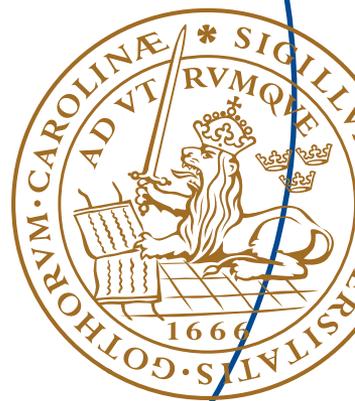


Master's Thesis

**Field parametrization by
trigonometric expansion
combined with particle tracking
in superconducting cavities**

Tobias Lindqvist



Field parametrization by trigonometric expansion
combined with particle tracking in
superconducting cavities

Tobias Lindqvist
tobiaslindqvist@live.se

Department of Electrical and Information Technology
Lund University

Advisors: Emanuele Laface & Anders Karlsson

November 15, 2013

Abstract

The European Spallation Source research facility will be built in Lund. The driver of the spallation process will be a linear proton accelerator that will deliver 2.5 GeV protons to a tungsten target. At a beam power of 5 MW, ESS has set the ambitious goal of maintaining a beam loss of 0.1 W/m. To achieve such a goal the properties of the accelerator need to be explored in great detail. This master thesis examines and implements a model for tracking particles inside the superconducting elliptical cavities of a linear accelerator. The model uses a parametrization of the field to vastly reduce the amount of data required to describe the field. The model can be used to compare and examine different designs of the accelerator, and predict accelerator behavior once the accelerator is running.

Acknowledgments

I wish to thank my advisor at ESS, Emanuele Laface, for his support and feedback with everything from writing code to writing this thesis. I want to thank my LTH advisor, Anders Karlsson, for very thorough and excellent feedback regarding the thesis. I wish to thank Mohammad Eshraqi for explaining concepts and discussing ideas with me. I want to thank Garry Trahern for being a great boss and giving time to make sure I enjoyed my stay at ESS. I also want to thank all the other people at ESS that I have had the good fortune of meeting during my time there. I want to thank Gustav Nilsson for giving me feedback when writing the thesis.

Table of Contents

1	Introduction	1
1.1	The European Spallation Source	1
2	Problem Formulation and Motivation	5
2.1	Definitions, Names and Notations	5
3	The Field Construction	9
3.1	Theory for the Fields	9
3.2	Parameters of the field	12
3.3	Spoke cavity parametrization	18
4	Particle Dynamics	21
4.1	Transit Time Factor	21
4.2	Particles in the cavity	21
5	Descriptions and structure of the code	25
5.1	Field parametrization	25
5.2	Particle tracking methods	26
5.3	Input for simulation	30
6	Simulations	31
6.1	Code performance and validity	31
6.2	Particle Simulations	36
7	Further development and discussion	39
7.1	The use of the model	39
7.2	Improving the parametrization	40
7.3	Improving the particle integration	40
7.4	Extending the code	40
	References	43
A	Field parametrization	45

This thesis is subdivided into seven chapters. The first is an introduction to the ESS, it briefly describes what the ESS will be and what it will be able to achieve. The second chapter contains a brief problem formulation. The third chapter describes how the electric field is parametrized and constructed in the cavities. Chapter four contains a description of the particle dynamics that is used to simulate beam dynamics, as well as the methods used to solve the equations for moving particles. The fifth chapter contains a more in depth description of parts of the code used to solve the problem. Chapter six gives a few examples on outputs from simulations as well as comparisons to other methods that performs particle tracking. The last chapter, chapter seven, is a discussion on how the model can be used in the future, together with ideas on how to improve it further.

1.1 The European Spallation Source

ESS will be built outside Lund, Sweden. The aim of the facility is to produce neutrons for research purposes, and when the goals are met it will be the most powerful neutron source in the world. The facility will allow researchers from academia as well as industry to set up experiments relating to a vast variety of different fields and examine the molecular building blocks in various materials. ESS will, when fully operational, be the brightest neutron source in the world [1]. There are other major active spallation source in the world at the moment, among them ISIS in Oxfordshire United Kingdoms and the SNS in Oak Ridge USA.

1.1.1 The source for neutrons

The ESS will, as its name suggests, produce neutrons by the method of spallation. In order to achieve spallation high energy protons are fired on to a heavy metal target. In the target the protons will interact with the nuclei in the target which will in turn decay and produce between 10-20 neutrons depending on the material and proton energy[3]. An alternative method to spallation is to use a nuclear reactor to drive a nuclear chain reaction that will yield neutrons. The neutron is a hadronic particle with zero charge and a mass of $939.5 \text{ MeV}/c^2$, and it is unstable in its free form and decays with a mean life time of 15 minutes[8]. The neutron is a basic constituent in almost all atomic nuclei where it exists in a

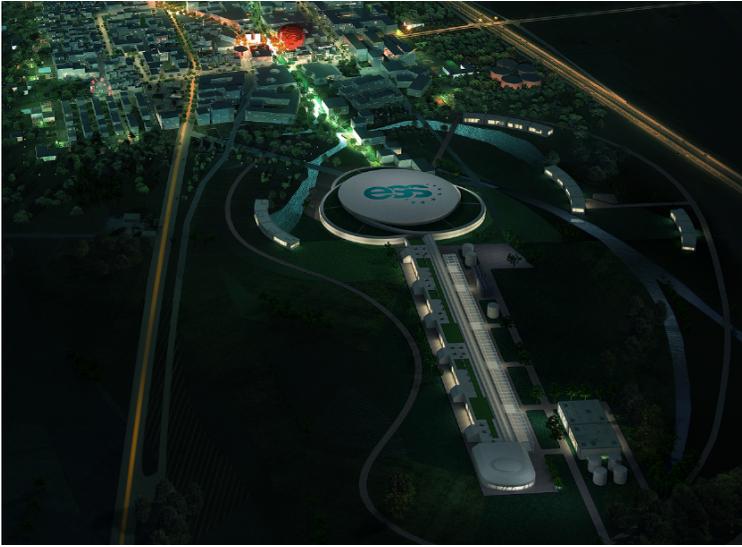


Figure 1.1: The picture show an artist rendition of the completed ESS research facility.

stable condition. The neutron is composed of three charged quarks which gives it a magnetic moment. A neutron can interact either by weak interaction, strong interaction, or by its magnetic moment. The short life time of free neutrons make them rare outside the comfort of the atomic nuclei. However, the fact that the neutron is a common naturally occurring massive particle without charge puts it in a special place, and makes it an effective tool to be used for scientific purposes. The ESS will utilize cold neutrons, which means that the neutrons travels at roughly 2200 m/s, this will give them a wavelength of 0.18 nm. The wavelength of cold neutrons is ideal for the various experiments the ESS will be able to set up.

1.1.2 Where are neutrons needed

The research fields that will be able to benefit from the ESS facility are many in number, and their variety in applications wide. Some examples are [1]

- **Life Science** is interested in the behaviors of biomolecules which can be examined in structure at an atomic scale. Neutrons will also be able to resolve the internal dynamics of biomolecules.
- **Archeology** has found use of neutrons in their ability to probe deep within a material without destroying it. Neutrons can examine the internal structure and composition of cultural artifacts that can give us insight to our own heritage.
- **Fundamental and particle physics** applications within this field can be found for example in the search for the neutron electric dipole moment.

A search could result in clues as to why there is an observed asymmetry between matter and anti-matter in the universe.

This is just a few examples of the vast range of fields that is able to harvest the knowledge that ESS provides.

1.1.3 The ESS LINAC and Target

LINAC stands for linear accelerator which means that it accelerates the particles along a straight tube. The ESS particle accelerator will generate an average beam of 5 MW power, with a peak power of 125 MW [4]. The LINAC consists of several parts that accelerates the particles. There are several accelerating components, a radio frequency quadrupole (RFQ), spoke cavities (SPK), and elliptical cavities (medium and high- β), giving it a total length of roughly 600 m. The LINAC layout can be seen below in Figure 1.2. The ESS LINAC will accelerate protons to 2.5 GeV before the protons hit the target and initiates the spallation process. The accelerator will deliver particle bunches at a frequency of 14 Hz, with a current of 50 mA.

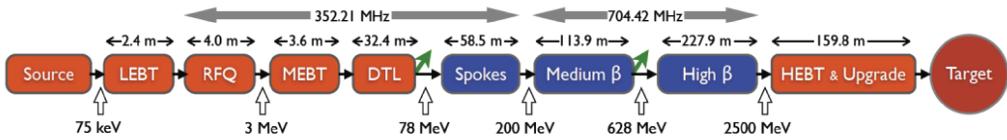


Figure 1.2: The figure describes the layout of the linac and gives the energies between sections. The red sections are in room temperature. The blue sections are at a temperature of only a few kelvin, they hold superconducting components. LEBT is the low energy beam transport, MEBT is the Medium EBT and finally the High EBT followed by room for potential upgrades [1].

