

TABLE OF CONTENTS

1.	THEORY	2
1.1.	Accelerating Charged Particles	2
1.1.1.	Relation between momentum and acceleration	2
1.1.2.	Lorentz Force	4
1.1.3.	Relativistic Lorentz Force	4
1.1.4.	Acceleration caused by lorentz force	5
1.2.	Particle Accelerators	7
1.2.1.	Key Concepts	8
1.2.2.	Acceleration Cavities	9
1.2.3.	Steerer Magnets	10
1.2.4.	Rhodotron Accelerator	11
1.2.4.1.	Cavity of a Rhodotron	11
1.2.4.2.	Acceleration cycle of Rhodotron	13
1.3.	Numerical Integration Methods	16
1.3.1.	Leapfrog	16
1.3.2.	Runge Kutta	17
1.4.	Simulation	20
1.4.1.	Basic Concepts in Programming	20
1.4.1.1.	Clock Cycle	20
1.4.1.2.	Concurrency	21
2.	DESIGN	22
2.1.	Cavity Design	22
2.2.	Magnet Design	27
2.2.1.	$n\lambda$ Technique	28
2.2.2.	L_{out} Parameter Sweep	30
2.3.	Initial Design at KAHVELab	33
3.	SIMULATION	40
3.1.	Proof of Concept	40

3.2. Intermediate Versions	43
3.2.1. L_{out} Optimization For Single e^-	43
3.2.2. ϕ_{lag} Optimization For Bunches	43
3.2.3. Simulation in 3D	45
3.2.4. Acceleration in Magnetic Field	46
3.2.4.1. RK4-1	47
3.2.4.2. RK4-2	47
APPENDIX A: Intermediate Codes	50
APPENDIX B: Example Simulation Runs	55
REFERENCES	58

1. THEORY

1.1. Accelerating Charged Particles

1.1.1. Relation between momentum and acceleration

In classical mechanics, Newton's second law defines force and states

$$\vec{F} = \frac{\partial \vec{p}}{\partial t} = m \frac{\partial \vec{v}}{\partial t} = m \vec{a} \quad (1.1)$$

In special relativity however, relativistic momentum is defined as $\vec{p} = \gamma m_0 \vec{v}$ where γ is the Lorentz Factor;

$$\gamma = \frac{1}{\sqrt{1 - v^2/c^2}} = \frac{1}{\sqrt{1 - \beta^2}}$$

Considering these two statements, we can find the relation between momentum and acceleration as follows

$$\begin{aligned} \frac{\partial \vec{p}}{\partial t} &= m_0 \frac{\partial(\gamma \vec{v})}{\partial t} = m_0 \left\{ \frac{\partial \gamma}{\partial t} \vec{v} + \gamma \frac{\partial \vec{v}}{\partial t} \right\} \\ \frac{\partial \gamma}{\partial t} &= \gamma^3 \vec{\beta} \cdot \frac{\partial \vec{\beta}}{\partial t} = \frac{\gamma^3}{c} \vec{\beta} \cdot \vec{a} \\ \frac{\partial \vec{p}}{\partial t} &= m_0 \left\{ \frac{\gamma^3}{c} (\vec{\beta} \cdot \vec{a}) \vec{v} + \gamma \frac{\partial \vec{v}}{\partial t} \right\} \end{aligned}$$

$$\vec{F} = \gamma m_0 \left\{ \vec{a} + \gamma^2 (\vec{\beta} \cdot \vec{a}) \vec{\beta} \right\} \quad (1.2)$$

It is clear that acceleration is not necessarily parallel to the force anymore.

To start separating the parallel and perpendicular components relative to $\vec{\beta}$, we

can find $\vec{\mathbf{a}}_{||}$ and $\vec{\mathbf{F}}_{||}$.

$$\vec{\mathbf{a}}_{||} = \frac{(\vec{a} \cdot \vec{\beta})}{\beta^2} \beta \quad \vec{\mathbf{F}}_{||} = \frac{(\vec{F} \cdot \vec{\beta})}{\beta^2} \beta \quad (1.3)$$

$$\begin{aligned} \vec{\mathbf{F}} \cdot \vec{\beta} &= \gamma m_0 \{ \vec{\mathbf{a}} \cdot \vec{\beta} + \gamma^2 (\vec{\beta} \cdot \vec{\mathbf{a}}) \beta^2 \} \\ &= \gamma m_0 (\vec{\mathbf{a}} \cdot \vec{\beta}) \{ \gamma^2 \beta^2 + 1 \} \end{aligned}$$

Using $\gamma^2 \beta^2 + 1 = \gamma^2$ we have,

$$\vec{\mathbf{F}} \cdot \vec{\beta} = m_0 \gamma^3 (\vec{\mathbf{a}} \cdot \vec{\beta})$$

Inserting this

$$\begin{aligned} \vec{\mathbf{F}}_{||} &= \frac{(\vec{\mathbf{F}} \cdot \vec{\beta})}{\beta^2} \vec{\beta} \\ &= m_0 \gamma^3 \frac{(\vec{\mathbf{a}} \cdot \vec{\beta})}{\beta^2} \vec{\beta} \\ &= m_0 \gamma^3 \vec{\mathbf{a}}_{||} \end{aligned} \quad (1.4)$$

Therefore from *equations 1.2 and 1.4*

$$\begin{aligned} \vec{\mathbf{F}} &= m_0 \gamma^3 \vec{\mathbf{a}}_{||} \beta^2 + m_0 \gamma \vec{\mathbf{a}} \\ &= m_0 \gamma^3 \vec{\mathbf{a}}_{||} \beta^2 + m_0 \gamma \{ \vec{\mathbf{a}}_{||} + \vec{\mathbf{a}}_{\perp} \} \\ &= m_0 \vec{\mathbf{a}}_{||} \gamma \{ \gamma^2 \beta^2 + 1 \} + m_0 \gamma \vec{\mathbf{a}}_{\perp} \\ &= m_0 \vec{\mathbf{a}}_{||} \gamma^3 + m_0 \gamma \vec{\mathbf{a}}_{\perp} \\ &= \vec{\mathbf{F}}_{||} + m_0 \gamma \vec{\mathbf{a}}_{\perp} \end{aligned}$$

We finally have two separate equations which are in similar form with *equation 1.1*.

$$\vec{\mathbf{F}}_{||} = \gamma^3 m_0 \vec{\mathbf{a}}_{||} \quad \vec{\mathbf{F}}_{\perp} = \gamma m_0 \vec{\mathbf{a}}_{\perp} \quad (1.5)$$

1.1.2. Lorentz Force

Force acting on a charged particle moving in electromagnetic fields is called Lorentz Force and is given by the formula

$$\frac{\partial \vec{p}}{\partial t} = \vec{F}_L = q(\vec{E} + \vec{v} \times \vec{B}) \quad (1.6)$$

where the q is the charge and \vec{v} is the velocity of the particle.

1.1.3. Relativistic Lorentz Force

Similar to non-relativistic version, relativistic Lorentz Force is given by the following 4-vector equality

$$\frac{\partial p^\mu}{\partial \tau} = q F^{\mu\nu} u_\nu \quad (1.7)$$

Where $\partial\tau = \partial t/\gamma$,

$$p^\mu = \begin{bmatrix} W/c \\ p_x \\ p_y \\ p_z \end{bmatrix} \quad F^{\mu\nu} = \begin{bmatrix} 0 & -E_x/c & -E_y/c & -E_z/c \\ E_x/c & 0 & -B_z & B_y \\ E_y/c & B_z & 0 & -B_x \\ E_z/c & -B_y & B_x & 0 \end{bmatrix} \quad u_\nu = \gamma \begin{bmatrix} c \\ -v_x \\ -v_y \\ -v_z \end{bmatrix} \quad (1.8)$$

Where W is the energy of the particle and γ is the lorentz factor mentioned in the previous section.

For $\mu = 0$, we have the time component of the equation;

$$\frac{\gamma}{c} \frac{\partial W}{\partial t} = \frac{q\gamma \vec{E} \cdot \vec{v}}{c} = \frac{q\gamma}{c} \frac{\vec{E} \cdot \partial \vec{r}}{\partial t} \quad (1.9)$$

$$\frac{\partial W}{\partial t} = q \frac{\vec{E} \cdot \partial \vec{r}}{\partial t} \quad (1.10)$$

This is the definition of work done by an electric field. For $\mu = 1, 2, 3$, we have the spacial components;

$$\frac{\partial \vec{p}}{\partial \tau} = \gamma \frac{\partial \vec{p}}{\partial t} = q\gamma(\vec{E} + \vec{v} \times \vec{B})$$

Which simplifies to non-relativistic Lorentz Force in *equation 1.6*.

1.1.4. Acceleration caused by lorentz force

Due to the nature of the cross product, lorentz force caused by a magnetic field is always perpendicular to the velocity of the particle. Therefore the acceleration of the magnetic field is straightforward

$$\vec{F}_B = \vec{F}_\perp = \gamma m_0 \vec{a}_\perp = \gamma m_0 \vec{a}_B$$

The same thing cannot be said about electric field however. It can create force in any direction with respect to velocity. Therefore, we have the following equality;

$$\vec{a}_{||} = \frac{q}{\gamma^3 m_0} \vec{E}_{||} \quad \vec{a}_\perp = \frac{q}{\gamma m_0} \{ \vec{E}_\perp + \vec{v} \times \vec{B} \} \quad (1.11)$$

Acceleration due to electric field can be simplified as;

$$\begin{aligned} \vec{a}_{\{B=0\}} = \vec{a}_E &= \vec{a}_{||} + \vec{a}_{\perp\{B=0\}} \\ &= \frac{q}{m_0 \gamma} \left\{ \frac{\vec{E}_{||}}{\gamma^2} + \vec{E}_\perp \right\} \\ &= \frac{q}{m_0 \gamma} \{ \{1 - \beta^2\} \vec{E}_{||} + \vec{E}_\perp \} \end{aligned}$$

$$\begin{aligned}
&= \frac{q}{m_0\gamma} \{\vec{\mathbf{E}}_{||} + \vec{\mathbf{E}}_{\perp} - \beta^2 \vec{\mathbf{E}}_{||}\} \\
&= \frac{q}{m_0\gamma} \{\vec{\mathbf{E}} - \beta^2 \vec{\mathbf{E}}_{||}\}
\end{aligned}$$

Using the fact that $\vec{\mathbf{E}}_{||} = \vec{\beta}(\vec{\mathbf{E}} \cdot \vec{\beta})/\beta^2$, we finally have

$$\vec{\mathbf{a}}_E = \frac{q}{\gamma m_0} \left\{ \vec{\mathbf{E}} - \vec{\mathbf{v}} \frac{(\vec{\mathbf{E}} \cdot \vec{\mathbf{v}})}{c^2} \right\} \quad \vec{\mathbf{a}}_B = \frac{q}{\gamma m_0} (\vec{\mathbf{v}} \times \vec{\mathbf{B}}) \quad (1.12)$$

1.2. Particle Accelerators

Particle accelerators are sophisticated scientific instruments designed to accelerate charged particles, such as electrons, protons, or ions, to high speeds and energies. These accelerators play a crucial role in advancing our understanding of the fundamental properties of matter and the universe. They are widely used in various fields of research, including particle physics, nuclear physics, materials science, and medicine.

At their core, particle accelerators utilize electromagnetic fields to impart energy to particles and control their trajectories. These fields are generated by intricate arrangements of magnets and RF (radiofrequency) cavities within the accelerator structure. By precisely controlling these fields, accelerators can propel particles to speeds close to the speed of light, enabling them to acquire high energies.

Accelerators can be categorized into two main types: linear accelerators (linacs) and circular accelerators. Linacs accelerate particles in a straight line, while circular accelerators use magnetic fields to bend the particle trajectory into a circular path.

The acceleration process in accelerators involves multiple stages. Initially, particles are injected into the accelerator at a relatively low energy. As they progress through the accelerator, they are subjected to alternating electric fields that accelerate them, while magnetic fields guide their trajectories. Focusing elements, such as magnetic lenses or quadrupole magnets, ensure the particles remain tightly controlled.

As particles gain energy in the accelerator, they approach relativistic speeds, where relativistic effects become significant. Special relativity governs the increase in mass and energy of the particles as they approach the speed of light, providing insights into the behavior of matter at high energies.

Particle accelerators are essential tools for scientific research. They enable scientists to probe the fundamental constituents of matter, study particle interactions, and

explore the laws of physics. Accelerators have been instrumental in discovering and characterizing fundamental particles, such as quarks, leptons, and the Higgs boson. They also facilitate the production of high-intensity beams for applications in material science, radiotherapy, and industrial processes like particle irradiation and sterilization.

***Image
Coming
Soon***

Figure 1.1: A circular accelerator

***Image
Coming
Soon***

Figure 1.2: A linear accelerator

1.2.1. Key Concepts

- Phase Stability
- Phase Lag
- Shunt Impedance

1.2.2. Acceleration Cavities

Radiofrequency (RF) cavities, also known as accelerating cavities or resonant cavities, are key components in particle accelerators. These cavities generate strong electromagnetic fields at specific frequencies to accelerate charged particles through clever engineering.

RF cavities are typically hollow metallic structures made of or coated with high-conductivity materials such as copper. They are designed to resonate at a specific frequency, which is determined by the size and shape of the cavity. The cavity is often cylindrical or spherical in shape, and its inner surface is polished to minimize energy losses through resistive heating. The RF cavity is designed to be resonant, meaning that it naturally amplifies the electric fields at its resonant frequency. The resonant frequency is determined by the cavity's dimensions and the speed of light in the cavity material.

To achieve efficient energy transfer to the particles, the RF cavity is driven by an external RF power source operating at the resonant frequency. The power source supplies radiofrequency energy to the cavity, which causes the electric fields inside the cavity to oscillate at the desired frequency. These oscillating fields then transfer energy to the passing particles, increasing their kinetic energy by pushing and pulling on the charged particles as they pass through the cavity.

In addition to accelerating the particles, RF cavities are often designed to provide focusing forces. By carefully shaping the cavity and adjusting the electromagnetic fields, the particles can experience focusing effects as they pass through the cavity. This helps to maintain a tight and controlled beam. To ensure efficient acceleration, it is essential to maintain phase stability. This means that the particles should experience the strongest electric fields at the correct time during their passage through the cavity. Precise timing and synchronization of the RF power source with the particle beam are crucial to achieve phase stability and maximize energy transfer.

***Image
Coming
Soon***

Figure 1.3: A rf cavity used in KAHVELab

1.2.3. Steerer Magnets

Steerer magnets, also known as dipole magnets or bending magnets, are fundamental components used in particle accelerators to control the trajectory of charged particles. They utilize the Ampere's Law to exert a magnetic field that interacts with the charged particles in the accelerator.

According to the Lorentz Force Law (*Section 1.1.2*), when a charged particle moves through a magnetic field, it experiences a force perpendicular to both its velocity vector and the magnetic field direction. This force causes the particle's trajectory to curve, resulting in a bending effect.

