such a project: The vision for such a project, which would take us substantially long time would be to have a plasma simulation running real time in the cloud such that the user can change the fields, particle distributions and other aspects like interactions; if any, so that one can perform hours of studies on simulations and therefore be better able to use a plasma device.
7. Number of particles: We only used 100 particles in the plasma. However if one could let it run for hours on a slightly better computer than a personal computer, one could simulate several thousands of particles easily with the setup that's already built in this project.
8. Duration of simulation: We only used a few 100 steps of updates in the studies. However, several thousands steps of simulation could be run on a more advanced computer with the setup that's already built. Using more steps of updates allow one to vary the fields slowly and observe behavior in the larger time scales which we weren't able to.
9. New fields and distributions: We were only able to use 3 field configurations and 2 initial distributions. One could use 10s of different field configurations and distributions based on other studies in research. For example we only used uniform fields and changed them in epochs. Non uniform fields mimic fields of real coils, or multiple coils could be better.
10. Clever Algorithms: In our program we store lists of positions and velocities that we update one by one going through the list. Using clever algorithm for example, doing computations with matrices might significantly reduce the simulation times and might have allowed us to run simulations for much longer. However, having only done 3 studies at the end, we weren't ready to have more interesting discussions on the studies even if we ran them for longer.
11. Interaction with component in surface engineering processes: It would be interesting to have plasma interact with components used in surface engineering processes. The first thing to do towards that would be to create options for particle absorption and ejection from the chamber. We decided to work on studies 2 and 3 instead of working on this option and it seems like a good decision for our project.

## 2.5 Codes and Standards

### Particle - particle.ipynb

Currently the Boris algorithm as discussed earlier, is an update strategy defined to update the state of a particle.

Other strategies could be defined in new functions in the class.

```
1  class Particle:
2      #... other things
3
4      def Boris_update(self, afield, argsE, argsB):
5          # Define q_prime
6          q_prime = (self.charge / self.mass) * (dt / 2)
7
8          # Get E and B fields from the afield argument by passing in the
          # current position of the particle
9          argsE = V, center
10         E = afield.get_E_field(self.r, V, center)
11         argsB = n, I, R, B_hat, mu_0
12         B = afield.get_B_field(self.r, n, I, R, B_hat, mu_0)
13
14         #Boris velocity update
15         v_minus = self.v + q_prime * E
16         v_plus = v_minus + q_prime * 2 * np.cross(v_minus, B)
17         v_new = v_plus + q_prime * E
18
19         self.v = v_new
20
21
22         #could have also done:
23         #self.v += (2 * q_prime) * (E + np.cross( (self.v + q_prime * E),
            B))
24
25         #update position
26         self.r += v_new * dt
27
28     #... some other things
```

### Electric and Magnetic fields - field.ipynb

It can be defined as functions in this class. These fields could be based on modeling of apparatus used to create electric or magnetic fields such as coils, or based on analytic expressions.

```

1  class Field:
2      #... something
3
4      def uniform_E_field(self , E):
5          return E
6
7      def radial_E_field(self , r , V, center = [0,0,0]):
8          #Get the distance vector of the particle from the electrode
9          dr = [(r[0] - center[0]), (r[1] - center[1]), 0]
10         #Get the electric field
11         E = V * dr
12         return E
13
14     def uniform_B_field(self , B):
15         return B
16
17     def helmholtz_coil_B_field(self , n, I, R, B_hat, mu_0):
18         return ( (4/5)**1.5 * ( (mu_0 * n * I) / (R) ) * B_hat)
19
20     def two_helmholtz_B_field(self , n1, I1, R1, B1_hat, n2, I2, R2, B2_hat
21         , mu_0):
22         B1 = helmholtz(self , n1, I1, R1, mu_0, B1_hat)
23         B2 = helmholtz(self , n2, I2, R2, mu_0, B2_hat)
24
25         #Calculate the resultant of two magnetic fields
26         B_hat = B1_hat + B2_hat
27         return B_hat
28     #... something else

```

### Particle initialization - sampling.ipynb

Positions can be sampled such that all particles start at the same position (like for example if injected through a port), at the same distance but randomly distributed in angular positions relative to some point(origin as of now). Velocities can be sampled such that particles have the same speed, same velocity or Maxwellian distributed speeds or Maxwellian distributed velocities

```

1  class Sampler:
2      #... something
3
4      def sample_same_given_position(self , r , n):
5          positions = []
6          for i in range(n):
7              positions.append(r)
8
9          return np.array(positions)
10
11     def sample_same_given_distance_all_random_direction(self , d , n):
12         positions = []
13
14         for i in range(n):
15             positions.append(d * uniform_random_unit_vector())
16
17         return np.array(positions)
18
19     def sample_same_given_velocity_same_direction(self , v , n):
20         velocities = []
21         for i in range(n):
22             velocities.append(v)
23
24         return np.array(velocities)
25
26     def sample_same_given_speed_all_random_direction(self , s , n):
27         velocities = []
28
29         for i in range(n):
30             velocities.append(s * uniform_random_unit_vector())
31
32         return np.array(velocities)
33
34     def sample_velocity_uniformKE_same_given_direction(self):
35         pass
36
37     def sample_velocity_uniformKE_all_random_directions(self , n):
38         pass
39
40     def sample_Maxwellian_speed(self , v_median , K , T , m , n):
41         alpha = math.sqrt(K * T / m)
42         speeds = stats.maxwell.rvs(loc = v_median , scale = alpha , size = n
43             )
44         return speeds
45
46     def sample_Maxwellian_velocity_same_given_direction(self , v_median , K ,
47         T , m , v_hat , n):
48         speeds = sample_Maxwellian_speed(self , v_median , K , T , m , n)
49         velocities = np.outer(speeds , v_hat)
50
51         return np.array(velocities)

```

```

50
51     def sample_Maxwellian_velocity_same_random_direction(self, v_median, K
    , T, m, n):
52         speeds = sample_Maxwellian_speed(self, v_median, K, T, m, n)
53         direction = self.uniform_random_unit_vector()
54         velocities = np.outer(speeds, direction)
55
56         return np.array(velocities)
57
58     def sample_Maxwellian_velocity_all_random_direction(self, v_median, K,
    T, m, n):
59         speeds = sample_Maxwellian_speed(self, v_median, K, T, m, n)
60         velocities = []
61         for i in range(n):
62             velocities.append(speeds[i] * np.array(
                uniform_random_unit_vector()))
63
64         return np.array(velocities)
65
66     def sample_parabolic_velocity(self):
67         pass
68     #... something else

```

The initial positions and velocities could be passed around in lists but to be able to reuse some generated samples, and avoid sampling all the time, it is convenient to write sampled positions and velocities to files (csv format seems convenient). This functionality is described as:

```

1  class Sampler:
2      #... something
3      def write_to_csv_file(self, ...):
4
5          # write array to csv file
6
7      def write_file_name(self, ...):
8
9          # write the file name to the list of available files
10
11     #... something else

```

### Updating the particles - step.ipynb

The Step class is concerned with initializing particles, defining the fields, and updating the states of the particles (positions and velocities) .

```

1  class Step:
2
3      # ... something
4
5      ### PARTICELS INITIALIZATION SECTION
6      def initialize_particles(self, names, q_s, m_s, r_0_s, v_0_s, a_0_s, n
7          ):
8          for i in range(n):
9              self.particles.append(Particle(names[i], q_s[i], m_s[i], r_0_s
10                 [i], v_0_s[i], a_0_s[i] ))
11
12      ### FIELDS INITIALIZATION SECTION
13      def initialize_fields(self):
14          self.fields = Field()
15
16      ### TIME STEP SECTION
17      def update_particles(self, dt, argsE, argsB):
18          for particle in self.particles:
19              particle.update(self.fields, dt, argsE, argsB)
20      # ... something else