The elliptic cavities make up the part of the accelerator that gives the particles the major part of their energy. In Figure 1.2 they are referred to as medium and high β . The β -factor is the particle speed relative to the speed of light. Medium β means a speed around 70% of the speed of light and high β is around 90% of the speed of light. By the time the particles reach the heavy metal target their kinetic energy is larger than their rest mass and their speed will be roughly 95% of the speed of light. The elliptic cavities are superconducting Niobium cavities. The field in the cavities will have a peak gradient of about 30 MV/m and the field will oscillate at a frequency of 704.42 Hz. The ESS LINAC is likely to be the first facility incorporating spoke cavities into its accelerator [1]. The spoke cavities are also superconducting and are mechanically much stiffer than the elliptic cavities making them less sensitive to vibrations or deformations [1], the field in the spoke cavities oscillates at 352.21 Hz.

In order to produce the neutrons the LINAC accelerates protons into a large tungsten cylinder. It is in this heavy metal cylinder that the spallation process occurs and it the tungsten disc makes up the core of the target. The tungsten cylinder has a diameter of 2.5 m. A system of liquid helium will cool the tungsten cylinder, the structure is surrounded by a moderator system, the moderators are first liquid hydrogen followed by water. The moderation process slows down the neutrons to desired speed of roughly 2.2 km/s [1]. The entire target structure is shielded to protect the outside from the radiation produced in the spallation process.

Problem Formulation and Motivation

The goal is that the ESS LINAC operates at an average power of 5 MW with a beam loss of less than 0.1 W/m [1]. It corresponds to a loss of one particle in a 100000. This means that precise control and awareness of the accelerator design is essential.

The ESS will be going to use two systems to operate and develop the LINACs behavior, both in the planning stage and when running it. The first is the online model that in real time will be able to analyze and predict the behavior of the accelerator. The second is the offline model, that is used to make more in depth analysis and simulations of the accelerator. The drawback of the offline model is that it requires more computational power and time. Part of the reason of this thesis is also to see where the model presented here will fit in.

The main goals of the project in this thesis is to produce an accurate and fast code, written in C, that can generate a parametrization of the electrical field in a given elliptic cavity. There was also the goal to be able to simulate the particle dynamics in sections of the LINAC corresponding to the medium and high β sections.

The parametrization is done for two main reasons. The amount of data required to store a field can be vastly reduced from thousands of points to less than 20. This is not a good method for arbitrary electric fields, but it works well for the fields in the elliptic cavities since the general shape of the field is well known. The other reason for making the parametrization is that it gives a good physical connection at a low cost. The method used to describe the field is computationally cheap relative to a full FEM analysis solving the field in the cavity. The program will be able to make accurate predictions of particle paths, and compare it to other models.

2.1 Definitions, Names and Notations

In this section the definitions, notations and naming conventions for relevant variables and quantities are presented.

2.1.1 Coordinates

When treating particles in an accelerator it is usually so that the coordinates are defined relative a reference particle called the synchronous particle. The synchronous particle can be viewed as the ideal particle in the accelerator, it is used so the right accelerator behavior is achieved when the construction has been completed. That means that path of the synchronous particle is a design property of the accelerator and it will roughly correspond to the path of a particle at the center of a bunch. From the synchronous particle one can define the position of an arbitrary particle as its position relative to the synchronous particle. The coordinates are

$$(x(s), y(s), z(s), x'(s), y'(s), z'(s)) \quad (2.1)$$

The origin is defined on the synchronous particle. The variable s is the position of a particle down the accelerator. The z -direction is defined, for straight accelerators, in the direction of motion, i.e. along the accelerator. The z coordinate is parallel to s and describes the longitudinal position of a particle relative to the synchronous particle. The coordinates x and y span the plane perpendicular to the direction of motion. The coordinates x and y are called the transverse positions with y being defined in the direction of gravity. The coordinate

$$x' \equiv \frac{\partial x}{\partial s} \quad (2.2)$$

is the change in x position over an infinitesimal distance in s . The same is true for y' but slightly different for z' . The coordinate z' is defined using the relativistic momentum p . The energy E and momentum p , of a particle, is

$$E = \gamma E_0 = \gamma m_0 c^2, \quad \gamma = \frac{1}{\sqrt{1 - \beta^2}}, \quad p = \gamma m_0 \beta c \quad [5] \quad (2.3)$$

Where

$$m_0 = 938.27 \text{ MeV}/c^2 \quad (2.4)$$

is the proton rest mass,

$$c = 299,792,458 \text{ m/s} \quad (2.5)$$

is the speed of light in vacuum and

$$\beta \equiv \frac{v}{c} \quad (2.6)$$

is the β -value of a particle, it is defined as the particles relative speed to the speed of light, where v is the particles speed. By this z' is defined as

$$z' \equiv \frac{p - p_s}{p_s} \quad (2.7)$$

where p_s is the momentum of the synchronous particle. The coordinate z' is the deviation in momentum of a particle relative to the synchronous particle. A relevant quantity for the accelerating cavities is the geometrical beta, β_g , which is a result of the design of the cavities. It is defined as

$$\beta_g \equiv \frac{l f}{2c} \quad (2.8)$$

where l is the distance between the nodes in the oscillating electric field, f is the frequency of the oscillation. Another variable related to the particles that will be relevant is ϕ , it is the phase of the field in the cavity at the position of the particle.

2.1.2 Notations

Here a notation is adapted where coordinates and properties of the synchronous particle is referred to by the subscript s , so for example, β_s is speed of the synchronous particle relative to the speed of light. Now the coordinate system used has been defined. This leads to that the coordinates $(x_s, y_s, z_s) \equiv (0, 0, 0)$ since the origin of the coordinate system is defined on the synchronous particle. Vector quantities will be denoted with an arrow, e.g, the electric field is denoted \vec{E} . At times only parts of a vector quantity are relevant, in such cases the component will be denoted by a subscript referring to the direction, so E_z would be the electric field in the z -direction. Due to the radial symmetry of the problem, some times the subscript r will be used interchangeably with x or y , where $r = \sqrt{x^2 + y^2}$.

The Field Construction

This chapter is a description on how the field is parametrized for the model. It explains the field parametrization in both the elliptic cavities and the spoke cavities. It also shows examples of the result when performing the field parametrization.

3.1 Theory for the Fields

A good way to construct the field is to start at the basics, namely the Maxwell equations. In a vacuum, which is assumed for the interior of the cavities, the Maxwell equations can be written as [6].

$$\nabla \cdot \vec{E} = 0 \quad \nabla \cdot \vec{B} = 0 \quad \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad \nabla \times \vec{B} = \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \quad (3.1)$$

By eliminating the magnetic flux density \vec{B} , the PDE for the electric field is obtained (3.2).

$$\nabla^2 \vec{E} - \mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2} = 0 \Leftrightarrow \nabla^2 \vec{E} - \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2} = 0 \quad [6] \quad (3.2)$$

The field in the cavity is assumed to be symmetric with respect to the angular coordinate θ , this implies that $\frac{\partial \vec{E}}{\partial \theta} = 0$. The operator $\nabla^2 = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{\partial^2}{\partial z^2}$ when expressed in cylindrical coordinates. The derivation of the field is performed in cylindrical coordinates. Equation (3.2) must be satisfied in order for the model to correspond to a physical reality.

The starting point for the parametrization is field data of the electric field component E_z , in the center of the cavity, in the z -direction of the cavity. The given field is at a phase such that it is at a maximum amplitude. The field in the z -direction in the cavity can be seen below in Figure 3.1, the field corresponds to the field in a high- β cavity. The field is modulated by a time factor, $\sin(\omega t + \phi)$. Let the function $f(z)$ describe the spatial component of the electric field on the z -axis. It can be expressed as sum of cosine functions

$$f(z) = \sum_{m=0}^{\infty} A_m \cos\left(\frac{\pi m z}{L}\right) \quad [10] \quad (3.3)$$

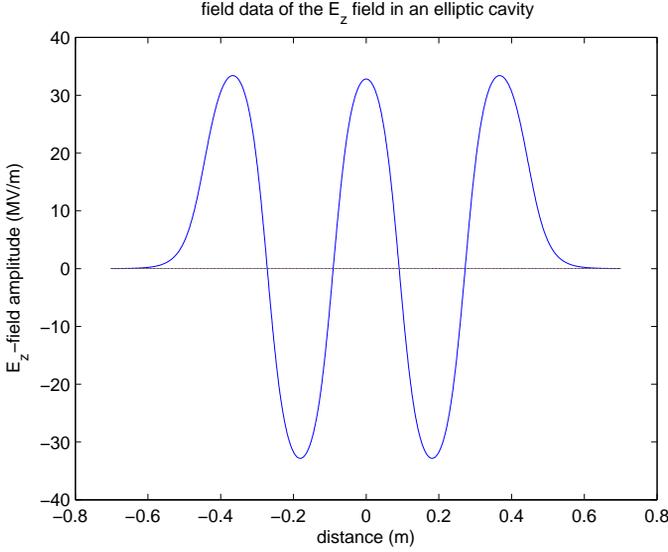


Figure 3.1: Plotted field data of the electric field on the z -axis, as a function of distance in a high- β cavity. The field data is taken from measurements of the manufacturer at Soleil in France.