***Image
Coming
Soon***

Figure 1.4: A steerer magnet used in KAHVELab

1.2.4. Rhodotron Accelerator

Rhodotron Accelerator is a type of particle accelerator that was proposed by *Jacques POTTIER* in his publication *A new type of rf electron accelerator: The Rhodotron* in 1989 [1]. First prototype was built at CEA Saclay later in 1992 [2]. Its named after the greek word for rose, *rhodos*, due to the shape of the design [3].

The design of a rhodotron mainly consists of a coaxial cylindrical RF cavity and steerer magnets surrounding it. RF cavity is fed by an external RF source, accelerating the electrons entering from an attached electron gun.

1.2.4.1. Cavity of a Rhodotron. Coaxial design of the cavity concentrates the electric field, while the magnetic field diminishes in the middle of the cylinders. Therefore the electrons are injected and accelerated in the plane of zero magnetic field where the electric field is strongest.

***Image
Coming
Soon***

Figure 1.5: SUPERFISH simulation of a coaxial cavity

Image Coming Soon

Figure 1.6: CST simulation of a coaxial cavity

With the expectation that the electrons will accelerate to speeds $\geq 0.9c$ after the first pass, a rhodotron cavity is designed so that the length of the path between successive passes is an integer multiple of λ , wavelength of the RF field ($l = p\lambda$). This constraint helps with phase stability and synchronization of the beam.

In the table below, optimized characteristics of a rhodotron cavity can be observed.

Table 1
Optimized characteristics

P	R_2 (m)	R_1/R_2	Z_{se} (MΩ)	Z_{sp} (MΩ)
1	0.27λ	$1/4$	$5.77\lambda^{1/2}$	$4.9\lambda^{1/2}$
2	0.5λ	$1/7$	$10.4\lambda^{1/2}$	$8.83\lambda^{1/2}$

Figure 1.7: Optimized characteristics of a rhodotron cavity [1]

Here, p is the integer multiplier in the equation ($l = p\lambda$) mentioned above, R_1 is the radius of the inner cylinder, R_2 is the radius of the outer cylinder, Z_{se} is effective shunt impedance, Z_{sp} is practical shunt impedance which was taken to be $0.85Z_{se}$.

Energy gain after each pass can be calculated by the following formula [1]

$$W_1 = Z_{sp}^{1/2} P^{1/2} \cos\phi \quad (1.13)$$

Where P is the dissipated power, ϕ is the phase lag, taken as 15° [1].

Considering $Z_{sp} \propto \lambda^{1/2}$, $W_1 \propto \lambda^{1/4}$, $V \propto \lambda^3$, where V is the volume of the cavity, implementing the $p = 1$ design in *figure 1.7* is much more space efficient.

Table 2
Energy W (MeV) for $P = 100$ kW and $f = 130$ MHz

R_2 (m)	2	3	4	5	6	7	8	9	10	11	12
0.62	1.8	2.7	3.6	4.5	5.4	6.3	7.2	8.1	9	9.9	10.8
1.15	2.4	3.6	4.8	6	7.2	8.4	9.6	10.8	12	13.2	14.4

Figure 1.8: Energy of a synchronous electron after each pass for both $p = 1$ and $p = 2$ [1]

Total energy gain after n passes W can then be found by *equation 1.13*, taking $\phi = 15^\circ$, $p = 1$, i.e $R_2 = 0.27\lambda$.

$$W \approx 2.14\lambda^{1/4}P^{1/2}n \quad keV \quad (1.14)$$

1.2.4.2. Acceleration cycle of Rhodotron. Electrons undergo four different stages during a single pass. They are accelerated between the cylindrical plates and are shielded from the changing RF field while inside the inner cylinder and outside the cavity.

First acceleration Electrons in the rhodotron cavity are accelerated by the electric

field created between two coaxial cylinders, towards the inner cylinder when they are ejected into the cavity.

Inner shielding While inside the inner cylinder, the cylinder acts as a faraday cage and shields the electrons inside while the electric field is being reversed.

Second acceleration Once the electrons leave the inner cylinder, they accelerate towards the outer cylinder by the reversed electric field until they leave the cavity.

Magnets After leaving the cavity, an electromagnet placed in their path steers the electrons back into the cavity in which time the electric field changes the direction again.

This cycle can be repeated as long as real world constraints such as; placements and dimentions of the electromagnets, power requirements due to increasing magnetic field in order for sharper turns, can be overcomed. After desired amount of pass has been completed, the electrons exit the accelerator.

This process is explained further in the following figures where T is the period of the electric field.

***Image
Coming
Soon***

Figure 1.9: $[0, \frac{T}{4}]$ time frame of a rhodotron

*Image
Coming
Soon*

Figure 1.10: $[\frac{T}{4}, \frac{T}{2}]$ time frame of a rhodotron

*Image
Coming
Soon*

Figure 1.11: $[\frac{T}{2}, \frac{3T}{4}]$ time frame of a rhodotron

*Image
Coming
Soon*

Figure 1.12: $[\frac{3T}{4}, T]$ time frame of a rhodotron

1.3. Numerical Integration Methods

1.3.1. Leapfrog

The Leapfrog method is a numerical method commonly used to solve ordinary differential equations (ODEs) that involve second-order time derivatives. Such an ODE is shown below:

$$\ddot{x} = \frac{d^2x}{dt^2} = f(x) \quad (1.15)$$

The Leapfrog method is a variant of the finite difference method, and it approximates the solution of an ODE by discretizing both time and space. The method gets its name from the way it calculates the values of the solution at each time step, which resembles a "leapfrogging" motion. It is a simple and efficient algorithm that is often used in simulations of physical systems, such as celestial mechanics or molecular dynamics. The idea is straight forward; in the time interval Δt ,

$$\begin{aligned} a(t_0) &= f(x_0) \\ x(t_0 + \Delta t) &= x(t_0) + v(t_0)\Delta t + a(t_0)\frac{\Delta t^2}{2} \end{aligned} \quad (1.16)$$

$$v(t_0 + \Delta t) = v(t_0) + \{a(t_0) + a(t_0 + \Delta t)\}\frac{\Delta t}{2} \quad (1.17)$$

For more stability, this version can be rearranged to what is called 'kick-drift-kick' form,

$$\begin{aligned} v(t_0 + \Delta t/2) &= v(t_0) + a(t_0)\frac{\Delta t}{2} \\ x(t_0 + \Delta t) &= x(t_0) + v(t_0 + \Delta t/2)\Delta t \end{aligned} \quad (1.18)$$

$$v(t_0 + \Delta t) = v(t_0 + \Delta t/2) + a(t_0 + \Delta t) \frac{\Delta t}{2}$$

This version provides more time resolution to our calculation; however, it increases the number of calculations needed by about 50%.

1.3.2. Runge Kutta

The Runge-Kutta methods, named after the German mathematicians Carl Runge and Martin Wilhelm Kutta, family of numerical methods used to solve ordinary differential equations (ODEs) that are in the following form

$$\frac{\partial y}{\partial x} = f(x, y) \quad (1.19)$$

The basic idea behind the Runge-Kutta method is to approximate the solution of an ODE by taking small steps and using a weighted average of function evaluations at different points within each step.

$$y_{n+1} = y_n + \delta x \sum_{i=1}^m b_i k_i \quad x_{n+1} = x_n + \delta x \quad (1.20)$$

where

$$k_i = f(x_n + c_i \delta x, y_n + \delta x \sum_{j=1}^{i-1} a_{ij} k_j) \quad (1.21)$$

These *equations 1.20 and 1.21* define the family of methods. To specify a pertic-

ular method, order m , coefficients a_{ij} , b_i and c_i should be provided. The coefficients of any Runge-Kutta method can be visualized by a tableau called *Butcher Tableau*.

0						
c_1	a_{21}					
c_2	a_{31}	a_{32}				
\vdots	\dots					
c_m	a_{m1}	a_{m2}	\dots	$a_{m,m-1}$		
	b_1	b_2	\dots	b_m		

Figure 1.13: Butcher Tableau

The simplest Runge-Kutta method is the *Euler's method*. Its *Butcher tableau* is as follows.

0			
		1	

The most commonly used version of the Runge-Kutta method is the fourth-order Runge-Kutta method, also known as RK4. The RK4 method involves four function evaluations per step and has an error term that is proportional to the step size raised to the fifth power. Two of the most used *Butcher tableaus* for RK4 given below.

0						
$1/2$	$1/2$					
$1/2$	0	$1/2$				
1	0	0	1			
	$1/6$	$1/3$	$1/3$	$1/6$		
0						
$1/3$	$1/3$					
$2/3$	$-1/3$	1				
1	1	-1	1			
	$1/8$	$3/8$	$3/8$	$1/8$		

Figure 1.14: Butcher Tableau for RK4

The second tableau in *figure 1.14* is called the "*3/8 rule*". Its main advantage is that its error coefficients are smaller than the other. But it costs more floating point operations per step. Resulting in slower calculations.

1.4. Simulation

A simulation software is a computer program or tool that enables the creation and execution of simulations to model and analyze real-world systems or processes. It allows users to replicate the behavior, interactions, and outcomes of the system or process under study, providing insights and predictions that can be valuable for decision-making, optimization, or understanding complex phenomena.

Simulation software provides a virtual environment where users can define the parameters, variables, and rules of the system being simulated. The software then uses mathematical and physical models, algorithms, and computational techniques to simulate the behavior of the system over time.

1.4.1. Basic Concepts in Programming

1.4.1.1. Clock Cycle. In programming, a clock cycle refers to a fundamental unit of time measurement used in computer systems. It represents the basic rhythm or timing mechanism of a computer's central processing unit (CPU) and is typically measured in terms of the CPU's clock speed, expressed in hertz (Hz).

A clock cycle represents one complete pulse or oscillation of the CPU's clock signal. It serves as a synchronization mechanism, coordinating the execution of instructions and the timing of various operations within the CPU. Each clock cycle is associated with a specific duration, which is determined by the clock speed of the CPU.

The concept of clock cycles is often used when considering the performance and efficiency of algorithms and code. Since the execution time of instructions is influenced by the number of clock cycles required, minimizing the number of clock cycles needed for a program or algorithm can lead to faster and more efficient code execution. Table below shows the amount of clock cycles required for some mathematical calculations in various processors.

MEASURED CPU TIMES OF BASIC MATHEMATICAL OPERATIONS							
OPERATION	DOUBLE PRECISION					QUADRUPLE PRECISION	
	i486	Pentium	PA-7000	μ SPARC-II	Alpha	PA-7000	μ SPARC-II
$x = y \dots$	0.46	0.24	1.00	1.00	0.41	5.1	2.8
$x = y + z \dots$	0.91	1.00	1.00	1.00	1.00	131	11.3
$x = yz \dots$	1.09	1.00	1.00	1.00	1.00	457	99.5
$x = y/z \dots$	2.81	4.49	4.92	3.32	6.59	1100	168
$x = y^{1/2} \dots$	12.0	19.6	4.95	24.4	38.5	1100	149
$x = \sin y \dots$	19.8	23.3	49.8	22.0	18.5	9510	1740
$x = \log y \dots$	20.2	23.9	47.8	31.9	22.0	5420	1240
$x = \tan^{-1} y \dots$	19.8	27.3	46.3	37.3	23.5	5930	1270
$x = \tan y \dots$	18.0	28.3	63.2	30.0	35.1	8450	2040
$x = \exp y \dots$	24.9	32.7	42.0	19.0	23.2	6560	1370

Figure 1.15: Amount of clock cycle for mathematical operations in various processors.
[4]

Note that *division* is by far the most time consuming basic mathematical operation between addition, subtraction and multiplication.

1.4.1.2. Concurrency. Concurrency in programming refers to the ability of a program to execute multiple tasks or processes simultaneously. It allows different parts of a program to make progress independently, potentially improving performance, responsiveness, and resource utilization.

Concurrency in single core can be achieved by implementing clever scheduling of list of operations called *threads*. This results in non-blocking execution of multiple threads, but only one thread would be executed at any given time. True concurrency on the other hand, can only be achieved by using multiple cores. Each core would be able to execute one thread at any time.

Programs that implement concurrency using multiple cores executes more operations per clock cycle. However, this does not directly lead to an improved performance due to the heavy burden of scheduling and managing multiple threads.

2. DESIGN

2.1. Cavity Design

The first and the most important design parameter for a cavity is the operating RF frequency. After an operating RF frequency is set and the desired $R1/R2$ relation *figure 1.7* is selected, design parameters of acceleration plane of the cavity is fully determined.

By following the cylindrical design mentioned in *section 1.2.4*, the only main design parameter remaining is the height of the cylindrical cavity. This parameter can be found using the constraint mentioned in *section 1.2.2*; the fact that operating RF frequency must be equal to resonant frequency of the cavity. For simple coaxial cavity, the height should be $\lambda/2$, where λ is the wavelength of the external RF supply [1]. Simulation tools such as *CST Studio*, *Poisson SUPERFISH* can be used to confirm this condition.

In the following examples, $p = 1$ from *figure 1.7* was used, and *it will be the focus of all further calculations*. Using $f_{RF} = 107.5$ MHz, $f_{RF} = 180$;

$$\begin{aligned}
 f_{107.5} &= 107.5 \text{ MHz} & f_{180} &= 180 \text{ MHz} \\
 \lambda_{107.5} &= \frac{c}{f_{107.5}} = 2.789m & \lambda_{180} &= \frac{c}{f_{180}} = 1.666m \\
 R_2 &= 0.27 \times \lambda_{107.5} = 0.753m & R_2 &= 0.27 \times \lambda_{180} = 0.450m & (2.1) \\
 R_1 &= \frac{R_2}{4} = 0.188m & R_1 &= \frac{R_2}{4} = 0.113m \\
 h &= \frac{\lambda}{2} = 1.394m & h &= \frac{\lambda}{2} = 0.833m
 \end{aligned}$$

In the following figures, *Poisson Superfish* and *CST* simulations of two cavities defined by *equation 2.1* can be observed.