```

To use sampled positions and velocities that have been saved in csv files, some reading functionality is useful to load these positions and velocities to be used during initialization.

```

1  class Step:
2
3      # ... something
4      def read_r_or_v_file_and_reshape(self, index):
5
6          array_from_file = self.read_r_or_v_file(index)
7          reshaped_array = self.resaper(array_from_file)
8          return reshaped_array
9
10     # ... something else

```

### 3. RESULTS AND DISCUSSION

#### 3.1 Checks

To verify that our program is working correctly, we perform some checks. We compare results from the program with calculations done by hand; of certain quantities and see if they agree. As of now, we are doing checks on the following aspects of the program:

1. Sampling ✓

1. Update ✓

## Sampling

As of now we our sampling of particle speeds is based on the Maxwell-Boltzmann distribution, that we have defined previously. We now check if the average speeds of the particles agrees with the calculations done by hand based on the plasma temperature. We recall the Maxwellian density function that we defined earlier.

$$\widehat{f}_M := \hat{f}(\mathbf{x}, \mathbf{v}, t) = \left( \frac{m}{2\pi KT} \right)^{\frac{3}{2}} \exp \left( -\frac{\mathbf{v}^2}{v_{th}^2} \right) \quad \dots(23)$$

where

$$v_{th}^2 = \frac{2KT}{m}$$

For our convenience, we use the density function with speed instead of velocity defined as:

$$\widehat{f}_m := \hat{f}(\mathbf{x}, v, t) = \left( \frac{m}{2\pi KT} \right)^{\frac{3}{2}} 4\pi v^2 \exp \left( -\frac{v^2}{v_{th}^2} \right) \quad \dots(24)$$

which is obtained by integrating over the solid angle in the velocity variable.

For this density function we calculate the average speed by with the expression:

$$\begin{aligned} \langle v \rangle &= \int_{v=-\infty}^{v=\infty} dv \, v \, \widehat{f}_m = \int_{v=-\infty}^{v=\infty} dv \, v \, \left( \frac{m}{2\pi KT} \right)^{\frac{3}{2}} 4\pi v^2 \exp \left( -\frac{v^2}{v_{th}^2} \right) \quad \dots(25) \\ &= 4\pi \left( \frac{m}{2\pi KT} \right)^{\frac{3}{2}} \int_{v=-\infty}^{v=\infty} dv \, v^3 \exp \left( -\frac{v^2}{v_{th}^2} \right) \\ &= 4\pi \left( \frac{m}{2\pi KT} \right)^{\frac{3}{2}} \frac{v_{th}^4}{2} = 4\pi \left( \frac{1}{\pi v_{th}^2} \right)^{\frac{3}{2}} \frac{v_{th}^4}{2} \\ &= \frac{2}{\sqrt{\pi}} v_{th} = \frac{2}{\sqrt{\pi}} \sqrt{\frac{2KT}{m}} = \frac{2}{\sqrt{\pi}} \sqrt{\frac{2RT}{M}} \end{aligned}$$

For the Hydrogen atom particle distribution, we get

$$\langle v \rangle = \frac{2}{\sqrt{\pi}} \sqrt{\frac{2 \cdot 8.31446261815324 \text{ J K}^{-1} \text{ mol}^{-1} \cdot 10000 \text{ K}}{1.008 \times 10^{-3} \text{ kg mol}^{-1}}} = 14492.952993825973 \text{ ms}^{-1}$$

From section 1.1.1 Maxwellian sampling in the **checks.ipynb** notebook, we take the following:

### a. Initialization

### b. Inspection

Upon inspection we get an average speed of distribution to be:

$$14202.764572898674 \text{ ms}^{-1}$$

We see that the mean average speed of the sampled distribution and that expected from the analytic expression are close; in fact within 2.0431826454479785% error.

To be confident that our program indeed samples particles based on Maxwellian distribution as we expect it to, let's check another distribution. For convenience, let's assume Hydrogen molecules to be particles sampled with Maxwellian distribution, and check if the average speed of the sampled distribution agrees with the that expected from analytic calculation. For this we follow a similar procedure.

Again upon inspection we get an average speed of the distribution to be:

$$10149.754879907316 \text{ ms}^{-1}$$

For the Hydrogen molecule particle distribution, we get

$$\langle v \rangle = \frac{2}{\sqrt{\pi}} \sqrt{\frac{2 \cdot 8.31446261815324 \text{ J K}^{-1} \text{ mol}^{-1} 10000 \text{ K}}{2 \times 1.008 \times 10^{-3} \text{ kg mol}^{-1}}} = 10248.06534135222 \text{ ms}^{-1}$$

We see that the mean average speed of the sampled distribution and that expected from the analytic expression are close for this distribution too; in fact, within 0.9685993662716086% error. Because the errors are considerably small compared to statistical variation in the sampling, we believe the Maxwellian distribution sampling part of the program to be working as we expect it to.

### Update

We recall the Boris Algorithm, that we use to update the particles in the plasma simulation.

$$\begin{aligned} \mathbf{v}^- &= \mathbf{v}_k + q^j \mathbf{E}_k \\ \mathbf{v}^{+-} &= \mathbf{v}^- + 2q'(\mathbf{v}^- \times \mathbf{B}_k) \\ \mathbf{v}_{k+1} &= \mathbf{v}^{+-} + q^j \mathbf{E}_k \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta t \mathbf{v}_{k+1} \end{aligned} \quad \boxed{\dots(26)}$$

where  $q' = \frac{q}{m} \frac{\Delta t}{2}$ .

To verify that the simulation works as we expect it to, we perform a calculation and compare it to the output of an algorithm. From section 1.2.1 Boris Update in the **checks.ipynb** notebook, we take the following:

The program gives the following output. For example, we can check the particles 1 and 7 in the update. After the 3rd step of the update, the velocity of particle 1 was:

$$[39771.83801956555, -5029.491038, 4533.343932509505]$$

and after the 4th update, it changed to:

$$[48909.86581773698, -5029.491038, 8798.399243869582]$$

Its position after the 3rd step of the update was:

$$[-0.48985112694809785, -0.0020117964152, 0.00017005183797547688]$$

which was updated after the 4th step of the update to:

$$[-0.4849601403663242, -0.0025147455189999997, 0.001049891762362435]$$

We now check if the same is true, with a calculation of the update done by hand.

$$q' = \frac{q}{m} \frac{\Delta t}{2} = \frac{1.602176634 \times 10^{-19} \text{ C}}{1.008 \text{ a.m.u.} \times 1.6605390666 \times 10^{-27} \text{ kg a.m.u.}^{-1}} \frac{10^{-7} \text{ s}}{2} = 4.78597877761177 \text{ C s kg}^{-1}$$

$$\mathbf{v}^- = \mathbf{v}_3 + q' \mathbf{E} = \begin{bmatrix} 39771.83801956555 \\ -5029.491038 \\ 4533.343932509505 \end{bmatrix} + 4.78597877761177 \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 44557.81679718 \\ -5029.491038 \\ 4533.34393251 \end{bmatrix}$$