The assumption for the field is that it is excited in the TM_{01} -mode which gives axial symmetry. The TM_{01} -mode is an eigenfunction to the wave equation in cylindrical coordinates. A goal in designing the cavities is for the field to mainly be composed of the TM_{01} -mode since it focuses the field amplitude in the center of the cavity [1]. The general form of the TM -modes are

$$TM_{lm} = J_l(kr) \cos\left(\frac{\pi mz}{L}\right) \cos(l\theta) \quad [10] \quad (3.4)$$

where θ is the angular coordinate J_l is the ordinary Bessel function of order l and r is the radial coordinate which is scaled by a factor k . The m is the order of the cosine function which is scaled by a factor $\frac{1}{L}$ where L is the distance between the nodes of the field. With the assumption of TM_{01} the $l = 0$, thus the term containing θ can be neglected throughout this thesis. The entire electric field can be written as $\vec{E} = E_r(r, z, t)\vec{e}_r + E_z(r, z, t)\vec{e}_z$ for the whole cavity. The field in the z -direction becomes

$$E_z(r, z, t) = \sum_{m=0}^{\infty} A_m J_0(k_m r) \cos\left(\frac{\pi mz}{L}\right) \sin(\omega t) \quad (3.5)$$

where ω is the angular frequency of the field, c is the speed of light and J_0 is the ordinary Bessel function of order zero. Inserting the expression in (3.5) into the wave Equation (3.2) the following condition arises

$$k_m^2 = \left(\frac{m\pi}{L}\right)^2 - \left(\frac{\omega}{c}\right)^2 \quad (3.6)$$

This gives a condition on the radial scaling of the field that depends on the longitudinal frequency as well as the frequency in time.

The radial component of the field can be found using Gauss law. This is made easy since the TM_{01} -mode has no component of the electric field in the θ -direction. Together with the assumption that the vacuum is perfect it gives, in cylindrical coordinates

$$\nabla \cdot \vec{E} = 0 \Leftrightarrow \frac{\partial E_z}{\partial z} = -\frac{1}{r} \frac{\partial(rE_r)}{\partial r} \quad (3.7)$$

This can be solved by some analysis

$$rE_r = -\int_0^r r' \frac{\partial E_z}{\partial z} dr' \quad (3.8)$$

The only r dependance in the E_z function lies in the Bessel function J_0 , so one only has to integrate the Bessel function. The series expansion of the Bessel function is

$$J_a(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!(m+a)!} \left(\frac{x}{2}\right)^{2m+a} \quad (3.9)$$

The integral that is needed to evaluate is

$$\int x J_0(x) dx = \int \sum_{m=0}^{\infty} \frac{(-1)^m}{m!m!} \left(\frac{x}{2}\right)^{2m} x dx = \quad (3.10)$$

$$\sum_{m=0}^{\infty} \frac{(-1)^m}{m!m!(2m+2)} \frac{x^{2m+2}}{2^{2m}} = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!(m+1)!} x \left(\frac{x}{2}\right)^{2m+1} = x J_1(x) \quad (3.11)$$

An expression for the electrical field in the radial direction is now obtained as

$$E_r(z, r, t) = -\sum_{m=0}^{\infty} \frac{A_m}{k_m} J_1(k_m r) \frac{\partial \cos(m\pi z/L)}{\partial z} \sin(\omega t) = \quad (3.12)$$

$$= \sum_{m=0}^{\infty} \frac{A_m m \pi}{L k_m} J_1(k_m r) \sin(m\pi z/L) \sin(\omega t) \quad (3.13)$$

The corresponding magnetic field is obtained from the Ampere's circuital law $\nabla \times \vec{B} = \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \Rightarrow$

$$\frac{1}{r} \left(\frac{\partial(rB_\theta)}{\partial r} - \frac{\partial B_r}{\partial \theta} \right) = \mu_0 \epsilon_0 \frac{\partial E_z}{\partial t} \quad (3.14)$$

The field is symmetrical in the θ -direction so $\frac{\partial B_r}{\partial \theta} = 0$, the other term can be found by integrating the Bessel function in the same way as was done for E_r . One also need to take the time derivative of E_z and use $\mu_0 \epsilon_0 = \frac{1}{c^2}$ to arrive at

$$B_\theta(z, r, t) = \sum_m^{\infty} \frac{A_m \omega}{k_m c^2} J_1(k_m r) \cos\left(\frac{m\pi z}{L}\right) \cos(\omega t) \quad (3.15)$$

The z and r -component of the \vec{B} -field is zero since only TM-modes are considered.

3.2 Parameters of the field

Figure 3.1 shows the typical variation of E_z along the z -axis. It is this component of the field that accelerates the particles. It is clear from Figure 3.1 that the field is sinusoidal in shape. In order to make a close approximation of the field it is subdivided into five sections, as in Figure 3.2. The central section, denoted

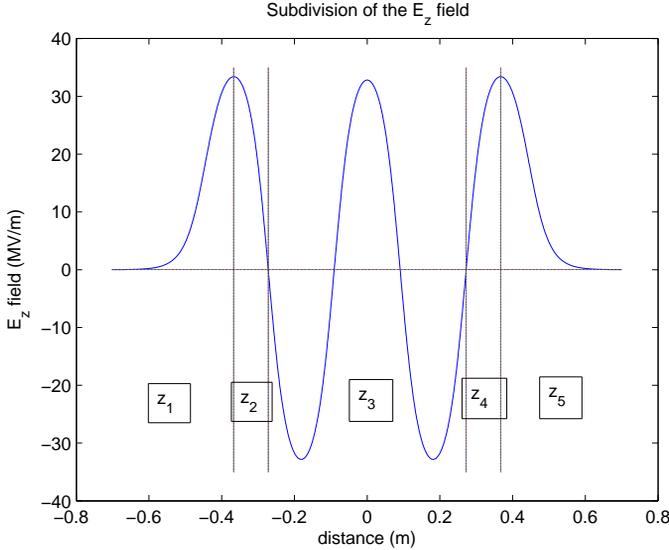


Figure 3.2: The figure shows the field amplitude in MV/m, of the electric field in the z -direction at a phase corresponding to a maximal amplitude. The field is that of a high- β cavity. The figure shows how the field is subdivided into five sections, from z_1 to z_5 . In each of these sections a different set of parameters will be used to describe the shape of the field.

z_3 in Figure 3.2, is defined between the first and last zeroes of the field. The sections denoted as z_2 and z_4 are defined between section z_3 and the first and last anti-nodes of the field.

For the sections z_2 , z_3 and z_4 a first and a third order cosine terms are used for the expansion of the field. This expression of the field is done in accordance to Equation (3.5) where $m = 1, 3$. This parametrization is written as

$$E_z(r, z, t) = \left(A J_0(k_1 r) \cos\left(\frac{\pi z}{L}\right) + B J_0(k_3 r) \cos\left(\frac{3\pi z}{L}\right) \right) \sin(\omega t) \quad (3.16)$$