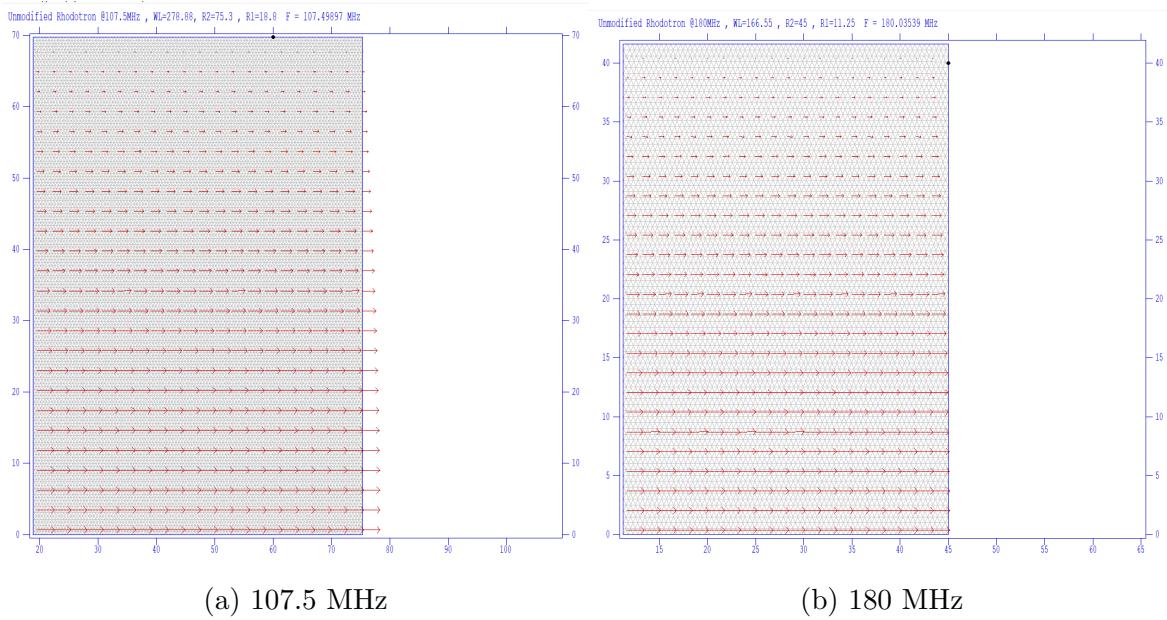


Figure 2.1: Poission Superfish results for *equation 2.1*

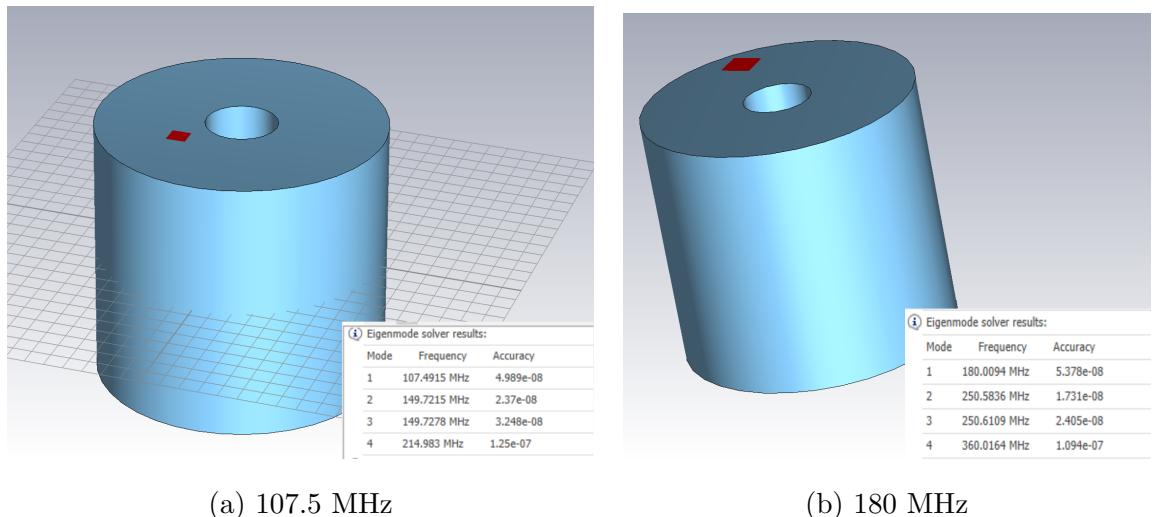
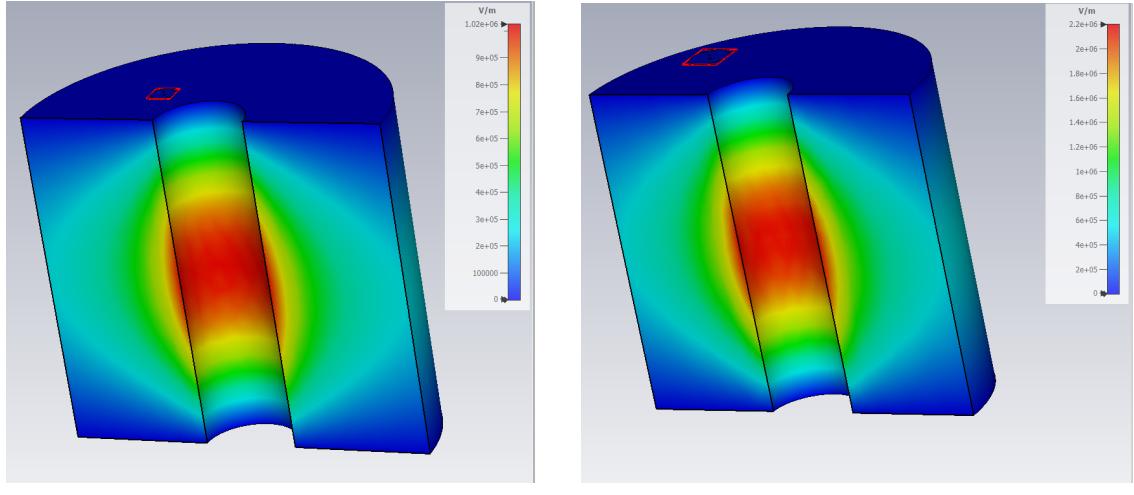


Figure 2.2: CST Eigenmode results for *equation 2.1*

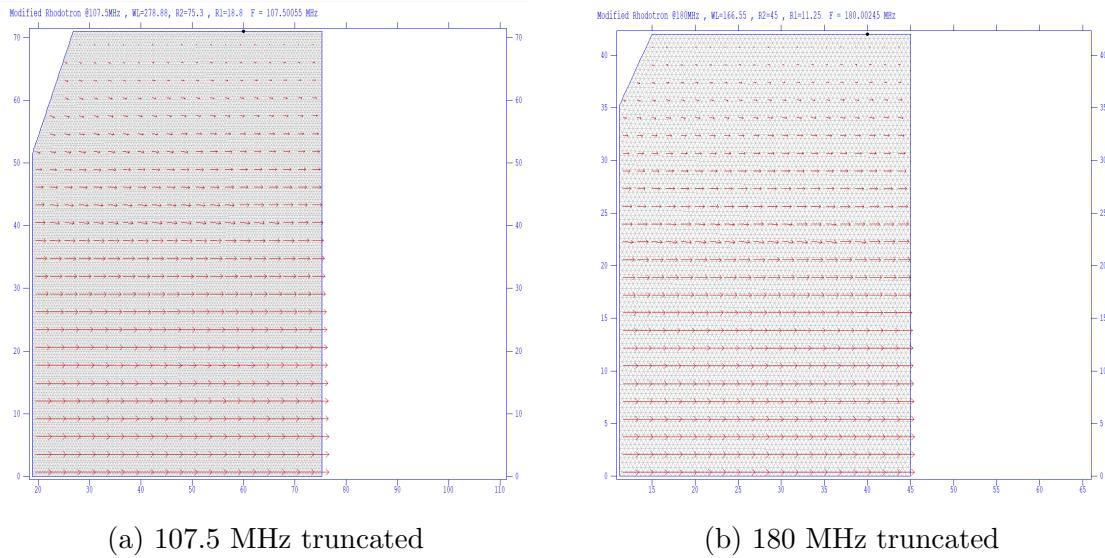


(a) 107.5 MHz

(b) 180 MHz

Figure 2.3: CST Electric field solutions for *equation 2.1*

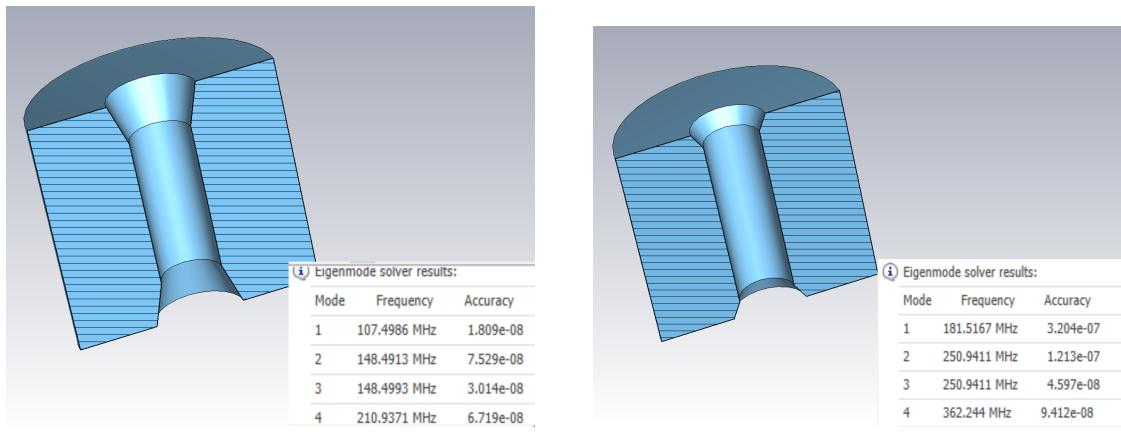
As mentioned by POTTIER, truncated cone terminations in inner cylinder can improve the shunt impedance Z of the cavity [1]. Below, *Poisson superfish* results of such a modification with matching height increase to keep resonant frequency can be found.



(a) 107.5 MHz truncated

(b) 180 MHz truncated

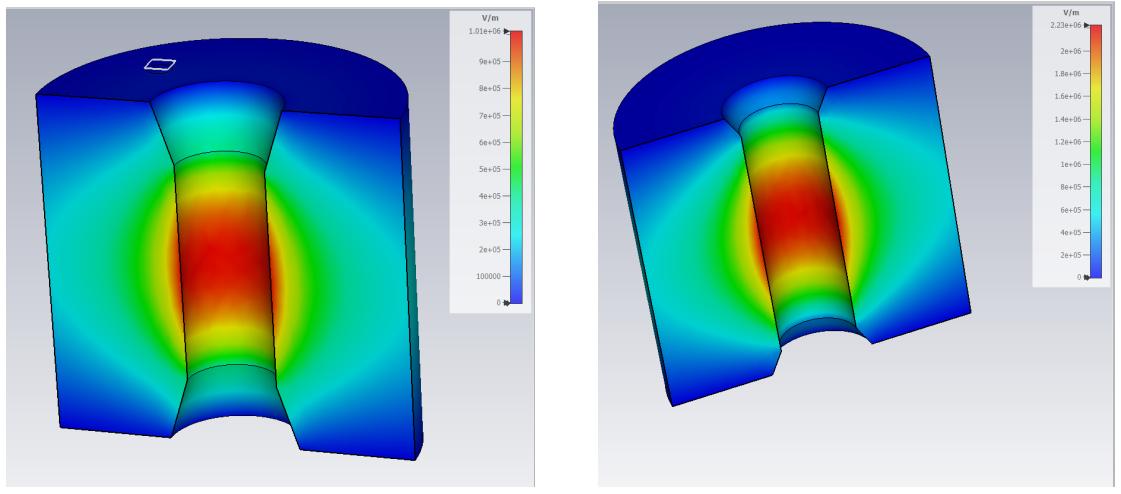
Figure 2.4: Poisson SUPERFISH field results of truncated cavities