...(27)

$$\begin{aligned} \mathbf{v}^+ = \mathbf{v}^- + 2q' (\mathbf{v}^- \times \mathbf{B}) &= \begin{bmatrix} 44557.81679718 \\ -5029.491038 \\ 4533.34393251 \end{bmatrix} + 2 \cdot 4.78597877761177 \left( \begin{bmatrix} 44557.81679718 \\ -5029.491038 \\ 4533.34393251 \end{bmatrix} \times \begin{bmatrix} 0.0 \\ 0.01 \\ 0.0 \end{bmatrix} \right) \\ &= \begin{bmatrix} 44123.88704013 \\ -5029.491038 \\ 8798.39924387 \end{bmatrix} \end{aligned}$$

$$\mathbf{v}_4 = \mathbf{v}^+ + q' \mathbf{E} = \begin{bmatrix} 44123.88704013 \\ -5029.491038 \\ 8798.39924387 \end{bmatrix} + 4.78597877761177 \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 48909.86581774 \\ -5029.491038 \\ 8798.39924387 \end{bmatrix}$$

$$\mathbf{x}_4 = \mathbf{x}_3 + \Delta t \mathbf{v}_4 = \begin{bmatrix} -4.89851127 \times 10^{-01} \\ -2.01179642 \times 10^{-03} \\ 1.70051838 \times 10^{-04} \end{bmatrix} + 10^{-7} \begin{bmatrix} 48909.86581774 \\ -5029.491038 \\ 8798.39924387 \end{bmatrix} = \begin{bmatrix} -0.48496014 \\ -0.00251475 \\ 0.00104989 \end{bmatrix}$$

We see that the values for  $x_3$ ,  $x_4$ ,  $v_3$  and  $v_4$  for particle 1 are close (the latter digits differ due to the differing precision of calculations); i.e. the same when done by hand and in the program.

Similarly, for particle 7. After the 3rd step of the update, the velocity of the particle was:

$$[38895.85871094631, -8670.854777, 13914.969423620274]$$

and after the 4th update, it changed to:

$$[47135.881299118584, -8670.854777, 18096.176367366777]$$

Its position after the 3rd step of the update was

$$[-0.48985112694809785, -0.0020117964152, 0.00017005183797547688]$$

which was updated after the 4th step of the update to:

$$[-0.4849601403663242, -0.0025147455189999997, 0.001049891762362435]$$

$$\begin{aligned} \mathbf{v}^- &= \mathbf{v}_3 + q'\mathbf{E} = \begin{bmatrix} 38895.85871095 \\ -8670.854777 \\ 13914.96942362 \end{bmatrix} + 4.78597877761177 \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 43681.83748856 \\ -8670.854777 \\ 13914.96942362 \end{bmatrix} \\ \mathbf{v}^+ &= \mathbf{v}^- + 2q'(\mathbf{v}^- \times \mathbf{B}) = \begin{bmatrix} 43681.83748856 \\ -8670.854777 \\ 13914.96942362 \end{bmatrix} + 2 \cdot 4.78597877761177 \left( \begin{bmatrix} 43681.83748856 \\ -8670.854777 \\ 13914.96942362 \end{bmatrix} \times \begin{bmatrix} 0.0 \\ 0.01 \\ 0.0 \end{bmatrix} \right) \\ &= \begin{bmatrix} 42349.90252151 \\ -8670.854777 \\ 18096.17636737 \end{bmatrix} \end{aligned} \quad \boxed{\dots(28)}$$

$$\mathbf{v}_4 = \mathbf{v}^+ + q'\mathbf{E} = \begin{bmatrix} 42349.90252151 \\ -8670.854777 \\ 18096.17636737 \end{bmatrix} + 4.78597877761177 \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 47135.88129912 \\ -8670.854777 \\ 18096.17636737 \end{bmatrix}$$

$$\mathbf{x}_4 = \mathbf{x}_3 + \Delta t \mathbf{v}_4 = \begin{bmatrix} -0.48966698 \\ -0.00346834 \\ 0.00388755 \end{bmatrix} + 10^{-7} \begin{bmatrix} 47135.88129912 \\ -8670.854777 \\ 18096.17636737 \end{bmatrix} = \begin{bmatrix} -0.48495339 \\ -0.00433543 \\ 0.00569717 \end{bmatrix}$$

We see that the values for  $x_3$ ,  $x_4$ ,  $v_3$  and  $v_4$  for particle 7 are close (the latter digits differ due to the differing precision of calculations); i.e. the same when done by hand and in the program.

Based on these calculations, we believe that our particle update works as expected; according to the Boris Algorithm, and this part of the program works correctly.

### 3.2 Study 1 Plasma Stream

We now create our first plasma system, where we can change certain parameters of the fields; so as to simulate controlling plasma in a chamber by changing the electric and magnetic fields as required. We discuss the program written in **study1.ipynb**.

#### Setup

After making the imports, we first create the system as in a batch of particle of a run object instance.

After creating a batch of particle, we take that batch and update it. Here we consider constant magnetic field of 10 mT along the y-axis [0,1,0] and changing electric field configurations of [0, 0, 1, -1, 2, -2, 3, -3, 4, -4, 5, -5]  $1000 \text{ Vm}^{-1}$  along the x-axis [1,0,0]. For a time duration of 0.1 ms we update the batch of particles under different configurations of the electric field, using time steps of 0.001 ms and get the positions and velocities of the particles during the history of updates.

#### Varying Electric field, Constant Magnetic field

The following data was used for the study:

- 100 particles of Hydrogen (relative atomic mass:  $1.008 \text{ g mol}^{-1}$ )
- duration of 1 step of update: 0.1 ms

#### Sampling of the Initial Positions and Velocities of the particles

In this study the following sampling (initialization) data was used:

- Speeds are sampled from a Maxwellian distribution with plasma temperature 10000 K.
- Velocity directions are sampled from uniform distribution.
- Positions are sampled such that all particles start at [-0.5, 0, 0] (A box  $1\text{m} \times 1\text{m} \times 1\text{m}$  from [-0.5, -0.5, -0.5] to [0.5, 0.5, 0.5] may be considered for reference)

The standard Maxwellian distribution was used for the initial speeds of the particles and all particles were initialized at the same position.

## Configurations of the Electric and Magnetic Fields

The following field configurations were used in the study:

- Magnetic field: 10 mT along the y-axis [0, 1, 0]
- Electric field  $1000 \text{ Vm}^{-1}$  scaled by [0, 0, 1, -1, 2, -2, 3, -3, 4, -4, 5, -5] for 100 steps each.

The positions of 1 and 10 particles in the simulation are plotted respectively in figures (1) and (2).