Figure 3.2 display the field on the z -axis, at a time when $\sin(\omega t) = 1$. The field is denoted $E_z(z)$ and since $J_0(0) = 1$ it is given by

$$E_z(z) = A \cos\left(\frac{\pi z}{L}\right) + B \cos\left(\frac{3\pi z}{L}\right) \quad (3.17)$$

where the parameters A , B and L are unknown at this stage. In the central section, z_3 , there is another unknown, it is the number of periods of the field. Based on examinations of the presence higher order modes, only two cosine terms are deemed sufficient to describe the field in the three central sections. For these sections the L 's are found by looking at the distances between the zeros of the field and taking the average, this ensures that the field is of the correct length. A two term expansion in the three central parts of Figure 3.2 is sufficient to approximate the field to a high degree. Three different fits are made for each section z_2 , z_3 and z_4 because they do not necessarily have the same L , A or B . This difference arises from the opening at the end of the cavity which results in a perturbation of the field, which causes the first and last anti-nodes to be "pushed out" towards the opening. The cosine terms, in section z_3 , are fitted using the least square method which minimizes

$$\sum_i \left(f(z_i) - A \cos\left(\frac{\pi z_i}{L}\right) - B \cos\left(\frac{3\pi z_i}{L}\right) \right)^2 \quad (3.18)$$

for A and B . For section z_2 and z_4 the finding of the parameters were done in two different ways. One was by fitting the amplitude at the boundary between z_1 and z_2 , and fitting the amplitude at the boundary of z_4 and z_5 respectively. This fit is combined with a fit of the derivatives at the boundaries to section z_3 . The other way is to fit the function in sections z_2 and z_4 using the least square method, like in section z_3 . In the sections denoted as z_1 and z_5 in Figure 3.2 the field is assigned the following form

$$C \exp\left(-\left|\frac{z-\mu}{\sigma}\right|^p\right) \quad (3.19)$$

where μ is chosen as the z -coordinate where the field peaks on the boundary to section z_2 or z_4 . The constant C is found as the sum of parameters, $A + B$ from section z_2 or z_4 . Neither μ or C need to be given as parameters. The two parameters are p and σ . The values for the parameters p and σ are found analytically by choosing two field values and solving

$$f(z_n) = C \exp\left(-\left|\frac{z_n-\mu}{\sigma}\right|^p\right) \quad \text{and} \quad f(z_m) = C \exp\left(-\left|\frac{z_m-\mu}{\sigma}\right|^p\right) \quad (3.20)$$

The points z_n and z_m are taken around half the amplitude of C and at 20% of the amplitude of C , $f(z_n)$ and $f(z_m)$ is field data values at those points. The field in the three central sections has been simplified to

$$E_z(z, r, t) = \left(A J_0(k_1 r) \cos\left(\frac{\pi z}{L}\right) + B J_0(k_3 r) \cos\left(\frac{3\pi z}{L}\right) \right) \sin(\omega t) \quad (3.21)$$

$$E_r(z, r, t) = -\left(\frac{A\pi}{Lk_1} J_1(k_1 r) \sin\left(\frac{m\pi z}{L}\right) + \frac{B3\pi}{Lk_3} J_1(k_3 r) \sin\left(\frac{3\pi z}{L}\right) \right) \sin(\omega t) \quad (3.22)$$

and

$$B_\theta(z, r, t) = \left(\frac{A\omega}{c^2 k_1} J_1(k_1 r) \cos\left(\frac{\pi z}{L}\right) + \frac{B\omega}{c^2 k_3} J_1(k_3 r) \cos\left(\frac{3\pi z}{L}\right) \right) \cos(\omega t) \quad (3.23)$$

In section z_1 and z_5 the field becomes

$$E_z(z, r, t) = (AJ_0(k_1r) + BJ_0(k_3r)) \sin(\omega t) \exp\left(-\left|\frac{z-\mu}{\sigma}\right|^p\right) \quad (3.24)$$

$$E_r(z, r, t) = \left(\frac{A}{k_1}J_1(k_1r) + \frac{B}{k_3}J_1(k_3r)\right) \sin(\omega t) \frac{\partial}{\partial z} \exp\left(-\left|\frac{z-\mu}{\sigma}\right|^p\right) \quad (3.25)$$

and

$$B_\theta(z, r, t) = \frac{\omega}{c^2} \left(\frac{A}{k_1}J_1(k_1r) + \frac{B}{k_3}J_1(k_3r)\right) \cos(\omega t) \exp\left(-\left|\frac{z-\mu}{\sigma}\right|^p\right) \quad (3.26)$$

The parametrization defined in Equation (3.19) is not physical in the sense that it fulfills the wave equation. Figure 3.3 shows the field $E_z(z)$ together with the parametrization. Figure 3.3 also includes the relative error between the field data and the parametrization.

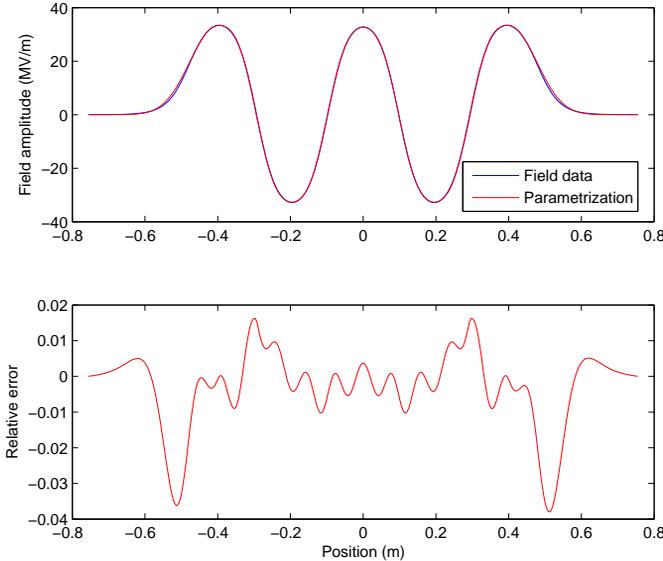


Figure 3.3: The parametrization is shown in red while the field that it is derived from is shown in blue. The second plot shows the difference between the parametrization and the actual field data, normalized over the maximum amplitude of the field. This indicates to what degree the expansion in only two cosine terms can approximate the field. The largest error is present in the sections z_1 and z_5 .

It is not clear how valid the parametrized model is in the radial direction since no full field map is available to compare against. However a field plot may

indicate the correctness. If one freezes the field in time and looks at the parts that contribute energy to particles, namely the electric field in the E_r and E_z direction a plot of the field can be made. A field plot can also be shown as a potential. However, it is worth mentioning that it is not truly the potential of the field since the actual field varies in time and is thus no longer conservative. The field is contained within a superconductor, and for a superconductor the resistance of the walls is zero. This implies that the electric field lines are perpendicular to the walls of the cavity and the potential of the field should indicate the walls of the cavity. The potential is calculated as

$$V(z, r) = \int_{(z_0, r_0)}^{(z, r)} \vec{E} \cdot d\vec{l} \quad [6] \quad (3.27)$$

The plot of the field lines as well as the potential can be seen in Figure 3.4, the potential was numerically calculated using MATLAB. Figure 3.5 shows the shape

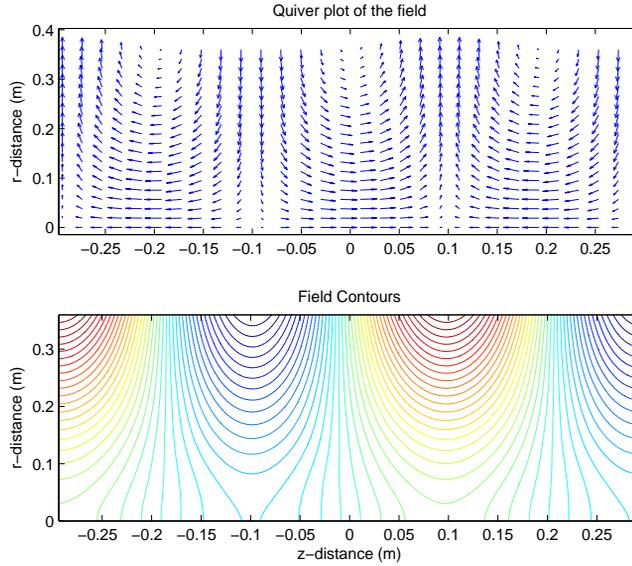


Figure 3.4: A quiver plot of the E_z -field and a potential plot, both derived from the parametrization. This is only displayed for the middle cosine section, z_3 in Figure 3.2. For a superconducting cavity the arrows in the quiver-plot should be perpendicular to the cavity walls. This also means that for a field in a superconductor the outline of the cavity walls seen as a potential line. The plot gives an indication of where the walls are.

of the E_z -field and the E_r -field, where the E_r -field is plotted at a 1 cm distance from the central axis. The shape of $E_z(z)$ and B_θ -field is very similar, they differ only slightly in the radial direction.

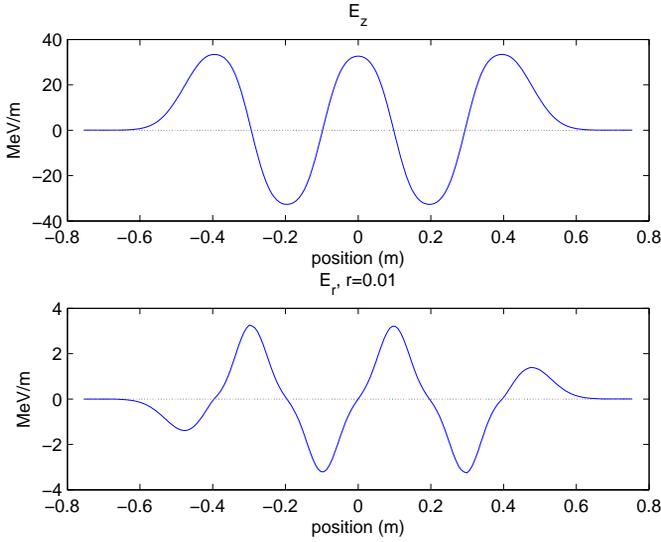


Figure 3.5: A plot showing the E_z and E_r -field. The E_r -field is shown at a radius $r = 0.01$ since E_r is 0 at $r = 0$.