(a) 107.5 MHz truncated

(b) 180 MHz truncated

Figure 2.5: CST eigenmode results of truncated cavities



(a) 107.5 MHz truncated

(b) 180 MHz truncated

Figure 2.6: CST electric field results of truncated cavities

1367 All calculated values below refer to the mesh geometry only.	
1368 Field normalization (NORM = 0):	EZERO = 1.00000 MV/m
1369 Frequency	= 107.4987 MHz
1370 Normalization factor for E0 = 1.0000 MV/m	= 10001.202
1371 Stored energy	= 0.0087215 Joules/cm
1372 Using standard room-temperature copper.	
1373 Surface resistance	= 2.70007 milliohm
1374 Normal-conductor resistivity	= 1.21410 microohm-cm
1375 Operating temperature	= 20.0000 C
1376 Power dissipation	= 120.2895 W/cm
1377 Q = 48972.0 Shunt impedance	= 4697.002 Ohm/m
1378 r/Q = 54190.222 Ohm Wake loss parameter	= 9.15051 V/pC
1379 Average magnetic field on the outer wall	= 2654.74 A/m, 953.183 miliGauss
1380 Maximum H (at X,Y = 61.05,69.72)	= 2654.52 A/m, 953.03 miliGauss
1381 Maximum E (at X,Y = 75.3,0.0)	= 0.999958 MV/m, 0.085835 Kilp.
1382 Ratio of peak fields Emax/Eavg	= 3.3359 mT/(MV/m)
1383 Peak-to-average ratio Emax/E0	= 1.0000
1384	
1385 Wall segments:	
1386 Segment Xend Vend Emax Power P/A dF/dX dF/dY	
1387 (cm) (cm) (MV/m) (W) (W/cm^2) (MHz/mm) (MHz/mm)	
1388 -----	
1389 1 18.800 69.720 0.9999 33.22 476.5 1.1154E-06 0.000	
1390 2 75.300 69.720 7.2838E-03 53.85 953.0 0.000 -0.1542	
1391 3 75.300 0.0000 1.0000 33.22 476.5 1.1140E-06 0.000	
1392	
1393 Total 120.3	
1394	
1395	
1396	
1397	
1398	
1399	
1400	
1401	
1402	
1403	
1404	
1405	
1406	
1407	
1408	
1409	
1410	
1411	
1412	
1413	
1414	
1415	
1416	
1417	
1418	
1419	
1420	
1421	
1422	
1423	
1424	
1425	
1426	
1427	
1428	
1429	
1430	
1431	
1432	
1433	
1434	
1435	
1436	
1437	
1438	
1439	
1440	
1441	
1442	
1443	
1444	
1445	
1446	
1447	
1448	
1449	
1450	
1451	
1452	
1453	
1454	
1455	
1456	
1457	
1458	
1459	
1460	
1461	
1462	
1463	
1464	
1465	
1466	
1467	
1468	
1469	
1470	
1471	
1472	
1473	
1474	
1475	
1476	
1477	
1478	
1479	
1480	
1481	
1482	
1483	
1484	
1485	
1486	
1487	
1488	
1489	
1490	
1491	
1492	
1493	
1494	
1495	
1496	
1497	
1498	
1499	
1500	
1501	
1502	
1503	
1504	
1505	
1506	
1507	
1508	
1509	
1510	
1511	
1512	
1513	
1514	
1515	
1516	
1517	
1518	
1519	
1520	
1521	
1522	
1523	
1524	
1525	
1526	
1527	
1528	
1529	
1530	
1531	
1532	
1533	
1534	
1535	
1536	
1537	
1538	
1539	
1540	
1541	
1542	
1543	
1544	
1545	
1546	
1547	
1548	
1549	
1550	
1551	
1552	
1553	
1554	
1555	
1556	
1557	
1558	
1559	
1560	
1561	
1562	
1563	
1564	
1565	
1566	
1567	
1568	
1569	
1570	
1571	
1572	
1573	
1574	
1575	
1576	
1577	
1578	
1579	
1580	
1581	
1582	
1583	
1584	
1585	
1586	
1587	
1588	
1589	
1590	
1591	
1592	
1593	
1594	
1595	
1596	
1597	
1598	
1599	
1600	
1601	
1602	
1603	
1604	
1605	
1606	
1607	
1608	
1609	
1610	
1611	
1612	
1613	
1614	
1615	
1616	
1617	
1618	
1619	
1620	
1621	
1622	
1623	
1624	
1625	
1626	
1627	
1628	
1629	
1630	
1631	
1632	
1633	
1634	
1635	
1636	
1637	
1638	
1639	
1640	
1641	
1642	
1643	
1644	
1645	
1646	
1647	
1648	
1649	
1650	
1651	
1652	
1653	
1654	
1655	
1656	
1657	
1658	
1659	
1660	
1661	
1662	
1663	
1664	
1665	
1666	
1667	
1668	
1669	
1670	
1671	
1672	
1673	
1674	
1675	
1676	
1677	
1678	
1679	
1680	
1681	
1682	
1683	
1684	
1685	
1686	
1687	
1688	
1689	
1690	
1691	
1692	
1693	
1694	
1695	
1696	
1697	
1698	
1699	
1700	
1701	
1702	
1703	
1704	
1705	
1706	
1707	
1708	
1709	
1710	
1711	
1712	
1713	
1714	
1715	
1716	
1717	
1718	
1719	
1720	
1721	
1722	
1723	
1724	
1725	
1726	
1727	
1728	
1729	
1730	
1731	
1732	
1733	
1734	
1735	
1736	
1737	
1738	
1739	
1740	
1741	
1742	
1743	
1744	
1745	
1746	
1747	
1748	
1749	
1750	
1751	
1752	
1753	
1754	
1755	
1756	
1757	
1758	
1759	
1760	
1761	
1762	
1763	
1764	
1765	
1766	
1767	
1768	
1769	
1770	
1771	
1772	
1773	
1774	
1775	
1776	
1777	
1778	
1779	
1780	
1781	
1782	
1783	
1784	
1785	
1786	
1787	
1788	
1789	
1790	
1791	
1792	
1793	
1794	
1795	
1796	
1797	
1798	
1799	
1800	
1801	
1802	
1803	
1804	
1805	
1806	
1807	
1808	
1809	
1810	
1811	
1812	
1813	
1814	
1815	
1816	
1817	
1818	
1819	
1820	
1821	
1822	
1823	
1824	
1825	
1826	
1827	
1828	
1829	
1830	
1831	
1832	
1833	
1834	
1835	
1836	
1837	
1838	
1839	
1840	
1841	
1842	
1843	
1844	
1845	
1846	
1847	
1848	
1849	
1850	
1851	
1852	
1853	
1854	
1855	
1856	
1857	
1858	
1859	
1860	
1861	
1862	
1863	
1864	
1865	
1866	
1867	
1868	
1869	
1870	
1871	
1872	
1873	
1874	
1875	
1876	
1877	
1878	
1879	
1880	
1881	
1882	
1883	
1884	
1885	
1886	
1887	
1888	
1889	
1890	
1891	
1892	
1893	
1894	
1895	
1896	
1897	
1898	
1899	
1900	
1901	
1902	
1903	
1904	
1905	
1906	
1907	
1908	
1909	
1910	
1911	
1912	
1913	
1914	
1915	
1916	
1917	
1918	
1919	
1920	
1921	
1922	
1923	
1924	
1925	

```

-----  

930 All calculated values below refer to the mesh geometry only.  

Field normalization (NORM = 0): EZERO = 1.00000 MV/m  

940 Frequency = 107.50055 MHz  

Normalization factor for E0 = 1.0000 MV/m = 10177.202  

944 Stored energy = 0.0087482 Joules/cm  

Using standard room-temperature copper.  

945 Surface resistance = 2.70499 milliohm  

Normal-conductor resistivity = 1.72410 microOhm-cm  

948 Operating temperature = 20.0000 C  

Power dissipation = 117.3652 W/cm  

949 Q = 50246.6 Shunt impedance = 4911.634 Mohm/m  

r/Q = 54024.066 Ohm Wake loss parameter = 9.12258 V/pC  

Average magnetic field on the outer wall = 2685.64 A/m, 975.599 mT/cm^2  

Maximum H (at X,Y = 73.55,71) = 2707.68 A/m, 991.587 mT/cm^2  

Maximum E (at X,Y = 75.3,0.8) = 1.00401 MV/m, 0.086182 Kilp.  

Ratio of peak fields Bmax/Emax = 3.3890 mT/(MV/m)  

Peak-to-average ratio Emax/E0 = 1.0040  

950 Wall segments:  

951 Segment Xend Vend Emax Power P/A dF/dx dF/dy  

952 (cm) (cm) (MV/m) (W) (W/cm^2) (MHz/mm) (MHz/mm)  

953 -----  

954 18.000 0.0000  

955 1 18.000 51.400 0.9956 16.34 317.9 4.2318E-02 0.000  

956 2 26.800 71.000 0.3137 18.47 872.4 -4.5918E-02 -1.8742E-02  

957 3 75.300 71.000 4.0276E-02 47.31 975.5 0.000 -0.1358  

958 4 75.300 0.0000 1.004 35.25 496.4 -3.5967E-03 0.000  

959 Total 117.4  

-----  

960 All calculated values below refer to the mesh geometry only.  

Field normalization (NORM = 0): EZERO = 1.00000 MV/m  

961 Frequency = 180.00245 MHz  

962 Normalization factor for E0 = 1.0000 MV/m = 10088.517  

963 Stored energy = 0.0031145 Joules/cm  

964 Using standard room-temperature copper.  

965 Surface resistance = 3.50026 milliohm  

966 Normal-conductor resistivity = 1.72410 microOhm-cm  

967 Operating temperature = 20.0000 C  

968 Power dissipation = 90.5379 W/cm  

969 Q = 38995.8 Shunt impedance = 372.719 Mohm/m  

970 r/Q = 32337.101 Ohm Wake loss parameter = 9.14879 V/pC  

971 Average magnetic field on the outer wall = 2665.59 A/m, 1.24351 W/cm^2  

972 Maximum H (at X,Y = 43.5,42) = 2678.17 A/m, 1.2553 W/cm^2  

973 Maximum E (at X,Y = 45,0,0) = 1.00154 MV/m, 0.070842 Kilp.  

974 Ratio of peak fields Bmax/Emax = 3.3603 mT/(MV/m)  

975 Peak-to-average ratio Emax/E0 = 1.0015  

976 -----  

977 Wall segments:  

978 Segment Xend Vend Emax Power P/A dF/dx dF/dy  

979 (cm) (cm) (MV/m) (W) (W/cm^2) (MHz/mm) (MHz/mm)  

980 -----  

981 11.250 0.0000  

982 1 11.250 35.250 0.9981 17.74 0.5033 7.6469E-02 0.000  

983 2 15.000 42.000 0.1761 9.122 1.181 8.1176E-02 -4.5098E-02  

984 3 45.000 42.000 3.1924E-02 37.29 1.243 0.000 -0.3870  

985 4 45.000 0.0000 1.002 26.38 0.6282 -4.6971E-03 0.000  

986 Total 90.54  

987 -----  

988

```

(a) 107.5 MHz truncated

(b) 180 MHz truncated

Figure 2.8: Poisson Superfish calculations with *truncated* cavities

One can observe the shunt impedance gain between truncated and unmodified cavities in *figures 2.7 and 2.8*;

$$\Delta Z_{107.5} = 2.5\% \quad \Delta Z_{180} = 2.7\%$$

2.2. Magnet Design

Magnet design in rhodotron type accelerators depend heavily on the design of the cavities. Because the limiting factors usually are frequency and total volume of the accelerator, magnet design parameters are considered after cavity design has been completed. Considering the nature of coaxial cavity, simplest path for beams inside magnets is major segment of a circle, which was used as the reference for following discussions.

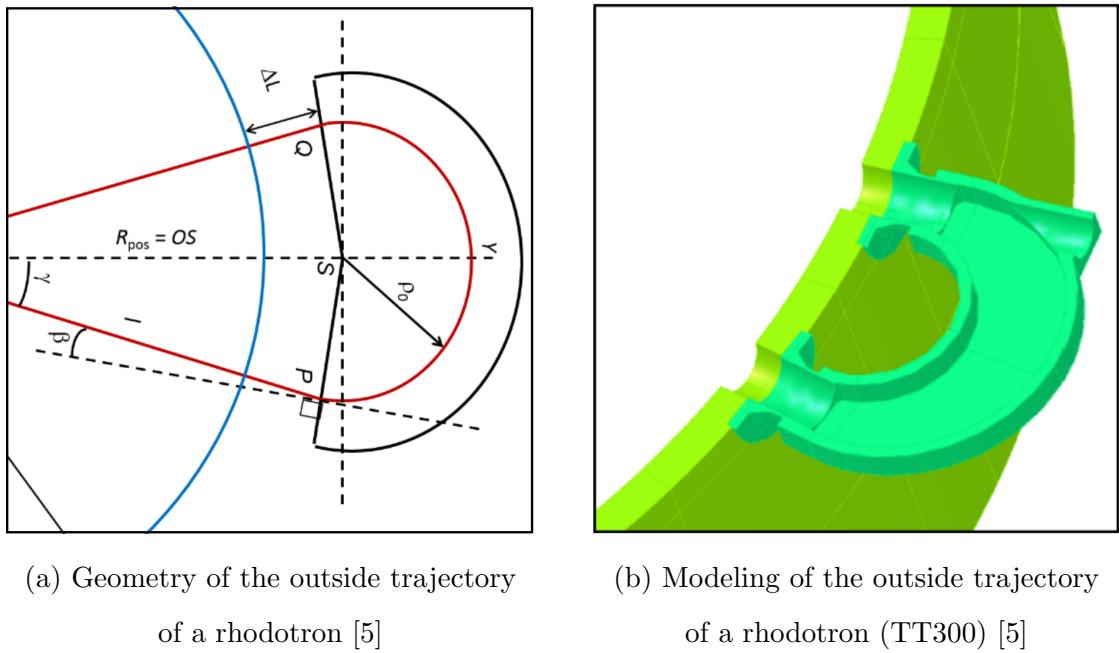


Figure 2.9: From "Design and Simulation Tools for the High-Intensity Industrial Rhodotron Electron Accelerator," by W.Kleeven, M.Abs, J.Brison, E.Forton, J.Hubert and J.Van de Walle

In rhodotron accelerators, magnets are used for not only guiding the electron beam to desired path, but also for synchronizing them with the RF field in the cavity. This synchronization, also called phase stability, is the fundamental constraint of magnet design. Time spent outside of the cavity should be precisely tuned to maximize acceleration in the succeeding pass.

Two different approaches have been used to achieve phase stability. These ap-

proaches are discussed further in the following sections.

2.2.1. $n\lambda$ Technique

First approach for phase stability considiration assumes that from the start of the current pass, $\beta_{avg} \approx 1$ for a synchronous particle.

To maintain the phase synchronization, a particle that started the current pass at $t = 0$, should start the next pass at $t = nT$, where n is an integer and T is the period of RF field inside cavity. As discussed earlier, by assuming the electron is traveling at c , phase stability can be achieved by designing the whole trajectory to be $n\lambda$, where λ is the wavelength of the RF field. In other words, total trajectory of synchronous particle needs to satisfy the following equality;

$$L_{pass} = n\lambda$$

Since $n > 1$ would require unnecessarily long and expensive beam guide and magnets, $n = 1$ is the most efficient choice.

$$L_{pass} = \lambda \quad (2.2)$$

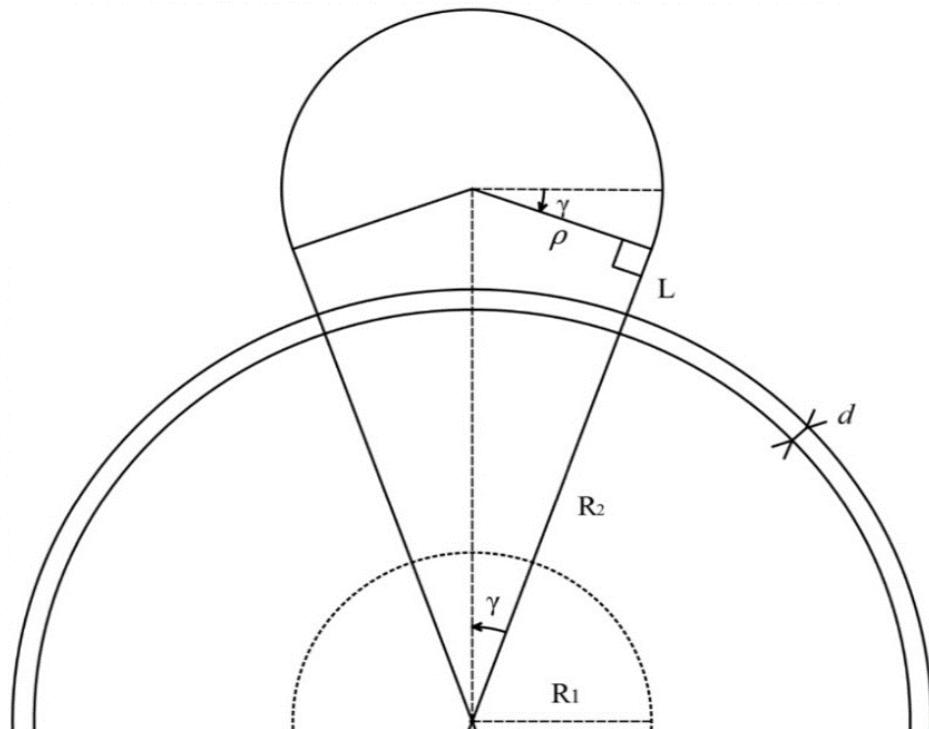


Figure 2.10: Trajectory of a particle in single pass

From *figure 2.10*, where γ is half the angle between trajectories of successive passes, ρ is the radius of the circular path inside magnet, L is the length of the magnet guide, d is the thickness of the cavity wall;

$$\begin{aligned}
 L_{in} &= 2R_2 \\
 L_{out} &= 2L + 2d + (\pi + 2\gamma)\rho \\
 L_{pass} &= L_{in} + L_{out} \\
 \lambda &= 2R_2 + 2d + 2L + (\pi + 2\gamma)\rho
 \end{aligned} \tag{2.3}$$

After the cavity design is complete, only L , γ and ρ remain, which are the design parameters of magnets. Observing the *figure 2.10* and using *equation 2.3*,

$$\begin{aligned}
\tan(\gamma) &= \frac{\rho}{R_2 + d + L} \\
\rho &= \tan(\gamma)(R_2 + d + L) \\
2\rho &= \tan(\gamma)(\lambda - (\pi + 2\gamma)\rho)
\end{aligned}
\tag{2.4}$$

$$\begin{aligned}
\rho(2 + \tan(\gamma)(\pi + 2\gamma)) &= \tan(\gamma)\lambda \\
\rho &= \frac{\lambda}{\pi + 2\gamma + \frac{2}{\tan(\gamma)}}
\end{aligned}
\tag{2.4}$$

$$L + d = \frac{\rho}{\tan(\gamma)} - R_2
\tag{2.5}$$

Equations 2.4 and 2.5 define 2 constraints. As already discussed, it has been assumed that cavity design is set (*i.e* R_2 , λ , d are already defined). Therefore only the magnet design parameters ρ , γ and L are left. Together with the constraints, one free variable defines the whole magnet design. Considering the importance of γ for maximum number of possible passes, it will be used as the free magnet design variable in the further discussions.