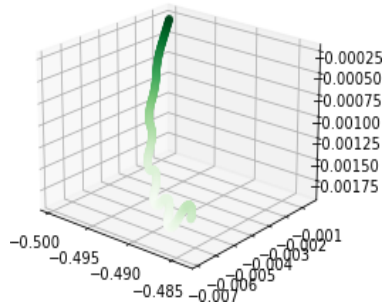


Figure 1: position of a particle

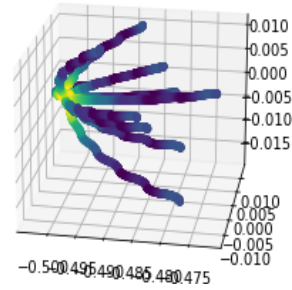


Figure 2: positions of 10 particles

The components of positions and velocities for 10 particles are also plotted

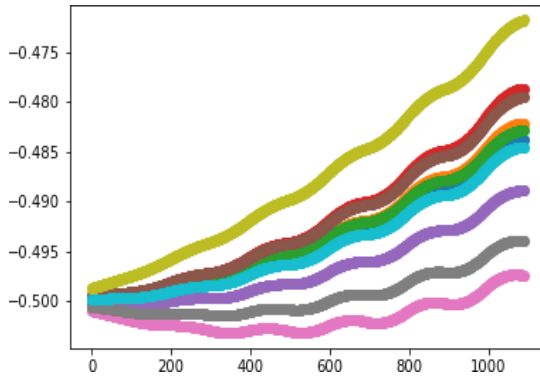


Figure 3: x-component of positions

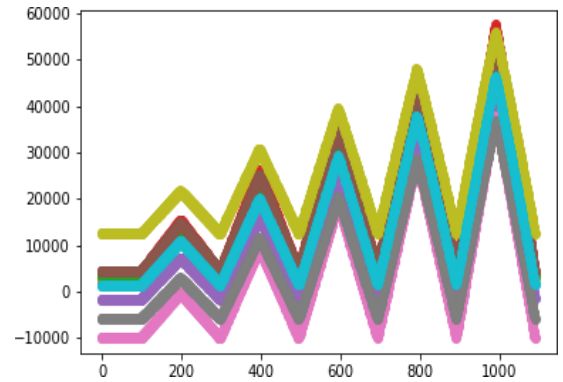


Figure 4: x-component of velocities

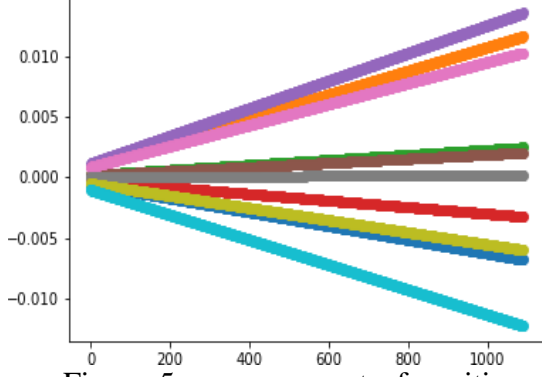


Figure 5: y-component of positions

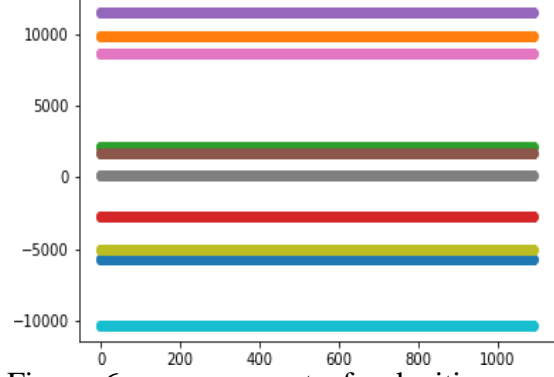


Figure 6: y-component of velocities

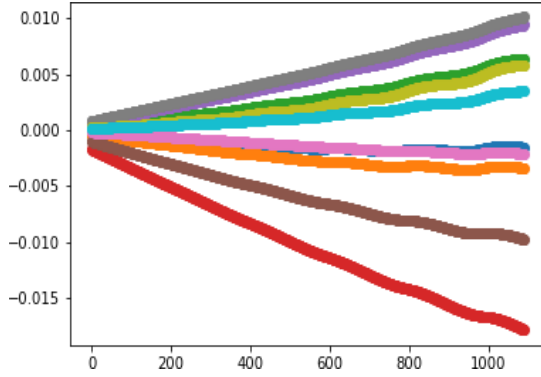


Figure 7: z-component of positions

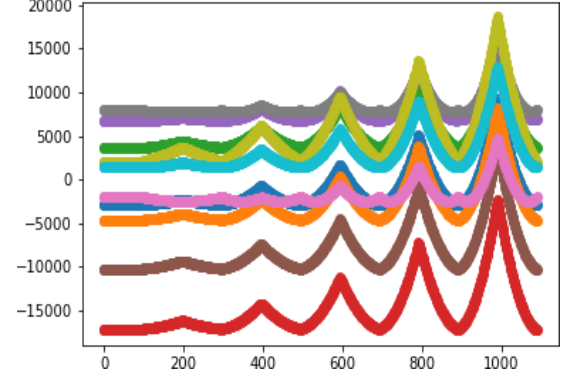


Figure 8: z-component of velocities

Figures (3), (5) and (7) show that the particles all start at  $[-0.5, 0, 0]$  as intended and spread out based on their initial velocities and velocities after subsequent updates. As the electric field is, the magnetic field, and according to the Lorentz force equation(1) the acceleration of a particle,  $\mathbf{a} \propto \mathbf{E} + (\mathbf{v} \times \mathbf{B})$ , there is non-zero acceleration along the  $x$ -axis, zero acceleration and non-zero acceleration along  $\mathbf{v} \times \mathbf{B}$  dependent on the velocity of a particle at a given instance. In figure (4), the increase and decrease in the  $x$ -component of velocities are marked by sharp changes after every 100 steps when the sign of the electric field was reversed; the effect of both  $\mathbf{E}$  and  $\mathbf{v} \times \mathbf{B}$  terms is observed in the acceleration. In fig(6), it is observed that the  $y$ -component of velocities stay the same as expected. In fig (8), the effect of  $\mathbf{v} \times \mathbf{B}$  term on the  $z$ -component of acceleration can be observed. The events of reversals of the electric field can hence also be observed.

### 3.3 Study 2- Maxwellian distributed speeds and Helmholtz coils

In this study the same sampling data as in study 1 was used.

#### Reversed Magnetic field

For the first part of this study, the following field configurations were used:

#### Fields Setup

- Magnetic field due to a Helmholtz coil (number of turns: 1000 in each coil, radius: 0.1m): for 100 steps each
- first 20A [0,0,1]
- second -20A [0,0,1]third 20A [0,0,1].
- Electric field constantly set to 0.
- The orientation of the coil was fixed in the  $z$ -direction and the direction of the current was changed while keeping the amplitude at a constant value of 20 A. The positions of 1 and 10 particles in the simulation are plotted respectively in figures (9) and (10).