The parametrization describes the field in the cavity with 16 parameters. Three parameters in each of the sections z_2 , z_3 and z_4 , two parameters in each of the sections z_1 and z_5 , and finally three more parameters giving the start and endpoint of the field as well as the number of nodes the field has. An example of a set parameters is given in Table 3.1, they correspond to the field shown in Figure 3.3.

3.2.1 Making the parameters physically intuitive

When dealing with the parameters it is convenient to give them a physical interpretation. The parameters can then be reformulated in order to simplify the interpretation. The parameters A and B add up to a total amplitude

$$\text{Amp} = A + B \quad (3.28)$$

and the ratio A/B defines the coupling as

$$\text{Coupling} = A/B \quad (3.29)$$

Thus the parameters A and B are defined by one amplitude and a coupling factor. The parameter L that describes the spatial frequency of the field in the three central sections varies in each section. In the sections denoted z_2 and z_4 the L is typically larger than in section z_3 , this difference can be expressed as

$$L_{z_2} = L_{z_3} + dL_{z_2} \quad (3.30)$$

Parameters			
Sections z_2 - z_4	A (MV/m)	B (MV/m)	L (m)
z_2	35.793	-2.3916	0.20476
z_3	34.591	-1.9033	0.19591
z_4	35.795	-2.3928	0.20477
Sections z_1, z_5	P	σ	
z_1	2.1501	0.10870	
z_5	2.1612	0.10900	
General parameters			
	starting position (m)	final position (m)	nr. of zeros
	-0.7575	0.7575	4

Table 3.1: The table gives the parameters that generate the field in Figure 3.3.

This gives the displacement of the anti-nodes in section z_2 and z_4 as dL_{z_2} and dL_{z_4} . The parameters in the first and last section, z_1 and z_5 in Figure 3.2 are not as easy to re-interpret.

3.2.2 Comments on the parametrization

In Table 3.1 the parameters reveal that the field is almost symmetric with respect to $z = 0$. The model can be simplified further by assuming a symmetric field and by that reducing the number of parameters from 16 to 10.

One of the initial motivation was to construct a field that satisfies the wave equation. This is however not the case in section z_1 and z_5 , seen in Figure 3.2. The expression

$$e^{|\frac{z-\mu}{\sigma}|^p} \quad (3.31)$$

does not satisfy Equation (3.2). A possible way to satisfy this equation in the absence of a cosine expression for the z-direction is instead to say that the field is proportional to an evanescent wave

$$e^{a(z-\mu)} \quad (3.32)$$

with $a > 0$ in section z_1 and $a < 0$ in section z_5 , and μ marks the boundary between z_1, z_2 and z_4, z_5 . The question that arises is what kind of field would that correspond to?

By using the expression for the evanescent wave the field becomes

$$E_z(z, r, t) = \sum_{m=0}^{\infty} A_m J_0(k_m r) e^{a_m(z-\mu)} \sin(\omega t) \quad (3.33)$$

Using $m = 1, 2$ the expression is kept short and uses the same number of Bessel functions as in section z_2 to z_4 . Inserting one of the terms into the wave Equation

(3.2) the condition for k_m becomes

$$k_m^2 = -(a_m)^2 - \left(\frac{\omega}{c}\right)^2 \quad (3.34)$$

This implies a complex value for k_m and the arguments of the Bessel functions will extend into the complex plane. Values for the Bessel function when extended along the imaginary axis are still real and described by the modified Bessel functions. This would however create a discontinuity of the field at the boundary between z_1, z_2 as well as z_4, z_5 . This would make the field unrealistic in another way.

3.3 Spoke cavity parametrization

An extension the model, beyond the medium and high β -sections, has also been implemented. The section prior to the medium and high β -section in Figure 1.2 shows a section simply named spokes. This section contains the spoke cavities, which are, like the medium and high- β cavities, superconducting. The frequency of the oscillation of the field in the spoke cavities is half of that in the sections containing the elliptic cavities, namely 352.21 MHz. The spoke cavities are designed to have a geometrical β of 0.5 which mean that they are designed for lower energy particles. The field in a typical spoke cavity can be seen in Figure 3.6. In order

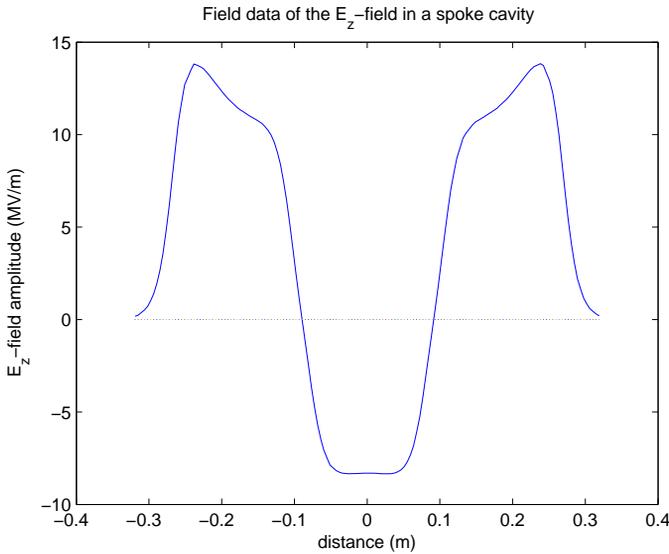


Figure 3.6: The field amplitude of E_z on the central axis, inside a spoke cavity.

to parametrize the field in the spoke cavity the same subdivision is done for that field as for the field in the elliptic cavities. The parametrization differs slightly in

the three central sections where the field is fitted to the function.

$$E_z(z, r, t) = \sum_{m=0}^{\infty} A_m J_0(k_m r) \cos \frac{\pi m x}{L} \sin(\omega t) \quad (3.35)$$

To describe the field in the spoke cavities one more cosine term is used in the expansion, i.e., $m = 1, 3, 5$. The field data shown in Figure 3.6 is taken when $r = 0$ and $\sin(\omega t) = 1$, the expression becomes.

$$E_z(z) = A \cos\left(\frac{\pi z}{L}\right) + B \cos\left(\frac{3\pi z}{L}\right) + C \cos\left(\frac{5\pi z}{L}\right) \quad (3.36)$$

Thus each spoke cavity requires three additional parameters. The fit in the spoke cavity, for the three central sections, is done using the least square method. And the same method as for the elliptic cavities is used when fitting the parameters to section z_1 and z_5 . The same radial construction is used in the spoke cavities as in the elliptic cavities, but it utilizes an additional Bessel function together with $\cos\left(\frac{5\pi z}{L}\right)$. The results of the spoke field parametrization can be seen in Figure 3.7. The error in the spoke cavity parametrization is slightly larger than the

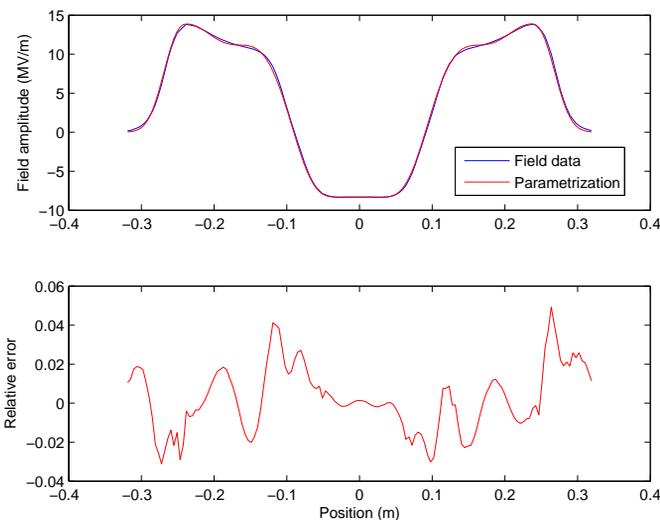


Figure 3.7: The top figure show how close the parametrization replicate the spoke field. The bottom figure show the relative error between the spoke field data and the parametrization, it is normalized over the maximum field amplitude.

parametrization for the elliptic cavities. The fit is still close and can be used to make simulations in the same manner as in the elliptic cavities.