2.2.2. L_{out} Parameter Sweep

As mentioned previously in *section 2.2.1*, *equation 2.3* assumes that the particles are fast enough so that $\Delta t_{pass} \approx nT$. This assumption is not guaranteed however, especially in the first few passes if the particles are not accelerated enough. This scenario can happen when RF power is not sufficiently high.

Consider an RF supply with $f = 1$ GHz, $P = 50$ kW. Using *equation 1.14*, after the first pass the synchronous electron that entered the accelerator with 40 keV and $\phi = 15^\circ$ phase lag,

$$W_{gain} = 2.14 \times 0.2998^{1/4} \times 50000^{1/2} = 354keV$$

$$W_{total} = 394keV$$

$$\beta_{40keV} \approx 0.374$$

$$\beta_{394keV} \approx 0.825$$

It is clear that β_{avg} is not fast enough to sustain phase stability if the magnet is designed for $\beta = 1$. For these cases, *equation 2.3* fails to deliver phase stability. Another approach for designing a magnet would be to find a new constraint, K for total path length.

$$L_{pass} = K$$

$$L_{in} + L_{out} = K$$

Since L_{in} is set previously from cavity design, we can remove it from our equation and continue with

$$L_{out} = K \quad (2.6)$$

To start, an optimization criteria for the following pass such as

- Ensure the phase stability of the synchronous electron
- Maximize the energy gain of the synchronous electron
- Minimize the energy spread of the beam
- Minimize the phase lag spread of the beam
- All of the above with decided weights

needs be selected. Using a simulation software such as *CST Studio Particle Module*, the optimum value of K for the criteria can be found by simulating for different values of K (*sweeping*) and analyzing the results. This process can be repeated for each magnet until the desired beam characteristics in the end of the accelerator are achieved.

However, one caviat of this technique is that, available simulation softwares are not well suited for this kind of process. As mentioned above, *CST Studio Particle Module* has a *parameter sweep* functionality. But the calculation times are too slow to be useful in this particular problem. *A custom built software offering magnet optimizing sweep functionality will be discussed in the following chapter.*

2.3. Initial Design at KAHVELab

A rhodotron with operation frequency of 107.5 MHz was decided to be built at KAHVELab. After considering the earlier simulations mentioned in *sections 2.1 and 2.2*, an initial design was created.

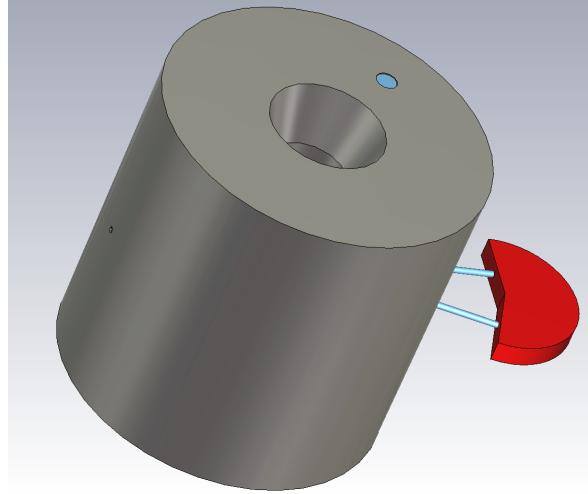
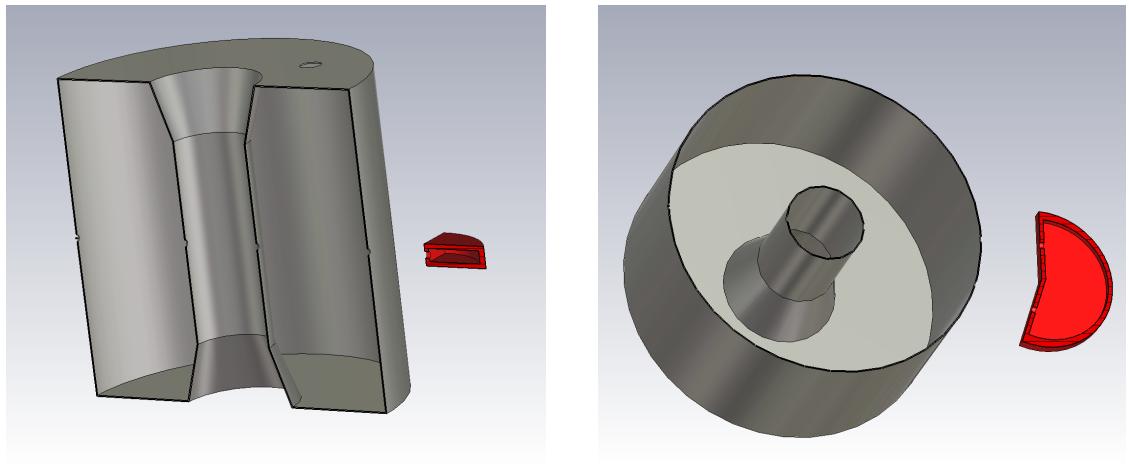


Figure 2.11: Initial design of the proposed rhodotron

$$\gamma = 9^\circ$$



(a) Axial cross section.

(b) Cross section at acceleration plane.

Figure 2.12: Initial design of the proposed rhodotron.

The magnets used in this design is the simplest in terms of design parameters. It consists of an iron casing encapsulating the shape of desired magnetic field, which are

created by two coils inside.

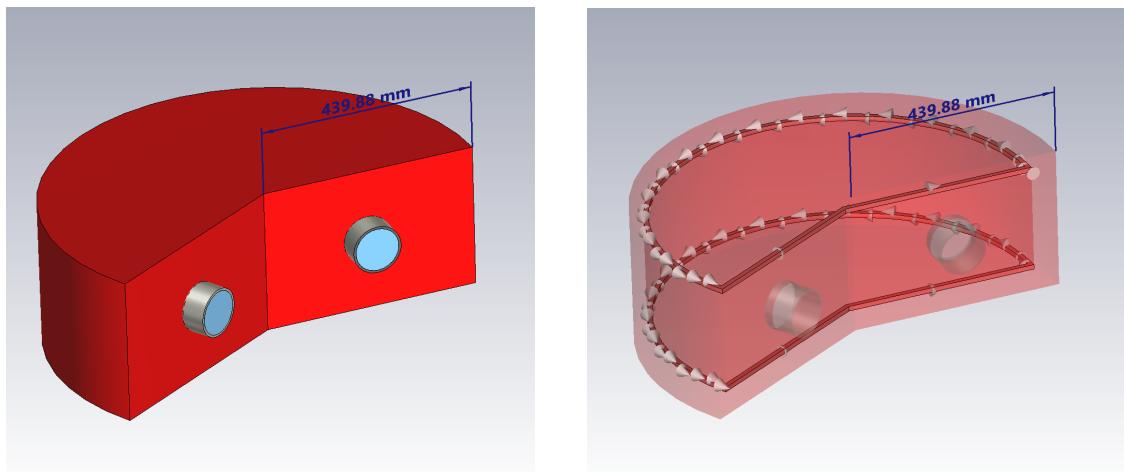


Figure 2.13: Initial magnet design.

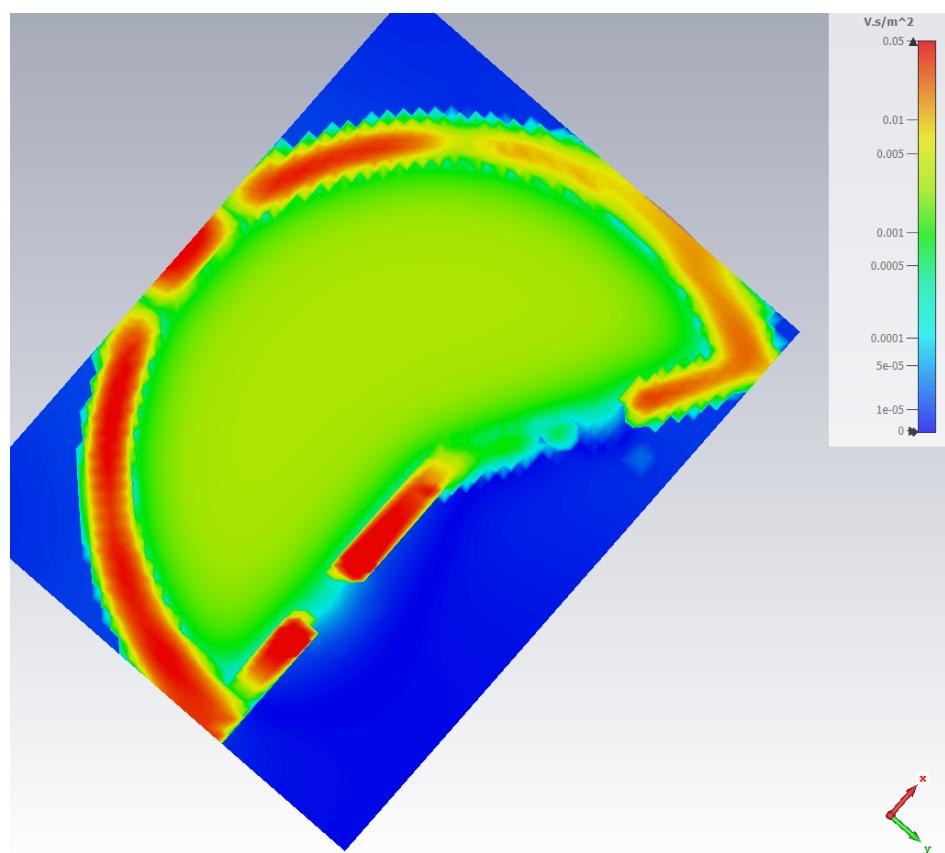


Figure 2.14: Magnetic field in acceleration plate of initial magnet design.

figure 2.14 shows that this design provides relatively uniform magnetic field inside, although magnetic field gradient in the openings can be improved.

After adding two more magnets and using $\gamma = 15^\circ$, the following second prototype design was created

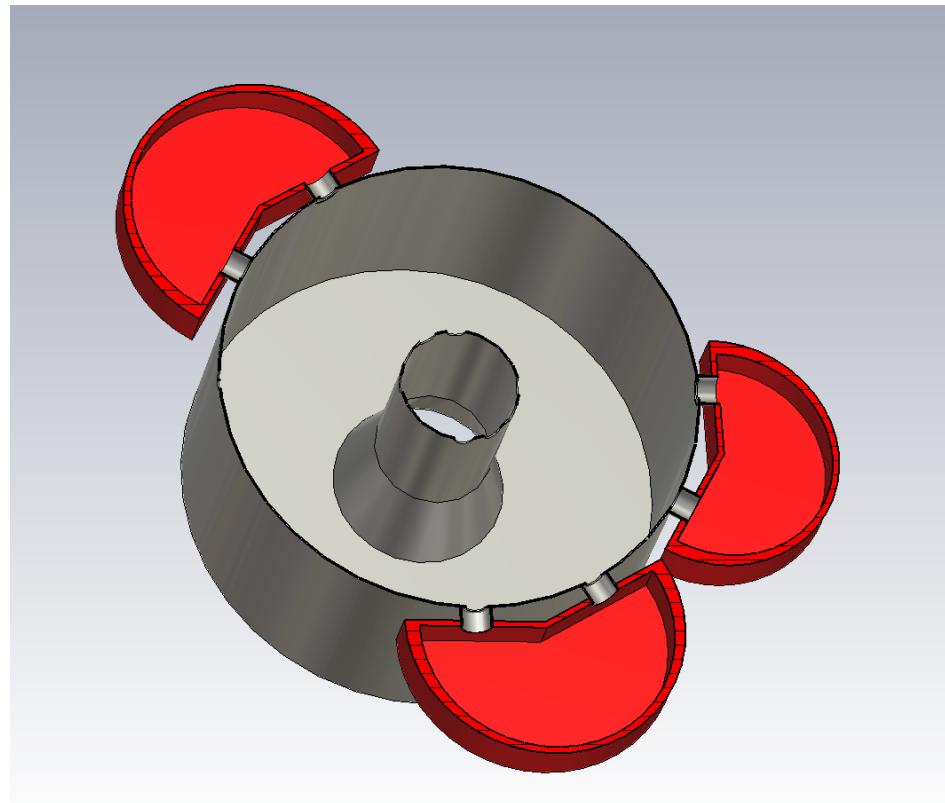
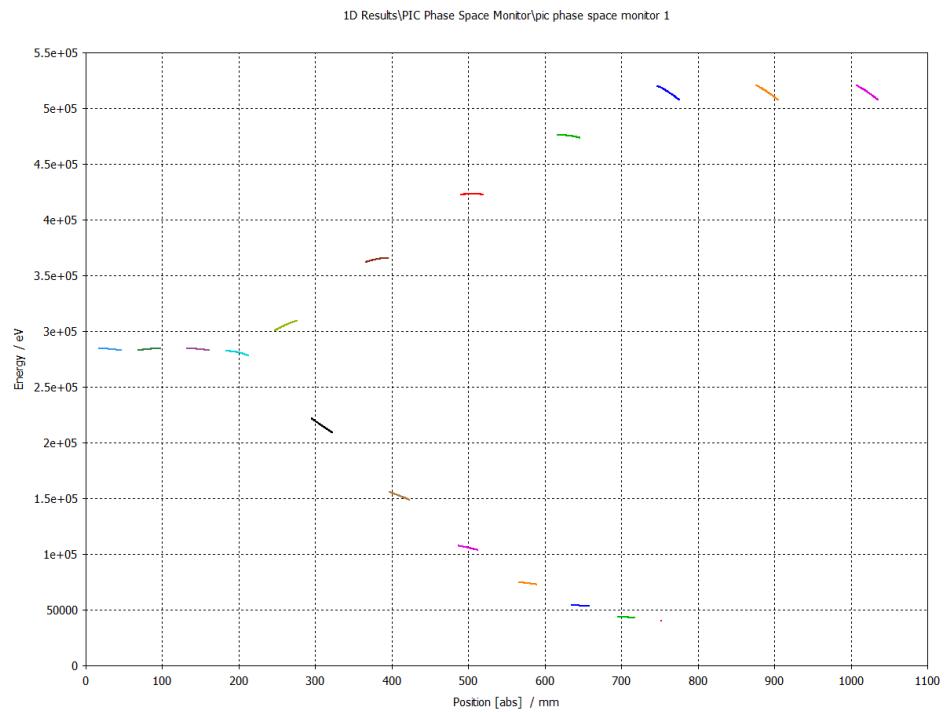


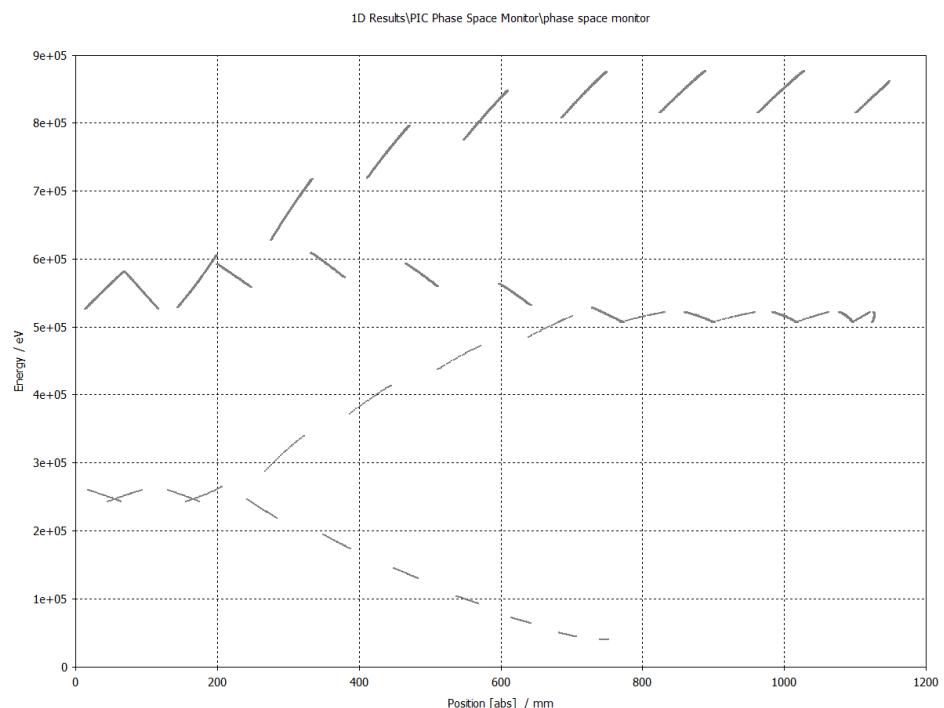
Figure 2.15: Initial cavity with three magnets as shown in *figure 2.13*

$$\gamma = 15^\circ$$

These designs, together with 40 keV e^- gun injecting with phase lag of 15° for 0.1 ns, was simulated at RF power of 12 kW using *CST Studio Particle Module PIC Solver*. Results of these simulations can be found in *figure 2.16*.