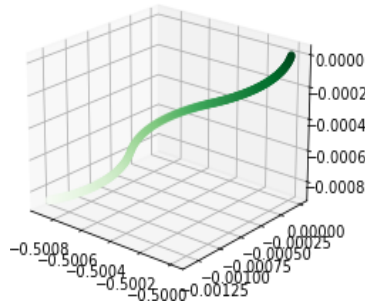


Figure 9: position of a particle

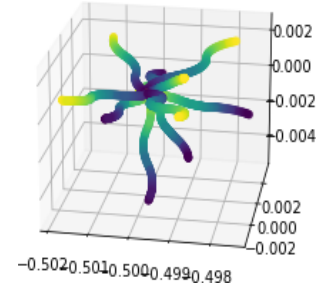


Figure 10: positions of 10 particles

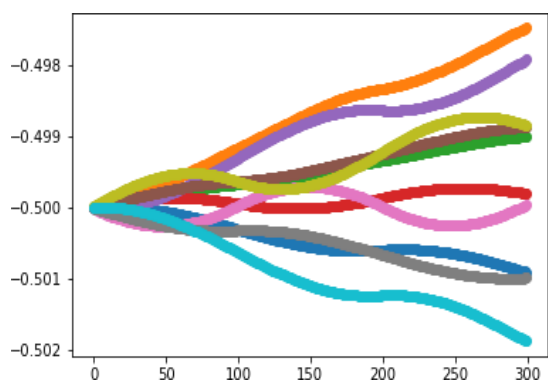


Figure 11:  $x$ -component of positions

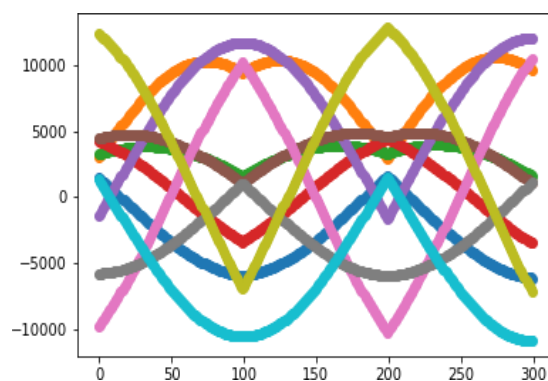


Figure 12:  $x$ -component of velocities

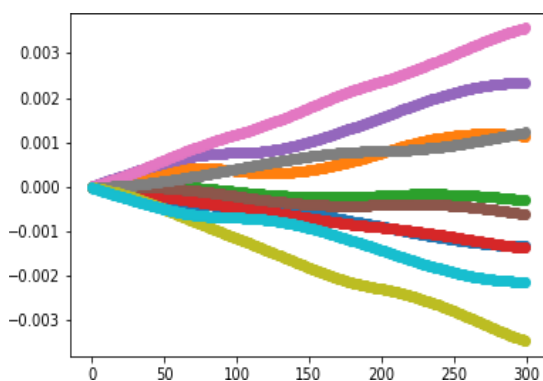


Figure 13:  $y$ -component of positions

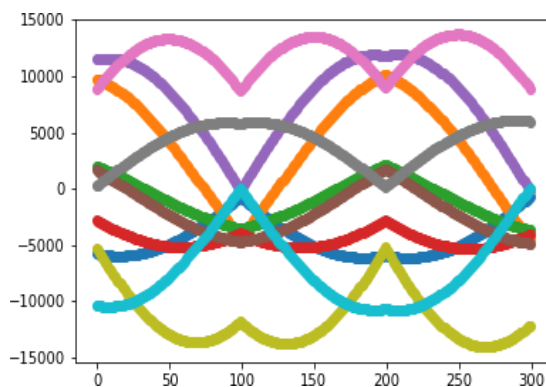


Figure 14:  $y$ -component of velocities

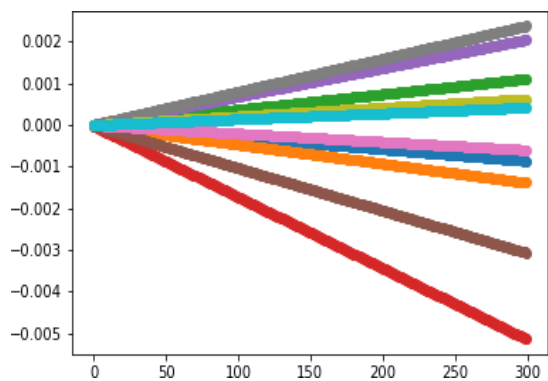


Figure 15:  $z$ -component of positions

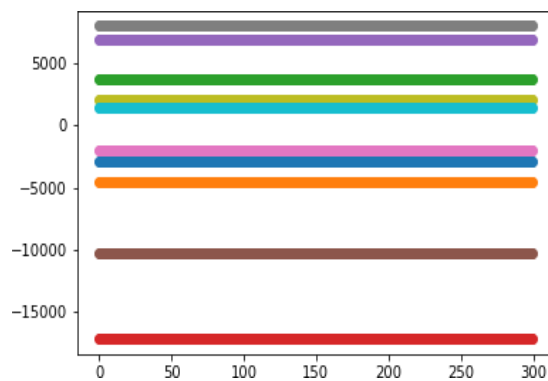


Figure 16:  $z$ -component of velocities

Figures (11), (13) and (15) show that the particles all start at  $[-0.5, 0, 0]$  as intended and spread out based on their initial velocities and velocities after subsequent updates; like in study 1 as the same sampling procedure for positions was used. In figure (16), the  $z$ - component of velocities stay the same as the magnetic field (while the electric field is set to 0) and hence the acceleration along the  $z$ -axis is zero. Figures (12) and (14) show three epochs of velocity changes (which can in such cases be shown to be simple harmonic motion as done in section 2.2.1 of chapter 2 in [1]), with 2 sharp changes at the 100th and the 200th steps when the direction of the magnetic field

was reversed. Three epochs of curves can also be observed in figures (11), (13), (15) and (9), and interesting pattern is observed in figure (10) because of this behavior.

## **Changing Magnetic field strength and direction**

### **Fields Setup**

For the second part of the study, a slightly different magnetic field configuration was used. The axis of the coil and the current were both varied.

Magnetic field due to a Helmholtz coil (number of turns: 1000 in each coil, radius: 0.1m):  
for 100 steps each

first 100A current, orientation along the  $z$ -axis  $[0,0,1]$

second -20A current, orientation along the  $x$ -axis  $[1,0,0]$  third 50A current, orientation along the  $y$ -axis  $[0,1,0]$ .

- Electric field constantly set to 0.

The positions of 1 and 10 particles in the simulation are plotted respectively in figures (17) and (18).