Particle Dynamics

In this chapter a brief description of the relevant particle dynamics is presented. It is these equations that are later evaluated to predict particle paths in the ESS LINAC.

4.1 Transit Time Factor

In order to extract the coordinates of an arbitrary particle, one needs to know the coordinates of the synchronous particle. This is of importance at the entrance and exit of a cavity. A tool for this is the transit time factor T defined as

$$T(\beta, \phi) = \frac{\int E(0, z, t) dz}{\int E(0, z) dz} = \frac{\int E(0, z) \cos(\omega t + \phi) dz}{\int E(0, z) dz} \quad (4.1)$$

the term containing the time component can be coupled to the speed of a particle in the cavity via $\frac{dz}{dt} = v$. This gives that the phase at a point z in the cavity is

$$\int_{z_0}^z \frac{\omega}{\beta(z')c} dz' \quad (4.2)$$

where z_0 is the cavity entrance. With this the transit time factor becomes [7]

$$T = \frac{\int_{z_0}^{z_1} E(0, z) \cos(\int_{z_0}^z \frac{\omega z'}{\beta(z')c} dz' + \phi) dz}{\int_{z_0}^{z_1} E(0, z) dz} \quad (4.3)$$

The integration is performed over the whole cavity, z_0 is the cavity entrance and z_1 is the cavity exit. Figure 4.1 show the energy gained by a particle as a function of the phase of the electric field when the particle enters.

4.2 Particles in the cavity

The energy gained by an arbitrary particle in an elliptic cavity is in the paraxial approximation equal to

$$\begin{aligned} \Delta W &= \int \vec{F} \cdot d\vec{s} = \int q\vec{E} \cdot d\vec{s} = q \int E_z(r, s) \sin(\omega t) ds_z = \\ & q \int E_z(r, s) \sin(\omega \frac{s}{\beta c}) ds_z \end{aligned} \quad (4.4)$$

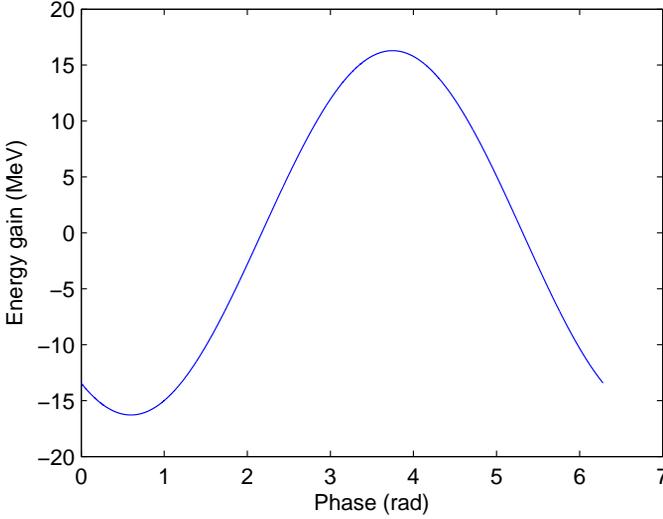


Figure 4.1: The energy gained by a particle as a function of its phase relative to the electric field when entering the cavity.

Here \vec{s} describes the path of the particle. The coordinate system is fixed in the cavity and is not centered on the synchronous particle. The direction s_z is parallel to the direction of z introduced in 2.1.1. The paraxial approximation is motivated by that the energy gain from moving in the transverse plane is small enough to be disregarded. Or in other words

$$\int E_r(s, r) \sin(\omega \frac{s}{\beta c}) ds_r \approx 0 \Leftrightarrow d\vec{s} \approx ds_z \quad (4.5)$$

Off axis there are components of the field that cause the protons to change direction, or diverge. The field, as discussed earlier, exhibits radial symmetry so it is convenient to look at the impulse given in the radial direction.

$$\Delta p_r = \int F_r dt = \int F_r \frac{ds}{\beta c} = \int \frac{q(\vec{E} + \vec{v} \times \vec{B})_r}{\beta c} ds = \int \frac{q(E_r - \beta c B_\theta)}{\beta c} ds \quad (4.6)$$

This changes the radial divergence, $r' = \sqrt{x'^2 + y'^2}$, by

$$\Delta r' = \frac{\Delta p_r}{p_z} = \frac{q \int (E_r - \beta c B_\theta) ds}{m_o \gamma \beta c} \quad [9] \quad (4.7)$$

Also, acceleration of a particle in the z -direction changes the radial divergence. The increase of velocity in the z -direction of a particle causes it to diverge less when traveling a certain distance. An increase in speed from β_0 to β would change the radial divergence r'_0 to r' by

$$r' = r'_0 \frac{\beta}{\beta_0} \quad (4.8)$$

The integration of an arbitrary particle is done by traversing it along the s -axis of the accelerator and calculating the new energy and x and y positions at a desired interval. This is done while keeping track of the phase of the oscillating field in the cavity. Eventually, at the end of the cavity, the final phase and velocity is compared to that of the synchronous particle to obtain the complete final position.

Descriptions and structure of the code

The particle integration code is written in a straightforward manner. The code is structured so variations of the problem can be implemented with only minor modifications. The code is included in the appendix.

5.1 Field parametrization

The parameters describing the field are extracted from a file containing values of the electric field on the z -axis. Some short notes are made here on that program. These considerations originate from the fact that the parameters are generated from a discrete set of points. When generating the coefficients, A , B and L in expression (3.17), some steps were taken to keep high accuracy even when the data describing the E_z -field was sparse. When estimating L , which gives the scaling of the cosine functions, the distance between the zeroes of the field is used. The zeroes of the field, $f(z)$, is found by linear fits between discrete points where

$$f(z_i) > 0 \quad \text{and} \quad f(z_{i+1}) < 0 \quad (5.1)$$

or vice versa in order to find the points corresponding to the zeroes of the function.

In addition, the boundaries between z_1 and z_2 as well as between z_4 and z_5 seen in Figure 3.2 are placed where the cosine functions are at an extreme point. To find the points corresponding to an extreme point from a discrete set of values, second order polynomials were fitted to the discrete extreme points. The discrete extreme points are found by requiring

$$|f(z_n)| > |f(z_{n-1})| \quad \text{and} \quad |f(z_n)| > |f(z_{n+1})| \quad (5.2)$$

For a sparse field the discrete maximum $f(z_n)$ is not close enough to the true maximum of the field. For that reason the polynomial

$$Az^2 + Bz + C \quad (5.3)$$

is made to fit

$$\begin{aligned} f(z_n) &= Az_n^2 + Bz_n + C \\ f(z_{n-1}) &= Az_{n-1}^2 + Bz_{n-1} + C \\ f(z_{n+1}) &= Az_{n+1}^2 + Bz_{n+1} + C \end{aligned} \quad (5.4)$$

The extreme point is then found by setting

$$\frac{d}{dz} (Az^2 + Bz + C) = 0 \Leftrightarrow z = -\frac{B}{2A} \quad (5.5)$$

This gives both the z -coordinate of the maximum z_{\max} , and the value at the maximum $f(z_{\max})$.

Two sets of parametrizing code were written in which the parameters A and B in section z_2 and z_4 in Figure 3.2 were found. One code uses the least square method and that gave a slight discontinuity of the field. Another code was therefore written. It fits the parameters A and B by two conditions. The first condition is on the amplitude $A+B$ to be equal to the value of the extreme point found in Equation (5.5) inserted into Equation (5.3). The second condition is to fit the derivative at the boundary between sections z_2, z_3 and between z_3, z_4 . Additionally, a field parametrization for the spoke cavities was also done. When finding the parameters for the spoke cavity an additional parameter is introduced in sections z_2, z_3 and z_4 . For these parameters the least square method was used in all three sections, z_2, z_3 and z_4 .

5.2 Particle tracking methods

In order to obtain a fast code one needs to reduce the amount of calculations and increase the accuracy of the calculations that are done. Two different methods were implemented and compared. The first one is a standard second order Runge Kutta algorithm known as the midpoint method. The second method utilizes the structure of the parametrization to see if computational time could be reduced that way.

5.2.1 General Idea

This section describes the general structure of the particle tracking code. The coordinates of each particle is read from a file. In the file the particle coordinates are given relative to the synchronous particle. The initial position of a particle relative to the electric field and the cavity is calculated from the coordinates of the particle together with the phase and energy of the synchronous particle at the entrance of the cavity. The transverse forces and energy gains are then calculated in steps of Δs after which a new direction and energy are calculated. This process is repeated until the particle reaches the end of each cavity where it is once again changed to the coordinate system of the synchronous particle. When the particles have passed through all the cavities their coordinates are saved in a file.