(a) One pass simulation of figure 2.11



(b) Two pass simulation of figure 2.15

Figure 2.16: $|r|$ vs *Energy* simulation of initial design.

The behavior of the beam in the *figure 2.16* after the first pass indicates that phase stability is not maintained. When the simulation results are examined further, it was observed that the beam started the second pass at $t \approx 12$ ns after the injection. Considering the starting phase lag $\phi_{lag} = 15^\circ$, this would mean phase lag of the second pass was $\phi_{lag2} = 120^\circ$. This result can be observed in *figure 2.16*, as the beam decelerates after a short acceleration in the beginning of the second pass.

After various similar observations in *CST* simulations, underlying cause of this problem was thought to be the $L_{pass} = n\lambda$ approach on magnet design failing in low energies, as anticipated and discussed earlier in the *section 2.2.2*. *The necessity for an alternative tool, capable of aiding researchers in improving beam behavior, was deemed evident.*

In KAHVELab, several improvements were made to these initial designs by *Mr. Sinan ÖZ*. They include curved edges on cavity which reduce the power loss on cavity walls, and redesigned iron casing on magnets for ease of production and maintenance. Improved designs can be seen in the figures below.

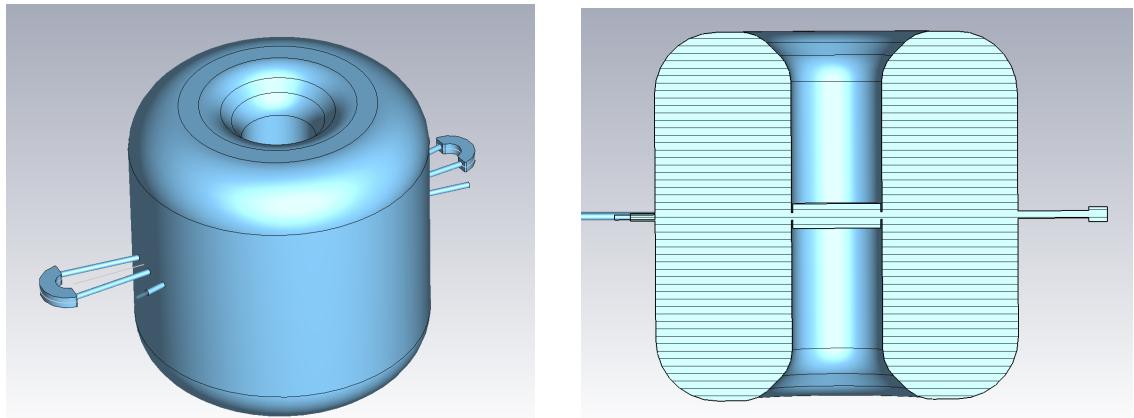


Figure 2.17: Improved cavity design.

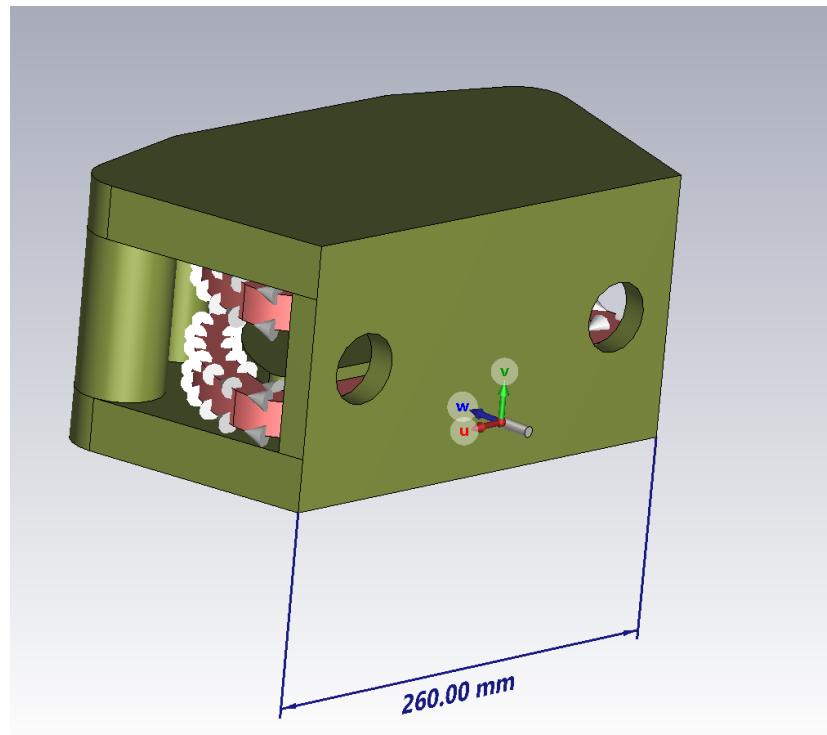


Figure 2.18: Improved magnet design.

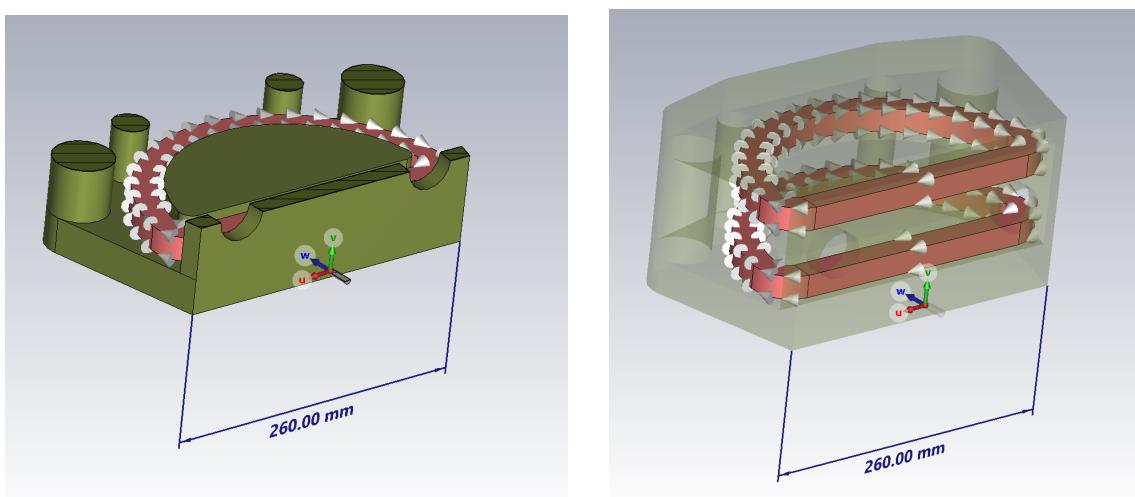


Figure 2.19: Cross section and coils of improved magnet design.

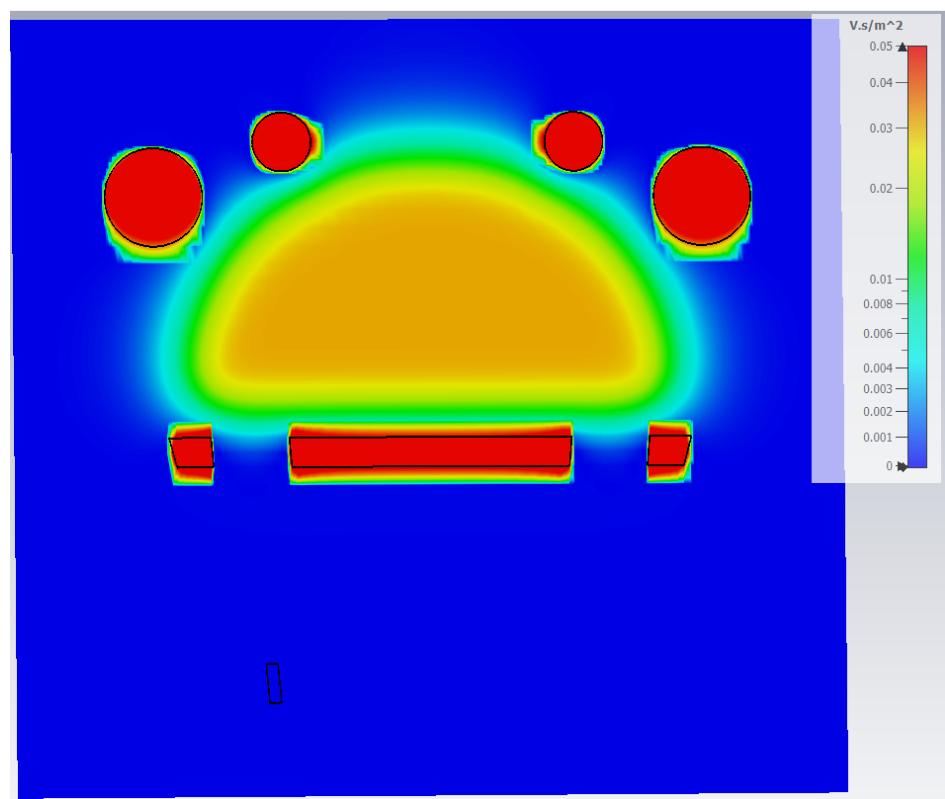


Figure 2.20: Magnetic field in acceleration plate of improved magnet design.

From figures 2.14 and 2.20, it can be observed that improved magnet design reduces magnetic field leak in enter and exit openings considerably. This reduction helps containing the spread of entering beam, decreasing the amount of focusing needed further into trajectory.

3. SIMULATION

3.1. Proof of Concept

Once the need for a custom built tool for designing and simulating a Rhodotron type accelerator was apparent, a simple 1D proof of concept was implemented.

This proof of concept worked with interacting an electron with the EM field provided, at discrete time steps dt . The core sequence of this software is provided below. Where \vec{r} is the position, \vec{v} is the velocity of the e^- ;

- Get electron energy E_{in}
- Get the RF field definitions
- Get the magnet design parameters if there is any
- Do following until the simulation time has ellapsed
 - (i) Calculate $\vec{E}_{||}(\vec{r})$
 - (ii) Calculate $\vec{a}_{E||}$ using *equation 1.11*
 - (iii) Calculate $\vec{r}(t + dt)$ using *equation 1.16*
 - (iv) Calculate $\vec{v}(t + dt)$ using *equation 1.17*
 - (v) Calculate new E_t

The implemenation of this logic can be observed in *figure 3.1*.

```

1  for (double t=0; t<SimuTime; t+=dT){
2      double RelBeta = v/c;
3      double RelGamma = 1.0 / sqrt(1.0-RelBeta*RelBeta);
4
5      double ef=Eradiation(r_pos*1000,t,0); // convert position to mm
6      double acc=ef*1E6*eQMratio/(RelGamma*RelGamma*RelGamma);
7
8      r_pos = r_pos + v * dT*ns + 1/2*acc*(dT*ns)*(dT*ns);
9      v = v + acc*dT*ns;
10
11     RelBeta = v/c;
12     RelGamma = 1.0 / sqrt(1.0-RelBeta*RelBeta);
13     Et=RelGamma*E0;
14 }
```

Figure 3.1: Core logic loop of the POC

After simulating the *synchronous electron* mentioned in *section 1.2.4.1* with $\phi_{lag} = 15^\circ$ for one pass in THE CAVITY, the results from *POC*, *CST* and *equation 1.13* are compared.

$$\begin{aligned}
 E_{Pottier} &= 0.565\text{MeV} \\
 E_{CST} &= 0.872\text{MeV} \\
 E_{POC} &= 0.873\text{MeV}
 \end{aligned} \tag{3.1}$$

$$P = 40\text{kW}, R_1 = 0.188\text{m}, R_2 = 0.753\text{m}, f = 107.5\text{MHz}$$

As can be observed in *equation 3.1*, *POC software* and *CST* produced very similar results. However, we cannot make the same claim regarding *equation 1.13*. This inconsistency between $E_{Pottier}$ and E_{CST} was noticed in the earlier simulations as well, which supported the idea of using another simulation software to reduce reliance on *CST*. Using a single software would have made the process error prone. Their results,

at the end of the day, depend heavily on the physical and mathematical models they use.

After the promising results obtained from *POC*, the decision was made to develop a more robust simulation software. This software was named *Rhodotron Simulation* for the current stage and started development in October, 2021. The development and improvement efforts are still ongoing.

During the development of *Rhodotron Simulation* software, several intermediate builds have been implemented and tested. The following sections in this chapter investigates their implementations further.

3.2. Intermediate Versions

3.2.1. L_{out} Optimization For Single e^-

Initial step of improving *POC* towards *RhodotronSimulation* was to implement L_{out} optimizations to help optimazing magnet designs, as discussed in *section 2.2.2*. First approach was to hang the e^- outside of the cavity for $t_{out} = L_{out}/v$, then inject it back to the cavity with reversed \vec{v} . Then sweep the t_{out} parameter to find the optimal value. This simple implementation can be found in *figure A.1 of Appendix A*.