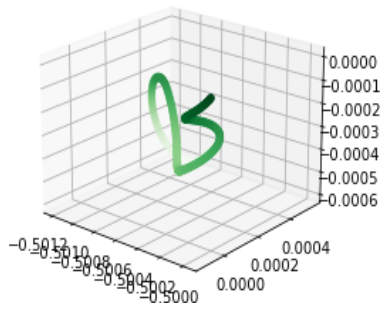


Figure 17: position of a particle

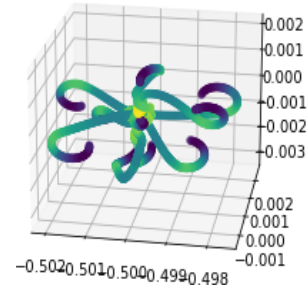


Figure 18: positions of 10 particles

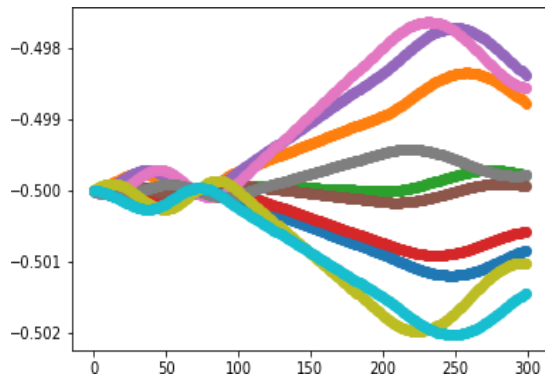


Figure 19:  $x$ -component of positions

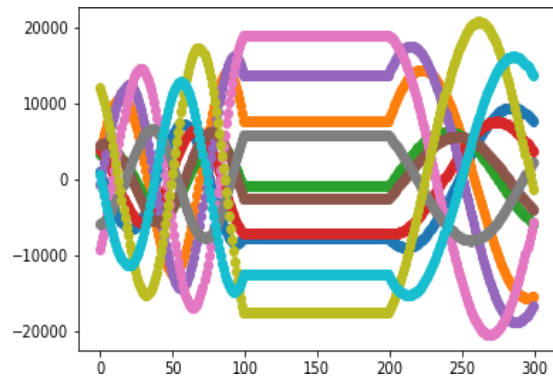


Figure 20:  $x$ -component of velocities

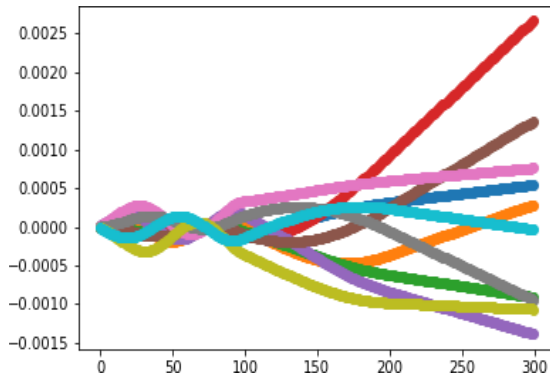


Figure 21:  $y$ -component of positions

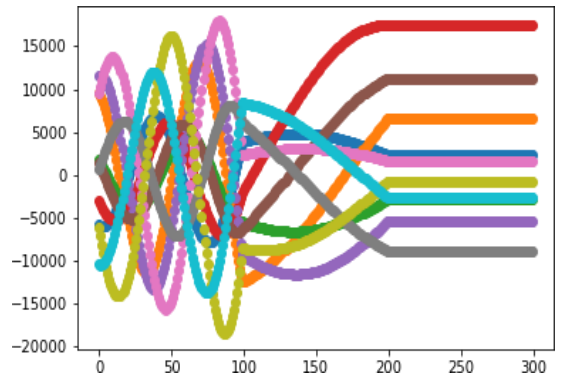


Figure 22:  $y$ -component of velocities

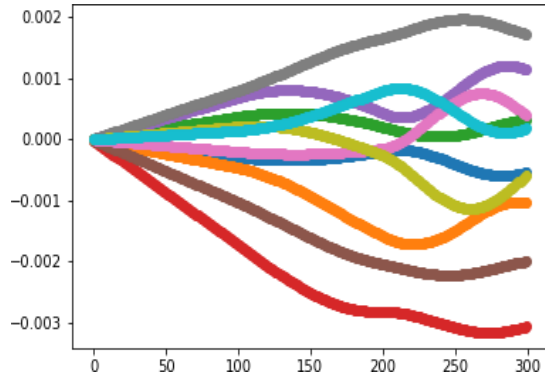


Figure 23:  $z$ -component of positions

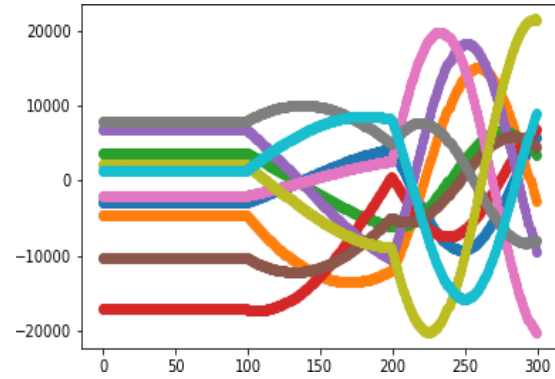


Figure 24:  $z$ -component of velocities

Like in the first part of the study figures (19), (21) and (23) also show that all the particles start at  $[-0.5, 0, 0]$ . Figures (20), (22) and (24) show 1 epoch of constant velocities when the magnetic field is along the respective axes (and hence the acceleration is zero along the respective axes) and 2 epochs of curved velocities of harmonic motion; as discussed in the first part of the study, when the magnetic field is along an axis perpendicular to the respective axes. 2 epoch of curves and 1 epoch of straight lines can also be seen in figures (19), (21) and (23). In figure (17), one can almost see that the three sections of the curve are in mutually perpendicular planes while one of the loops is much clearer. This behavior creates some interesting pattern in figure (18).

### 3.4 Study 3- Parabolic distributed speeds and Helmholtz coils

#### Sampling of the Initial Positions and Velocities of the particles

The following sampling data was used in the study:

- Speeds are sampled from a parabolic distribution with an equivalent temperature of 10000 K for a Maxwellian distribution.
- Velocity directions are sampled from uniform distribution.
- Positions are sampled such that all particles start at 0.5 m distance from the center  $[0,0, 0]$  (Particles injected or reflected from the walls of the chamber may be considered for reference).
- Positions are based on uniform distribution sampling of the position vector. The mean speed of a Maxwellian distribution is given by

$$\langle v \rangle = \sqrt{\frac{8RT}{\pi M}} = \sqrt{\frac{8 \cdot 8.31446261815324 \text{ J K}^{-1} \text{ mol}^{-1} \cdot 10000 \text{ K}}{\pi \cdot 1.008 \times 10^{-3} \text{ g mol}^{-1}}} = 14492.952993825971 \text{ m s}^{-1}$$

and the root mean square speed is given by  $v_{rms} = \sqrt{\frac{3RT}{M}}$ . Hence the variance,  $\sigma^2 = \langle v^2 \rangle - \langle v \rangle^2$  giving the standard deviation

...(29)

$$\sigma = \sqrt{\frac{RT}{M}} \sqrt{3 - \frac{8}{\pi^2}} = \sqrt{\frac{8.31446261815324 \text{ J K}^{-1} \text{ mol}^{-1} \cdot 10000 \text{ K}}{1.008 \times 10^{-3} \text{ g mol}^{-1}}} \cdot \sqrt{3 - \frac{8}{\pi^2}} = 13438.549997326772 \text{ m s}^{-1}$$

The speeds for the parabolic distribution were sampled using the Symmetric Beta distribution as defined in the SciPy library (`scipy.state.rdist[5]`) with parameter  $c = 4$  which gives a half parabolic distribution. Input parameters  $\text{loc} = 14492.952993825971$ ,  $\text{scale} = 30049.5113130523$  were used, which produces samples from the distribution with mean  $14492.952993825971 \text{ m s}^{-1}$  and standard deviation  $13438.549997326772 \text{ m s}^{-1}$ . The value for the scale was obtained by iterating until the value close to the required was produced until the maximum precision used by the library.

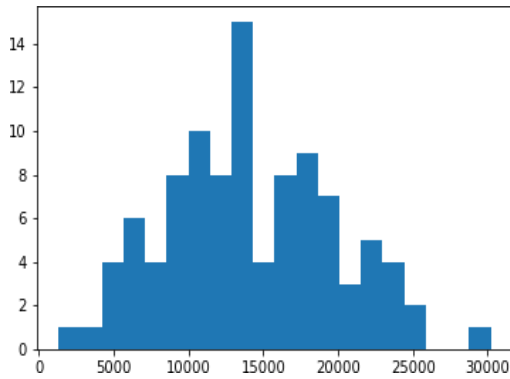


Figure 25: Maxwellian speeds of study 2

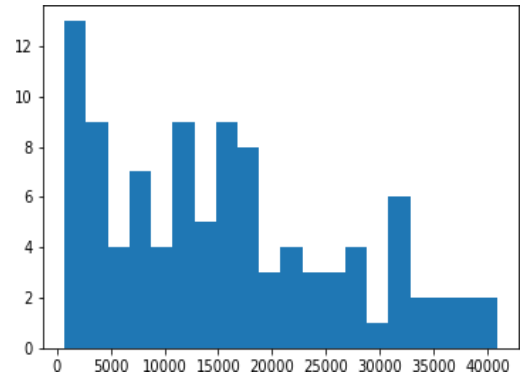


Figure 26: Parabolic speeds of study 3

On the left in figure (25) the histogram of Maxwellian sampled speeds from study 2 are plotted. On the right in figure (26) the histogram of speeds sampled from a Parabolic distribution are plotted.