5.2.2 Particle Coordinates Within the Cavity

The particle simulation inside the cavity is done for a set of longitudinal positions. This mean that the position in the z -direction is not a free variable. Instead velocity and time is used as free variables. This relates the position of an arbitrary particle to the position of the synchronous particle. The coordinate that replaces z is ϕ , which describes the phase of the field in the cavity when a particle is at a

certain position. At the entrance ϕ is calculated using the phase of the synchronous particle at the entrance, by

$$\phi = \phi_s - \omega \frac{z\sqrt{1 + (x')^2 + (y')^2}}{\beta c} \quad (5.6)$$

The coordinates x' and y' are the divergence of the particles defined in Equation (2.2). This equations says that if a particle is at a distance $z\sqrt{1 + x'^2 + y'^2}$ from the synchronous particle which is at the cavity entrance, the particle will enter the cavity at a different time depending on its own speed βc . The time difference is translated to a phase difference. Before finding the phase difference the velocity of the particle is needed. It is found by solving

$$z' = \frac{p - p_s}{p_s} = \frac{\beta\gamma - \beta_s\gamma_s}{\beta_s\gamma_s} = \frac{\frac{\beta}{\sqrt{1-\beta^2}} - \frac{\beta_s}{\sqrt{1-\beta_s^2}}}{\frac{\beta_s}{\sqrt{1-\beta_s^2}}} \quad (5.7)$$

The energy of the synchronous particle is known at the entrance of the cavity. The velocity of the synchronous particle, β_s , is determined from Equation (2.3). This implies that, the particle integration is done at varying time intervals depending on the position of the particle, relative to the synchronous particle.

After each step along the cavity the phase is also increased by $\frac{\omega\Delta s}{\beta c}$, and at the end of the cavity the phase of the particle is compared to the phase of the synchronous particle and the energy of the particle is compared to the energy of the synchronous particle. By using Equation (5.6) and (5.7) the coordinates z and z' are determined. The coordinates x , x' and y , y' are used to describe the transverse dynamics, and are treated the same way as in chapter 4.

5.2.3 Standard method

The midpoint method is a second order Runge Kutta method It requires that the problem is formulated as a first order differential equation

$$\frac{dy}{dt} = f(t, y(t)) \quad (5.8)$$

The variable that is of interest in the cavity is the speed, or the β , of a particle. The differential equation is solved over position, s , and not over time, t . By knowing the β of a particle, a connection between a translation in space to one in time or vice versa, is obtained.

$$\frac{d\beta}{ds} = f(s, \beta(s)), \quad \frac{d\varphi}{ds} = g(s, \beta(s)) \quad (5.9)$$

The phase in the cavity is

$$\varphi = \varphi_0 + \int_{t_0}^t \omega dt' \quad (5.10)$$

where t is the time that particle has spent in the cavity and φ_0 is the entrance phase. The time is coupled to the speed via

$$dt = \frac{ds}{\beta(s)c} \quad (5.11)$$

This, combined with Equation (5.10) gives

$$\varphi = \varphi_0 + \int_{s_0}^s \frac{\omega}{\beta c} ds' \quad (5.12)$$

In differential form this reads

$$\frac{\partial \varphi}{\partial s} = \frac{\omega}{\beta c} \quad (5.13)$$

The s -dependence of β can be derived from

$$\frac{d\beta}{ds} = \frac{d\beta}{dE} \frac{dE}{ds} \quad (5.14)$$

The factor $\frac{\partial E}{\partial s}$ can be found by using the Lorentz force law and using Equation (2.3). The relationship between force and energy is $dE = F ds$. Equation (2.3) is rewritten as

$$E = \gamma E_0 \Leftrightarrow 1/\gamma = \frac{E_0}{E} \Leftrightarrow 1 - \beta^2 = \left(\frac{E_0}{E}\right)^2 \Leftrightarrow \beta = \sqrt{1 - \left(\frac{E_0}{E}\right)^2} \quad (5.15)$$

This gives the required expression, were $E_0 = m_0 c^2$. Differentiating β with respect to energy gives

$$\frac{d\beta}{dE} = \frac{1}{\sqrt{1 - \left(\frac{E_0}{E}\right)^2}} \frac{E_0^2}{E^3} = \frac{1}{E_0 \beta \gamma^3} \quad (5.16)$$

The complete equation can be written as

$$\frac{d\beta}{ds} = \frac{E_z(s, r, t)}{m_0 \beta \gamma^3} = \frac{E_z(s, r) \sin(\phi_0 + \int_{s_0}^s \frac{\omega}{\beta c} ds')}{m_0 \beta \gamma^3} \quad (5.17)$$

The midpoint method gives an approximate solution of (5.8). The way it does so is summed up in equation (5.18).

$$y_{n+1} = y_n + f(t_n + h/2, y_n + \frac{1}{2} h f((t_n, y_n))) \quad [12] \quad (5.18)$$

where y_{n+1} is the estimated value of the coordinate y after one step in time, h is the step size used, and t_n is the time from where the step is evaluated.

5.2.4 Semi-analytic method

This method utilizes the fact that the three middle sections of the field are expressed terms of $\cos\left(\frac{\pi m z}{L}\right)$ functions. The energy gain for a particle in the field is

$$\Delta W_z = \int_{\Delta z} \left(A J_0(k_1 r) \cos\left(\frac{\pi z}{L}\right) + B J_0(k_3 r) \cos\left(\frac{3\pi z}{L}\right) \right) \sin(\omega t) dz \quad (5.19)$$

With the Bessel function assumed constant under the interval Δz one ends up with two integrals of the form

$$\int_{\Delta z} \cos\left(\frac{\pi z}{L}\right) \sin(\omega t) dz \approx \int_{\Delta z} \cos\left(\frac{\pi z}{L}\right) \sin\left(\frac{\omega z}{\beta c}\right) dz \quad (5.20)$$

This is an integral that has an analytic solution under the additional assumption that β is constant over the interval Δz . This approximation is fair since the speed changes far slower than the field and since the radial divergence is small. The semi-analytic method is implemented with the goal of being able to predict the dynamics accurately at a lower resolution, and therefore hopefully faster. There will be three different, but similar, integrals. Each integral corresponds to particle dynamics related to a component of the field

$$\begin{aligned} E_z(z) &\propto \int \cos(kz + \phi_1) \sin(wz + \phi_2) dz \\ E_r(z) &\propto \int \sin(kz + \phi_1) \sin(wz + \phi_2) dz \\ B_\theta(z) &\propto \int \cos(kz + \phi_1) \cos(wz + \phi_2) dz \end{aligned} \quad (5.21)$$

where the coefficients k , w , ϕ_1 and ϕ_2 depend on the particle coordinates inside the cavity and the phase of the field. It is not necessary to evaluate all these integrals since one can relate cosine and sine via phase shifts by $\pi/2$. Therefore the only integral that will be evaluated is the first one

$$\int_{z_1}^{z_2} \cos(kz + \phi_1) \sin(wz + \phi_2) dz = \quad (5.22)$$

$$\int_{z_1}^{z_2} \frac{e^{i(kz+\phi_1)} + e^{-i(kz+\phi_1)}}{2} \frac{e^{i(wz+\phi_2)} - e^{-i(wz+\phi_2)}}{2i} dz = \quad (5.23)$$

$$\begin{aligned} &= \int_{z_1}^{z_2} \frac{1}{4i} (e^{i(kz+\phi_1+wz+\phi_2)} - e^{i(kz+\phi_1-wz-\phi_2)} + \\ &\quad e^{-i(kz+\phi_1-wz-\phi_2)} - e^{-i(kz+\phi_1+wz+\phi_2)}) dz = \end{aligned} \quad (5.24)$$

$$= \int_{z_1}^{z_2} \frac{1}{2} \sin((k+w)z + \phi_1 + \phi_2) - \sin((k-w)z + \phi_1 - \phi_2) dz = \quad (5.25)$$

$$= \frac{1}{2} \left(\frac{\cos((k-w)z + \phi_1 - \phi_2)}{k-w} - \frac{\cos((k+w)z + \phi_1 + \phi_2)}{k+w} \right) \Big|_{z_1}^{z_2} \quad (5.26)$$

This is true as long as $k \neq w$ and $k \neq -w$, since if for example $k = w$ the term containing $k - w$ can be treated as a constant.

$$\int_{z_1}^{z_2} \sin((k-w)z + \phi_1 - \phi_2) dz = \int_{x_1}^{x_2} \sin(\phi_1 - \phi_2) dz = \sin(\phi_1 - \phi_2)(z_2 - z_1) \quad (5.27)$$

This shows another viable method to perform integration the three middle sections. The integral describing energy gain in the initial and final section is not analytically treated under the same assumptions.