Although the results from this optimization sweep were promising after they were simulated with *CST*, simulating one particle would not be sufficiently useful for designing a magnet.

3.2.2. ϕ_{lag} Optimization For Bunches

After successfully accelerating single e^- , particle bunches were implemented to approximate a real e^- gun. They were modeled as N electrons fired from an e^- gun at even time intervals. This approach was taken because the amount of time gun fires, defined as *Gun Active Time*, t_g , is a crucial part of pulsing e^- gun design.

Addition of bunches would immediately proven useful when finding optimal gun phase lag. ϕ_{lag} for a bunch was defined as the RF phase when the first e^- of the bunch entered the cavity, it defines the starting time of the current pass. To use the *parameter sweep* method, as used in L_{out} optimization, relevant bunch characteristics are defined as follows:

- μE : Average energy
- E_{rms} : Root mean square of energy
- R_{rms} : Root mean square of e^- positions

Optimal ϕ_{lag} would produce maximum μE while minimizing E_{rms} & R_{rms} . For the first pass, E_{rms} and R_{rms} would be vaguely dependent of each other; therefore, early implementation of ϕ_{lag} sweep was based on minimizing E_{rms} during simulation (*see figure 3.2*). For μE considerations, data from the software would be analyzed either manually or by using external tools such as *ROOT*.

```

1 int phase_opt(int phase_sweep_range){
2     double minrms = 1;
3     int opt_phase;
4     for(int RFphase = -phase_sweep_range; RFphase <= phase_sweep_range; RFphase++){
5         Bunch bunch1(RFphase);
6         double t1 = 0;
7         bunch1.bunch_gecis_t(t1);
8         bunch1.reset_pos();
9
10        if( bunch1.E_rms() < minrms ){
11            minrms = bunch1.E_rms();
12            opt_phase = RFphase;
13        }
14    }
15    return opt_phase;
16 }
```

Figure 3.2: ϕ_{lag} Optimization For Initial Bunch Design

Since ϕ_{lag} is relatively easy to change after production, another version of *figure 3.2* that was modified for given magnet design parameters was also implemented (*see figure A.2 in Appendix A*). This version can be useful for optimizing ϕ_{lag} in case of production issues in magnets.

After the bunch and ϕ_{lag} sweep implementations, L_{out} sweep was also updated to minimize E_{rms} . ρ and L calculations, using *equations 2.4 and 2.5*, were also integrated. Two example runs can be found in *figures B.1 and B.2 of Appendix B*.

3.2.3. Simulation in 3D

After successfully implementing $e^- - \vec{E}$ interaction in 1D and confirming the usefulness of this tool, the decision was made to proceed with implementing a 3D version of the *Rhodotron Simulation*. Although complete refactoring of the software was necessary, this upgrade was crucial for implementation of $e^- - \vec{B}$ interaction. The refactoring effort included proper implementation of *OOP*, details of which can be seen in *Appendix A*.

Magnets were modeled as major segments of a circle, defined with \vec{r}_{mag} , \mathbf{R} and $|\mathbf{B}|$. For the initial implementation, \vec{B} assumed to be uniform and has no leaks outside the magnet boundary (See *figure A.4* in *Appendix A*).

Interaction logic for $e^- - \vec{E}$ and $e^- - \vec{B}$ in 3D can be found in *figure 3.3*.

```

1  vector3d CoaxialRFField::actOn(Electron& e){
2      vector3d Efield = getField(e.pos);                      // Calculate E vector
3      vector3d F_m = Efield*1E6*eQMratio;                  // Calculate F/m vector
4      vector3d acc = (F_m - e.vel*(e.vel*F_m)/(c*c))/e.gamma(); // Calculate a vector
5      return acc;
6  }



---

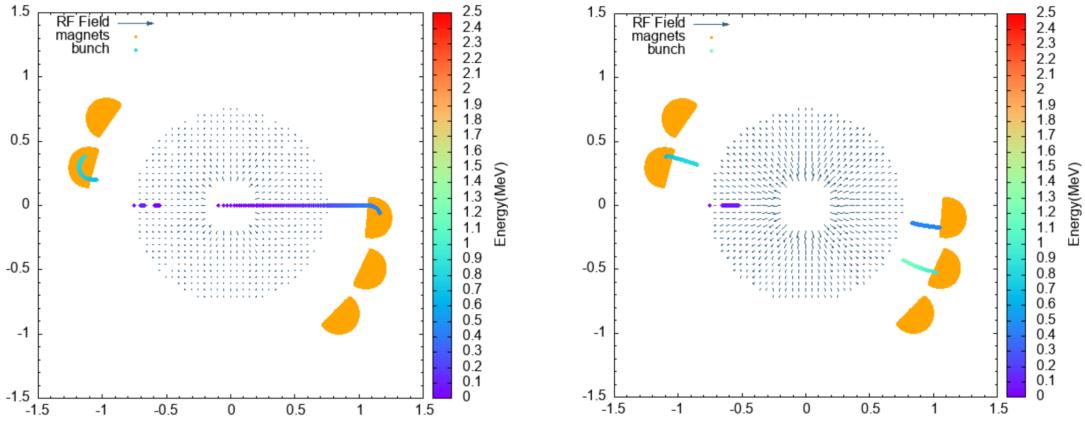

1  vector3d MagneticField::actOn(Electron& e){
2      if (isInside(e.pos) == -1)
3          return vector3d(0,0,0);
4      vector3d Bfield = getField(e.pos);                      // Calculate B vector
5      vector3d F_m = (e.vel % Bfield)*eQMratio;            // Calculate F/m vector
6      vector3d acc = (F_m)/e.gamma();                        // Calculate a vector
7      return acc;
8 }
```

Figure 3.3: $e^- - \mathbf{EM}$ interaction logic from *equation 1.12*

Where $*$ and $\%$ are, dot-product and cross-product respectably. (See *figure A.3* of *Appendix A*)

Simulating in 3D had one other benefit; it was now possible to visualize the

results by rendering the interaction data. For this purpose, *gnuplot* was integrated into *Rhodotron Simulation* to produce 2D visualization of acceleration plane. Rendered results could be stored as *gif* animations. Two of such renders can be seen in *figure 3.4*.



Unsyncronized e^- gun period, $T_g = 5\text{ns}$ Synchronized e^- gun period, $T_g = 9.3\text{ns}$

Figure 3.4: Example *gnuplot* renders of *Rhodotron Simulation*

$$P = 12\text{kW}, f = 107.5\text{MHz}$$

3.2.4. Acceleration in Magnetic Field

An issue regarding the $e^- - \vec{B}$ interaction became apparent when energy gain during these interactions was observed. A setup simulation was implemented in which a bunch of $100e^-$ at 1MeV was fired to a uniform magnetic field of 0.1T placed in $x > 0.05\text{m}$. Initial results at $dt = 0.01\text{ns}$ proved the suspicion of $e^- - \vec{B}$ interaction being broken. However, the energy gain would decrease tremendously as dt decreased.

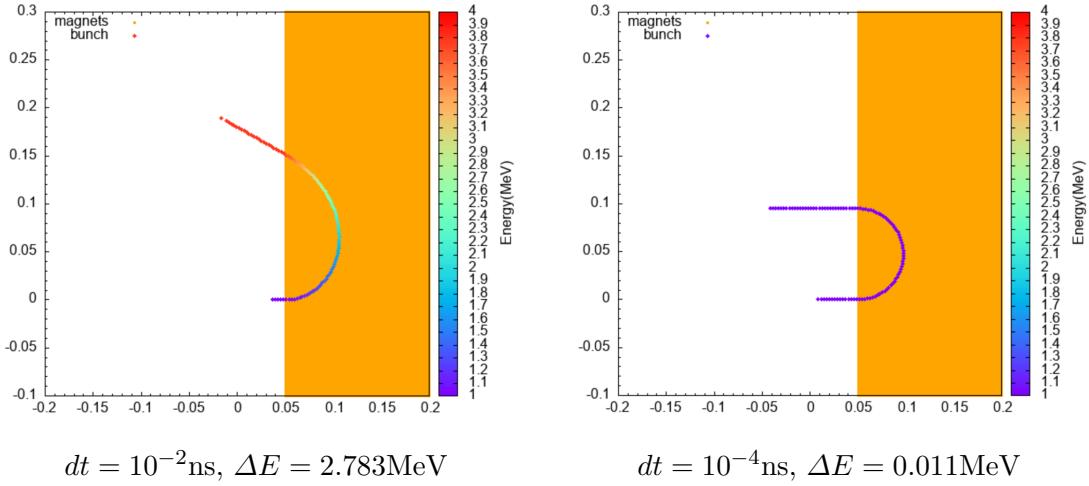


Figure 3.5: Energy gain of 1MeV bunch in $\mathbf{B}=0.1\text{T}$

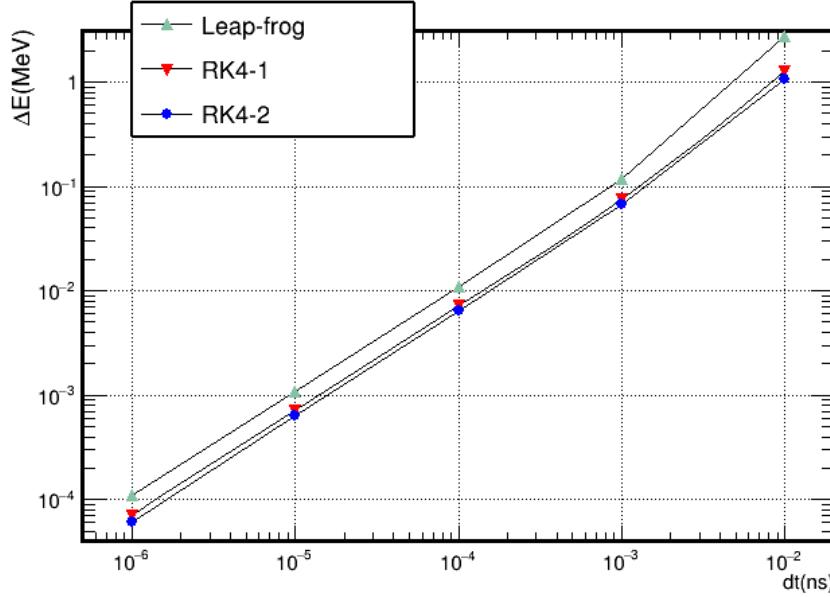
Decreasing dt would be the best way to increase accuracy of the results; however, this is not sustainable because time and computing power. Until this point, *Rhodotron Simulation* have been using *section 1.3.1* for \mathbf{e}^- - EM interactions. To test newer approaches, two additional version of \mathbf{e}^- - EM interaction that are using *section 1.3.2* were added.

3.2.4.1. RK4-1. First approach for integrating *section 1.3.2* into \mathbf{e}^- - EM interaction was to calculate $\vec{\mathbf{a}}_E$ and $\vec{\mathbf{a}}_B$ from *equation 1.12* using *RK4*. After $\vec{\mathbf{a}}_{EM} = \vec{\mathbf{a}}_E + \vec{\mathbf{a}}_B$ was calculated, \mathbf{e}^- would move and accelerate using the *Leap-frog* method. The idea was to produce more refined interaction results, leading to improved accuracy especially in \mathbf{e}^- - $\vec{\mathbf{B}}$. RF field was kept static during the RK4 computation, due to ongoing *multithreading* implementation efforts. The implementation can be found in *figures A.5 and A.6*.

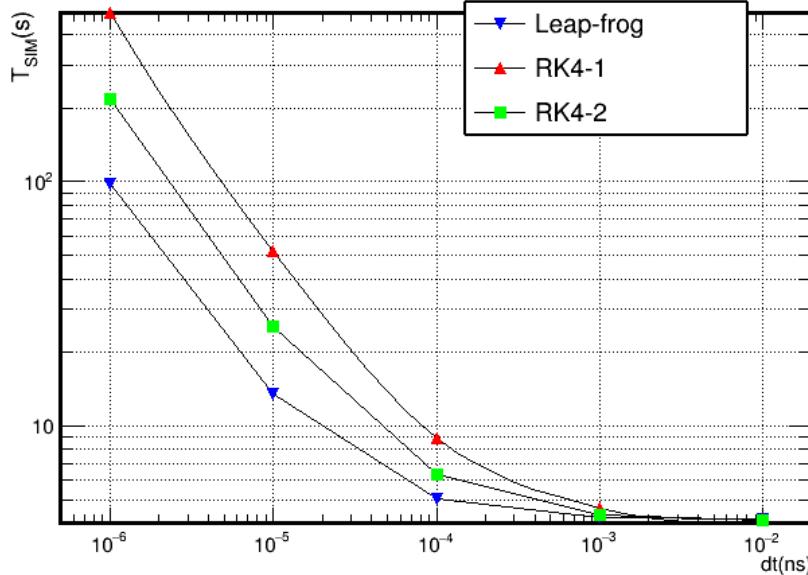
3.2.4.2. RK4-2. Following the implementation of *RK4-1*, revisions were made to the integration method for *RK4* to replace *Leap-frog*. Instead of calculating $\vec{\mathbf{a}}_{EM}$ using *RK4*, $\vec{\mathbf{r}}$ and $\vec{\mathbf{v}}$ would be determined directly.

These three methods were then tested in the same setup as *figure 3.5*. The results

can be found in *figure 3.6*.



dt vs ΔE



dt vs T_{sim}

Figure 3.6: Comparing Leap-frog, RK4-1, RK4-2 performance on $e^- - \vec{B}$ interaction

$$E_{in} = 1\text{MeV}, \mathbf{B}=0.1\text{T}, t_{end} = 5\text{ns}$$

The data from these tests can be found in *Tables B.1 and B.2 of Appendix B*. As

anticipated, *Leap-frog* produced the fastest yet least accurate results, whereas *RK4-2* method was proved to be the best performer for accuracy. When $dt = 10^{-5}\text{ns}$ was taken as reference point due to providing a good balance of accuracy and computational intensity, *RK4-2* provides around 41% less energy gain in $\vec{\mathbf{B}}$ with respect to the next performer *Leap-frog*. The tradeoff is the computational intensity mentioned in *section 1.3.2*; *Leap-frog* being around 47% faster than *RK4-2*.

This relation is not linear however; *RK4-2* is not as demanding in less computationally intensive runs and able to keep up with *Leap-frog* in execution times while increasing the accuracy difference from other methods, whereas *Leap-frog* excels at simplicity, being able to use higher time resolution for given computational power. Both methods can be used in different situations, therefore both were integrated into *Rhodotron Simulation* for the user to decide. *RK4-2* renders from these test can be found in *figure B.3 of Appendix B*.