Both distributions have the same mean and standard deviation. One can observe that a parabolic distribution approximates half of the Maxwellian distribution, which is why it could be interesting. It would also be interesting to have a double parabolic distribution that would approximate the entire Maxwellian distribution.

## Reversed Magnetic field

For the first part of this study, the same field configurations as in the first part of study 2 were used. As the fields used in the first part of this study are identical to the fields used in the first part of study 2, similar evolution behavior is expected (and observed) and hence the components of positions and velocities are not plotted and discussed; although they have been generated.

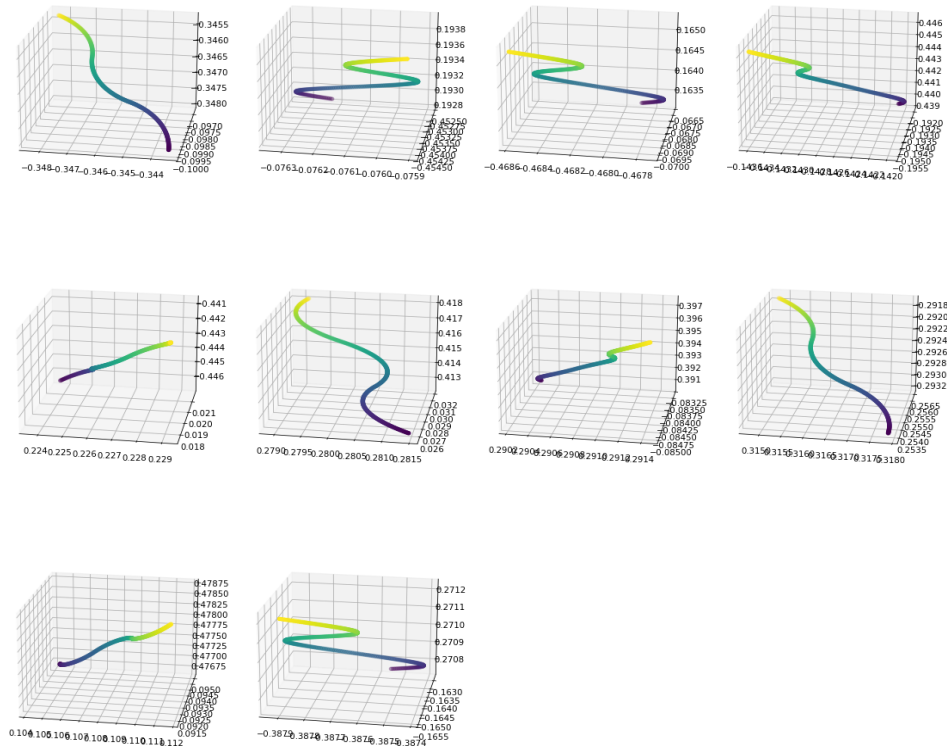


Figure 27: positions of 10 particles

Plotting the positions of the particles in figure (27), shows three epochs of curves for each particle as in the first part of study 2. Although, particles now all don't start in the same position as a different sampling procedure was used for the initialization of the positions.

### Changing Magnetic field strength and direction

For the second part of the study, the same field configurations as in the second part of study 2 were used. As the fields used in the second part of this study are identical to the fields used in the second part of study 2, similar evolution behavior is expected (and observed) and hence the components of positions and velocities are not plotted and discussed; although they have been generated.

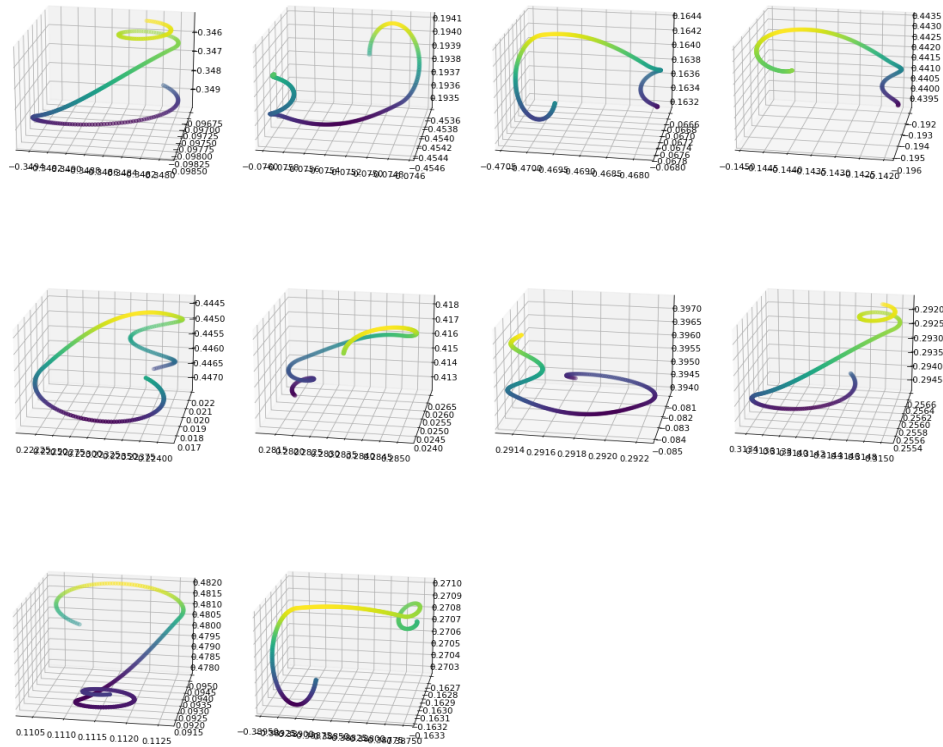


Figure 28: positions of 10 particles

Plotting the positions of the particles in figure (28), shows three circular curves of different orientations and radius; which is more clear for some particles than others, as in the second part of study 2. However, since the initial positions and velocities of the particles were sampled differently in study 3 than in study 2, a wider variety of trajectories are observed in the second part of study 3.

#### **4. CONCLUSIONS**

While setting up a real plasma device could take significant time and funding, setting up simulations like these can be done relatively quickly. Even being able to operate a plasma device might have its own challenges with operation costs and accessibility in terms of what parameters can be changed. It is difficult to run a real plasma device all the time. However, one can deploy such a program on the cloud rather easily, and have a continuously running simulation that the user can interact with. The advantages of the simulation really comes with how easy it is to run it and how long can it be run.

We set out to see if we could set up a plasma simulation from the ground up and we were able to do that along with 3 short studies. It is true that numerous works have been done in plasma simulations. However, these systems are often very complicated and intended for special purposes like astrophysical understanding or nuclear fusion. If we were to work on such a project, it might have taken us no less than a year or several. A lot of the complexity comes from the fact that a simulation solving Maxwell's equations will take much more memory and time. For a small scale plasma device, however, a much simpler model based on just Lorentz force is already powerful.

In the MEE4005 Surface Engineering class, Kaushal came across many plasma processes. At the beginning of the project; discussing the idea of the project, the members discussed that even though understanding a plasma system might seem more like a project in physics, if plasma is used in numerous processes in surface engineering techniques that are very important in the context of mechanical engineering, to be able to complement the understanding of surface engineering techniques with some understanding of how to use a plasma system seems important to better understanding and using plasma in surface engineering techniques.

#### **4.1 Contributions to the literature**

Study 1 mostly served the purpose of showing that the program works well. In study 1, the magnetic field was kept constant, while electric field was varied as one might in a lab experiment by scaling by certain numbers. Study 2 was about studying how one can control particles in a plasma by varying the magnetic field. In the first part of study 2, it was observed that one can trap the particles along the magnetic field. In the second part of study 2, it was observed that by changing the direction of the magnetic field, one can trap the particles in any direction. Possibly using a coil (like a Helmholtz coil setup) whose orientation can be changed, one can create a magnetic trap for a plasma chamber. This understanding is made clear by study 2. By controlling the current in the coil and its orientation, the plasma in the chamber can be controlled very well. In study 3, a different initial distribution of the velocities and positions for particles when they enter the chamber, were used. Initial distribution of positions describes where the particles are injected into the chamber. For example, maybe they are all injected at the same position through a port or valve like in study 2, or maybe they are sprayed through the walls of the chamber or perhaps one studies the flux of particles reflected off of the wall of the chamber; like in study 3. The velocity distribution of the particles entering the chamber also help one understand how the particles were prepared before entering the chamber; may be they were accelerated in an electric field, and also understand parameters like plasma temperature. In these studies the fields were changed in a few epochs so that these changes can be clearly noticed in the plots. Other strategies like changing the fields smoothly or slowly, would be difficult to observe well in the plots; although they would be very interesting from a computational perspective. With some extensions to a project like this, one could perform computational studies and understand optimization techniques to model and operate the behavior of a real plasma device; which would be very useful in operating real plasma devices and understanding plasma processes in surface engineering techniques.

## 4.2 Scope for future work

Further studies could be done with the available infrastructure or a similar one, including studies

that really help one use plasma devices better during plasma processes such as those in surface

engineering practices. Some interesting studies that could be done might be:

1. Define new field configurations; either analytical or based on expressions that mimic coils  
or instruments used to create fields.
2. Vary the fields (either slowly or abruptly) and see how the particles evolve.
3. Define new distributions to sample the initial position and velocity distributions.
4. Use different distributions in the same study to mimic particles entering the chamber through  
multiple routes.
5. Save the states of the particles at some point in the study and use that as initial  
distribution for another study.
6. Allow functionality to remove particles from the plasma to mimic absorption of particles  
in  
coating processes or ejection particles from the chamber.
7. Define interactions between the particles.
8. Create an interactive plasma simulation which keeps running on a computer and the user  
input can change the particles, fields and other properties of the simulation; so that it is like  
continuously running a plasma device in a lab.
9. Study the effect of plasma temperature on certain plasma processes.

## REFERENCES

- [1] Chen, F. F., “**Introduction to plasma physics and controlled fusion**”, New York: Plenum press, 1984. - [6] Myers, A., Colella, P., & Straalen, B. V. (2017).
- [2] Myers, A., Colella, P., & Straalen, B. V. (2017)., “**A 4th-Order Particle-in-Cell Method with Phase-Space Remapping for the Vlasov–Poisson Equation**” ,SIAM Journal on Scientific Computing, 39(3), B467-B485.
- [3] Qin, H., Zhang, S., Xiao, J., & Tang, W. M. (April, 2013), “**Why is Boris algorithm so good?**” .Princeton Plasma Physics Laboratory, PPPL-4872.
- [4] J.P. Boris, in 4th conference on numerical simulation of plasma, Washington, D.C., 1970, p.3

### Other References

- [1]Na, Yong-Su (2017), “**Introduction to nuclear fusion**” (Lecture 9 Mirror, lecture slide). SeoulNational University Open Courseware.
- [2] Föreläsning (2009), “**Charged particle motion in magnetic field**” , (lecture slide). Luleä Uni-versity of Technology.
- [3] Professor Sitaram Dash. (Fall Semester 2021)., “**MEE4005 Surface Engineering**” (lecturenotes). SMEC, VIT Vellore.
- [4] Matthew W. Kunz. (November 9, 2020), “ **Introduction to Plasma Astrophysics**” (lecturenotes). Princeton Plasma Physics Laboratory. - we learned about introductory concepts on plasma

## PUBLICATIONS

We are preparing a manuscript for submission to Plasma Physics Reports. We will make the submission in about a week.

## PLAGIARISM REPORT

Q  
*by* A B

---

**Submission date:** 26-Apr-2022 04:48PM (UTC+0530)

**Submission ID:** 1820791078

**File name:** 21BTECH10051\_thesis\_report\_1.docx (2.46M)

**Word count:** 10558

**Character count:** 54081

Q

ORIGINALITY REPORT

6%

SIMILARITY INDEX

5%

INTERNET SOURCES

2%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

[www.coursehero.com](http://www.coursehero.com)

Internet Source

3%

2

[vtechworks.lib.vt.edu](http://vtechworks.lib.vt.edu)

Internet Source

1%

3

Submitted to Engineers Australia

Student Paper

<1%

4

Submitted to Rochester Institute of Technology

Student Paper

<1%

5

[docplayer.net](http://docplayer.net)

Internet Source

<1%

6

Submitted to Clarkstown North High School

Student Paper

<1%

7

Submitted to University of Salford

Student Paper

<1%

8

Submitted to Aston University

Student Paper

<1%

9

Submitted to Imperial College of Science, Technology and Medicine

<1%

## **CONTRIBUTION STATEMENT**

### **Kaushal Timilsina:**

Kaushal worked on setting up the simulation, leading the work during the early stage; to set up the idea. Kaushal's significant contribution is in constants, particle, field, sampling, run and batch notebooks. Kaushal has also been working to manage the manuscript preparation process for submission to Plasma Physics Reports. Kaushal took the Surface Engineering class with Professor Sitaram Dash in the Fall Semester 2021; and studying many plasma processes inspired him to do a study on plasma simulation for the Capstone Project.

### **Hrishav Mishra:**

Hrishav's significant contribution is in working on studies 2 and 3 and leading the group during the final stage of work after review 2; to conclude the project. Hrishav also took the responsibility of working on the reports for the reviews and the A3 poster. Hrishav having a knack for learning about plasma physics and its applications in engineering which helped him to choose the topic with his teammates for the Capstone Project

### **Sashi Kant Shah:**

Sashi's significant contribution is in working on the checks performed to verify the program, and in study 1 and leading the group the review 1; to set up the study procedure. Sashi also took the responsibility of working on the presentations for the reviews. Sashi consulted with his teammates regarding the project, and a combination of engineering and physics sparked an instant interest of pursiot for his Capstone Project

Verified by



26/04/22

(Signature of the Guide)

## **PROGRAMME OUTCOMES ATTAINED**

<The guide is responsible to fill the POs attained by the student through this capstone project>

| <b>Programme Outcomes</b>   | <b>Attainment Level</b> |
|---|-------------------------|
| PO_1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.  | Attained up to 90%      |
| PO_2: Problem Analysis: Identity, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.   | Attained up to 90%      |
| PO_3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations. | Attained up to 90%      |
| PO_4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions for complex problems                           | Attained up to 90%      |
| PO_5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.   | Attained up to 90%      |
| PO_6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.   | Attained up to 85%      |
| PO_7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.   | Attained up to 95%      |

|  |                    |
|--|--------------------|
| PO_8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.   | Attained up to 90% |
| PO_9: Individual and Teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.   | Attained up to 90% |
| PO_10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. | Attained up to 90% |
| PO_11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.   | Attained up to 90% |
| PO_12: Life-long Learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.   | Attained up to 90% |

Verified by



26/04/2022  
( SITARAM DASH)

(Signature of the Guide)