APPENDIX A: Intermediate Codes

```

1  for(double i = 2; i < 9; i += dT_out){
2      t_dum += i;
3      double enow = gecis(r_pos, vel, Et, t_dum);
4      if( enow > maxE ){
5          maxE = enow;
6          t_opt = i;
7      }
8      t_dum = t;
9  }



---


1  double gecis(double r_pos, double vel, double Et, double &t){
2      for(; r_pos >= -R2 && r_pos <= R2 ; t+=dT){
3          vel = c*sqrt(Et*Et-E0*E0)/Et;
4          double RelBeta = vel/c;
5          double RelGamma = 1.0 / sqrt(1.0-RelBeta*RelBeta);
6
7          double ef=Eradiat(r_pos*1000,t,RFphase*deg_to_rad);
8
9          double acc=ef*1E6*eQMratio/(RelGamma*RelGamma*RelGamma);
10
11         r_pos = r_pos + vel * dT*ns + 1/2*acc*(dT*ns)*(dT*ns);
12         vel=vel+acc*dT*ns;
13         RelBeta = vel/c;
14         RelGamma = 1.0 / sqrt(1.0-RelBeta*RelBeta);
15         Et=RelGamma*E0;
16     }
17     return Et;
18 }
```

Figure A.1: L_{out} Optimization For Single e⁻

```

1 int phase_opt(const vector<double>& Louts, int phase_sweep_range){
2     double minrms = 1;
3     int opt_phase;
4     for(int RFphase = -phase_sweep_range; RFphase <= phase_sweep_range; RFphase++){
5         Bunch bunch1(RFphase);
6         double t1 = 0;
7         bunch1.bunch_gecis_t(t1);
8         bunch1.reset_pos();
9
10        for(int i = 0; i < Louts.size(); i++){
11            bunch1.bunch_gecis_d(Louts[i]);
12            bunch1.reset_pos();
13        }
14
15        if( bunch1.E_rms() < minrms ){
16            minrms = bunch1.E_rms();
17            opt_phase = RFphase;
18        }
19    }
20    return opt_phase;
21 }
```

Figure A.2: ϕ_{lag} Optimization For Initial Bunch Design

```

1 double vector3d::operator* (const vector3d& other){
2     double dot = 0;
3     dot += this->x * other.x;
4     dot += this->y * other.y;
5     dot += this->z * other.z;
6     return dot;
7 }

1 vector3d vector3d::operator% (const vector3d& other){
2     double x_ = (this->y * other.z) - (this->z * other.y);
3     double y_ = (this->z * other.x) - (this->x * other.z);
4     double z_ = (this->x * other.y) - (this->y * other.x);
5     vector3d crossed(x_, y_, z_);
6     return crossed;
7 }
```

Figure A.3: * and % operators of *vector3d* class

```

1  bool isInsideHalfSphere(vector3d e_position, double r, vector3d hs_position){
2      vector3d relative = e_position - hs_position;
3      // r/5 can be changed, use this for now
4      if (relative.magnitude() <= r && relative * hs_position.direction() >= -r/5){
5          return true;
6      }
7      return false;
8 }
```

Figure A.4: Logic of is e^- inside the shape of magnet

```

1  vector3d Electron2D::interactB_RK(const MagneticField& B, double time_interval){
2      if (B.isInside(pos) == -1){
3          return vector3d(0,0,0);
4      }
5      Electron2D e_dummy;
6      e_dummy.Et = Et;
7      e_dummy.pos = pos;
8      e_dummy.vel = vel;
9      double time_halved = time_interval*0.5;
10     // get k1
11     vector3d F_m = (e_dummy.vel % B.getField(pos))*eQMratio;
12     vector3d k1 = F_m * e_dummy.gamma_inv();
13     // get k2
14     e_dummy.move(time_halved);
15     e_dummy.accelerate(k1, time_halved);
16     F_m = (e_dummy.vel % B.getField(pos))*eQMratio;
17     vector3d k2 = F_m * e_dummy.gamma_inv();
18     // get k3
19     e_dummy.vel = vel;
20     e_dummy.accelerate(k2, time_halved);
21     vector3d k3 = F_m * e_dummy.gamma_inv();
22     // get k4
23     e_dummy.vel = vel;
24     e_dummy.move(time_halved);
25     e_dummy.accelerate(k3, time_interval);
26     F_m = (e_dummy.vel % B.getField(pos))*eQMratio;
27     vector3d k4 = F_m * e_dummy.gamma_inv();
28
29     return (k1 + k2*2 + k3*2 + k4)/6;
30 }
```

Figure A.5: RK4-1 implemenation of $e^- \cdot \vec{B}$

```
1 void Electron2D::interactRK_ActorE(const RFField& E, const MagneticField& B, double time_interval){
2     vector3d run_kut_E = interactE_RK(E, time_interval);
3     vector3d run_kut_B = interactB_RK(B, time_interval);
4
5     vector3d acc = run_kut_E + run_kut_B;
6
7     move(acc, time_interval/2);
8     accelerate(acc, time_interval);
9     move(acc, time_interval/2);
10 }
```

Figure A.6: RK4-1 implemenation of e^- -EM

```

1 void Electron2D::interactRK(RFField& E, MagneticField& B, const double time, double time_interval){
2     Electron2D e_dummy;
3     e_dummy.Et = Et;
4     e_dummy.pos = pos;
5     e_dummy.vel = vel;
6
7     // Calculate k1
8     vector3d acc_E = E.actOn(e_dummy);
9     vector3d acc_B = B.actOn(e_dummy);
10    vector3d acc = acc_E + acc_B;
11    e_dummy.move(time_interval);
12    e_dummy.accelerate(acc, time_interval);
13    vector3d pos_k1 = e_dummy.pos, vel_k1 = e_dummy.vel;
14
15    // Calculate k2
16    e_dummy.pos = (pos + pos_k1)*0.5;
17    e_dummy.vel = (vel + vel_k1)*0.5;
18    e_dummy.Et = e_dummy.gamma()*E0;
19    E.update(time + time_interval*0.5);
20
21    acc_E = E.actOn(e_dummy);
22    acc_B = B.actOn(e_dummy);
23    acc = acc_E + acc_B;
24    e_dummy.move(time_interval);
25    e_dummy.accelerate(acc, time_interval);
26    vector3d pos_k2 = e_dummy.pos, vel_k2 = e_dummy.vel;
27
28    // Calculate k3
29    e_dummy.pos = (pos + pos_k2)*0.5;
30    e_dummy.vel = (vel + vel_k2)*0.5;
31    e_dummy.Et = e_dummy.gamma()*E0;
32    E.update(time + time_interval*0.5);
33
34    acc_E = E.actOn(e_dummy);
35    acc_B = B.actOn(e_dummy);
36    acc = acc_E + acc_B;
37    e_dummy.move(time_interval);
38    e_dummy.accelerate(acc, time_interval);
39    vector3d pos_k3 = e_dummy.pos, vel_k3 = e_dummy.vel;
40
41    // Calculate k4
42    E.update(time + time_interval);
43
44    acc_E = E.actOn(e_dummy);
45    acc_B = B.actOn(e_dummy);
46    acc = acc_E + acc_B;
47    e_dummy.move(time_interval);
48    e_dummy.accelerate(acc, time_interval);
49    vector3d pos_k4 = e_dummy.pos, vel_k4 = e_dummy.vel;
50
51    E.update(time);
52    pos = (pos_k1 + pos_k2*2 + pos_k3*2 + pos_k4)/6;
53    vel = (vel_k1 + vel_k2*2 + vel_k3*2 + vel_k4)/6;
54    Et = gamma()*E0;
55 }
```

Figure A.7: RK4-2 implementation of e^- -EM

APPENDIX B: Example Simulation Runs

```

1 Optimal phase with the least RMS : -5
2
3 Simulation settings :
4 ph = -5 deg, gt = 1 ns, enum = 1000
5 dT = 0.001 ns, dT_out = 0.01 ns
6
7 For the 1th magnet:
8 Optimum out path = 0.81044 m
9 Magnet guide = 0.25852 m
10 Rho = 0.088477 m
11 Drift time of the first electron in the bunch : 7.688 ns
12 Drift time of the last electron in the bunch : 7.487 ns
13 Max energy = 0.47581 MeV
14 RMS = 0.0058165 MeV
15
16 For the 2th magnet:
17 Optimum out path = 1.0833 m
18 Magnet guide = 0.37766 m
19 Rho = 0.098898 m
20 Drift time of the first electron in the bunch : 5.597 ns
21 Drift time of the last electron in the bunch : 5.617 ns
22 Max energy = 0.89172 MeV
23 RMS = 0.0099018 MeV
24
25 For the 3th magnet:
26 Optimum out path = 1.1705 m
27 Magnet guide = 0.41573 m
28 Rho = 0.10223 m
29 Drift time of the first electron in the bunch : 5.314 ns
30 Drift time of the last electron in the bunch : 5.325 ns
31 Max energy = 1.298 MeV
32 RMS = 0.013879 MeV
33
34 Electron with the most energy : 623) 1.6999 MeV,           RMS of bunch : 0.017981 MeV
35
36 Total steps calculated : 12468052652
37 Simulation finished in : 632050015 us      ( 632.1 s )
38

```

Figure B.1: ϕ_{lag} , ρ & L optimization at
 $P = 12\text{KW}$, $R_1 = 0.188\text{m}$, $R_2 = 0.753\text{m}$, $t_g = 1\text{ns}$, $E_{in} = 40\text{KeV}$

```

1 Optimal phase with the least RMS : 0
2
3 Simulation settings :
4 ph = 0 deg, gt = 0.8 ns, enum = 1000
5 dT = 0.001 ns, dT_out = 0.01 ns
6
7 For the 1th magnet:
8 Optimum out path = 0.80787 m
9 Magnet guide = 0.2574 m
10 Rho = 0.088379 m
11 Drift time of the first electron in the bunch : 7.629 ns
12 Drift time of the last electron in the bunch : 7.48 ns
13 Max energy = 0.47579 MeV
14 RMS = 0.0038689 MeV
15
16 For the 2th magnet:
17 Optimum out path = 1.0833 m
18 Magnet guide = 0.37765 m
19 Rho = 0.098898 m
20 Drift time of the first electron in the bunch : 5.589 ns
21 Drift time of the last electron in the bunch : 5.605 ns
22 Max energy = 0.89169 MeV
23 RMS = 0.0068848 MeV
24
25 For the 3th magnet:
26 Optimum out path = 1.1705 m
27 Magnet guide = 0.41573 m
28 Rho = 0.10223 m
29 Drift time of the first electron in the bunch : 5.311 ns
30 Drift time of the last electron in the bunch : 5.318 ns
31 Max energy = 1.298 MeV
32 RMS = 0.0096887 MeV
33
34 Electron with the most energy : 629) 1.6999 MeV,           RMS of bunch : 0.012318 MeV
35
36 Total steps calculated : 12455378454
37 Simulation finished in : 631136046 us      ( 631.1 s )

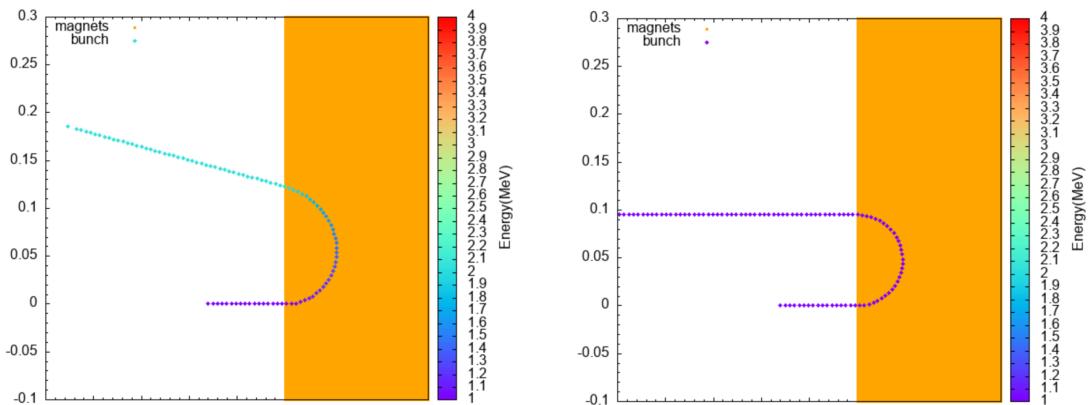
```

Figure B.2: ϕ_{lag} & ρ & L optimization at $P = 12\text{KW}, R_1 = 0.188\text{m}, R_2 = 0.753\text{m}, t_g = 0.8\text{ns}, E_{in} = 40\text{KeV}$

$dt(\text{ns})$	$T_{\text{simLF}}(s)$	$T_{\text{simRK1}}(s)$	$T_{\text{simRK2}}(s)$	$\Delta T_{\text{LF-RK2}}$
10^{-2}	4.14092	4.146	4.10945	0.77%
10^{-3}	4.23817	4.61703	4.30225	-1.51%
10^{-4}	5.04679	8.93529	6.3065	-24.96%
10^{-5}	13.5135	51.9024	25.4279	-88.17%
10^{-6}	97.6229	486.378	217.704	-123.01%

Table B.1: dt vs T_{sim} of Leap-frog, RK4-1, RK4-2 $E_{in} = 1\text{MeV}, \mathbf{B}=0.1\text{T}, t_{end} = 5\text{ns}$

$dt(\text{ns})$	$\Delta E_{\text{LF}}(\text{MeV})$	$\Delta E_{\text{RK1}}(\text{MeV})$	$\Delta E_{\text{RK2}}(\text{MeV})$	$\Delta E_{\text{LF-RK2}}$
10^{-2}	2.78323	1.26503	1.04713	165.8%
10^{-3}	0.11712	0.07469	0.06691	42.87%
10^{-4}	0.01101	0.00716	0.00644	41.5%
10^{-5}	0.00109	0.00071	0.00064	41%
10^{-6}	0.00011	0.00007	0.00006	45%

Table B.2: dt vs ΔE of Leap-frog, RK4-1, RK4-2 $E_{in} = 1\text{MeV}, \mathbf{B}=0.1\text{T}, t_{end} = 5\text{ns}$  $dt = 10^{-2}\text{ns}, \Delta E = 1.047\text{MeV}$ $dt = 10^{-4}\text{ns}, \Delta E = 0.006\text{MeV}$ Figure B.3: Energy gain of 1MeV bunch in $\mathbf{B}=0.1\text{T}$ using RK4-2

REFERENCES

1. Jacques Pottier, *A new type of rf electron accelerator: The rhodotron*, Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, Volumes 40–41, Part 2, 1989, Pages 943-945,
2. J.M. Bassaler, J.M. Capdevila, O. Gal, F. Lainé, A. Nguyen, J.P. Nicolaï, K. Umiastowski, *Rhodotron: an accelerator for industrial irradiation*, Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, Volume 68, Issues 1–4, 1992, Pages 92-95,A:
3. Y. Jongen. (2001). *Manufacturing of Electron Accelerators*. Ion Beam Applications s.a. (IBA) Chemin du Cyclotron 3, B-1348 Louvain-la-Neuve, Belgium.
4. Fukushima, Toshio. (2001). *Reduction of Round-off Errors in Symplectic Integrators*. The Astronomical Journal. 121. 1768-1775.
5. W. Kleeven, M. Abs, J. Brison, E. Forton, J. Hubert and J. Van de Walle, *Design and Simulation Tools for the High-Intensity Industrial Rhodotron Electron Accelerator*, 9th International Particle Accelerator Conference, 2018