For the three middle sections, z_2 , z_3 and z_4 , the semi-analytical method is used, the value over each step is taken from Equation (5.26). This is combined

with the values of the Bessel function that are generated by a function from the GSL [2](Gnu Scientific Library). For the sections z_1 and z_5 numerical integration is required, and the midpoint method is used there.

The so called semi-analytical method described above is so far only of first order. To increase the order a method similar to that of the midpoint method is used. At each step an initial β denoted by β_i is used to calculate the final β , denoted β_f . A third β is then estimated as $\beta = (\beta_f + \beta_i)/2$. By using this third β when calculating the actual energy gain, the rate of convergence of the method is increased from first to second order. The cost of improving the method to a second order is an additional energy gain evaluation.

5.3 Input for simulation

The simulations require three kinds of input.

1. The first is the input particles that one wish to simulate. Each particle is defined by 6 coordinates, see section 2.1.1.
2. The second is the parameters of the field described in the section 3.2, these parameters are a property of the cavity used in the linac.
3. The third input necessary is the input phase and velocity of the synchronous particle. It is a design property of the accelerator and is a result from the cavities used and how the designer decides on the phase between the cavities.

The synchronous phase is defined relative to the entrance phase of the particle that would gain maximum energy inside the cavity. The entrance phase of the field that gives a particle a maximum energy gain in the cavity is defined as 0. An advantage is that it does not matter if one defines the oscillating field as a sine function or a cosine function. It is standard to have a negative synchronous phase in linear accelerators to achieve longitudinal focusing of the bunch [7].

A program has been written that determines the entrance phase of the particle that gains the maximum energy in a cavity. This program returns synchronous phase as a function of cavity parameters, entrance phase and energy. It can do so for an arbitrary amount of cavities. It uses exactly the same kind of integration as the particle simulation program, so to ensure that if one runs the particle simulation program with a particle at a starting position $(0, 0, 0, 0, 0, 0)$ it would remain at that position, since it would be identical to the synchronous particle in position and velocity. It is important that the same resolution is set in the determination of the entrance phase and velocity of the synchronous particle as in the simulation of the motion of all the particles. If a simulation is run at a resolution different from the program extracting the entrance phase and velocity of the synchronous particle some obvious errors would surface.

This chapter presents results from simulations using the code. The two methods that have been implemented are compared with each other to see which is best. And shows simulations compared with a linear model. Benchmarking of the code has been performed versus the same code written in JAVA. The output from the simulations are compared to that of a linear transfer matrix model [11]. Additionally, some of the output is shown from larger simulations that display the general dynamics of particles in a linear accelerator.

6.1 Code performance and validity

In this section the best method implemented in this thesis for particle simulation is determined. The code is also compared to the linear transfer matrix model. This section also compares the results from the benchmarking of the code in C to the of one in JAVA.

6.1.1 Resolution dependency

In order to make the choice between the semi-analytical and the standard RK2 method performance tests have been constructed. The best way to define how well a method perform is to determine the rate of convergence and computational cost with respect to time. The problem does not have an analytic solution and finding the error is therefore not an exact matter. The error is defined as the difference between the output at a certain resolution and the output at the highest resolution that was simulated. Both methods converge to the same solution. Figure 6.1 shows how fast the RK2 and semi-analytical method converge. The RK2 has a third order convergence whereas the semi-analytical method displays a convergence rate of second order. An examination of the error as a function resolution and time can be seen in Figure 6.1. It shows how the semi-analytical method produces a smaller error at low resolution as expected. Computational time is however the relevant factor and since RK2 is faster and converges faster than the semi-analytical method it is the best choice for future simulations. Additional positive attributes of the RK2 method are that it is simpler, it is easier to modify and easier to understand. The semi-analytical method requires some carefulness when

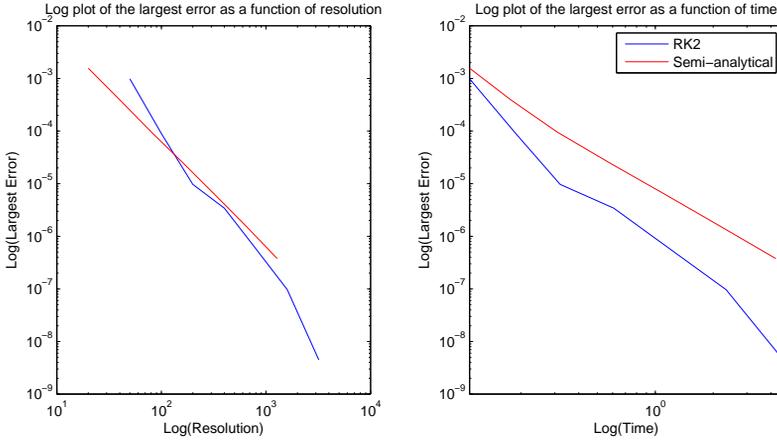


Figure 6.1: Comparison of the two methods implemented for particle tracking. The blue line marks the Runge Kutta and the red line marks the semi-analytical method. The second order Runge Kutta comes out as superior when looking at time it takes to run the simulation.

setting up the integrals in Equation (5.26) since they vary depending on which section of the cavity the particle is in.

Comments on resolution and step method

The results from the simulations show that the Runge Kutta method surpasses the semi-analytical method, a contributing factor to that is that it converges at a rate higher than second order. This is in one way a good thing, but when something acts unpredictable it can often show itself in other places, where it is not necessarily an advantage. The third order convergence is present when making both on and off-axis simulations, suggesting that the issue does not depend on any unpredicted coupling between longitudinal and transverse dynamics. This issue stretches further than the second order Runge Kutta method. When implementing higher order Runge Kutta methods it is suspected of preventing the right order of convergence. In fact, both third and fourth order Runge Kutta methods have been implemented, but both have converged at an order of 2.7, a slightly lower convergence rate than the second order. A definitive explanation for the anomalous rate of convergence of the second order method has not been found. However increase of the phase is hard to account for accurately since it requires knowledge of the speed $\beta(s)$ under the interval Δs .

$$\Delta\phi = \int_{s_0}^{s_0+\Delta s} \frac{\omega}{\beta(s)c} ds \quad (6.1)$$

It is suspected to be the cause of the high rate of convergence. Various ways of fixing it have been attempted, among them Taylor expansions, without success.

Another point to consider with these methods is that they are not symplectic. A symplectic method would keep the phase space volume constant. There are however no symplectic higher order RK-schemes that are explicit.

6.1.2 Comparing with a linear model

Making a few reality checks regarding how far a particle can move in one meter, how much energy can be gained etc. is not enough to know if the output is correct. For this reason the code has also been compared to a linear model. This was done initially as a way to validate the code but later to see where the linear model fails to describe the dynamics. A standard way to describe beam dynamics is to make linear models and represent dynamics by matrices, where particle coordinates are represented in the same way as explained in section 2.1.1. A physical component, like a cavity, is then described by a 6×6 matrix that maps the particle to a new position further down the beamline [13]. An example of a transfer matrix can be seen below in equation (6.2), see [13],

$$\begin{pmatrix} 1 & l & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & l & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{l}{(\gamma\beta_s)^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x'_0 \\ y_0 \\ y'_0 \\ z_0 \\ z'_0 \end{pmatrix} = \begin{pmatrix} x \\ x' \\ y \\ y' \\ z \\ z' \end{pmatrix} \quad (6.2)$$

This matrix maps the coordinates of a particle from the entrance to the end of an section with zero field of length l , where the synchronous particle has the speed β_s . The transfer matrices for cavities with present fields are more complicated, but a feature for all transfer matrices is that there is no coupling between the different planes of motion. The program TraceWin was used to generate the 6×6 matrices corresponding to various cavities at different positions throughout the cavity. The comparison was made using a cavity just at the end of the linac, a high beta cavity. The in the cavity is shown in Figure 3.1 and the synchronous phase was -15° .

The first simulation is one where the input particles hold a uniform distribution on the x-axis, with all other coordinates being zero. The output is shown in Figure 6.2. The fact that the difference between the two methods is small when simulating particles off-axis suggests that a linear model is a good approximation there. A thing that can not be captured by a linear model is the coupling between the transverse and longitudinal plane of motion. This coupling is present in actual accelerators, as well when simulating particle paths using the non linear method implemented. The data in Figure 6.3 is from the same simulation as the data in Figure 6.2. The interplay between longitudinal dynamics and transverse dynamics can be seen in Figure 6.3. Figure 6.3 shows how the particles far from axis have gained less energy and started to slip behind the synchronous particle, it also gives a hint on how the field is focused at the center of the cavity since at the exit the particles near the center have a higher momentum. This is however only showing the dynamics over one cavity, and particles that fall behind will generally feel a stronger field in the next cavity due to the choice of synchronous phase.

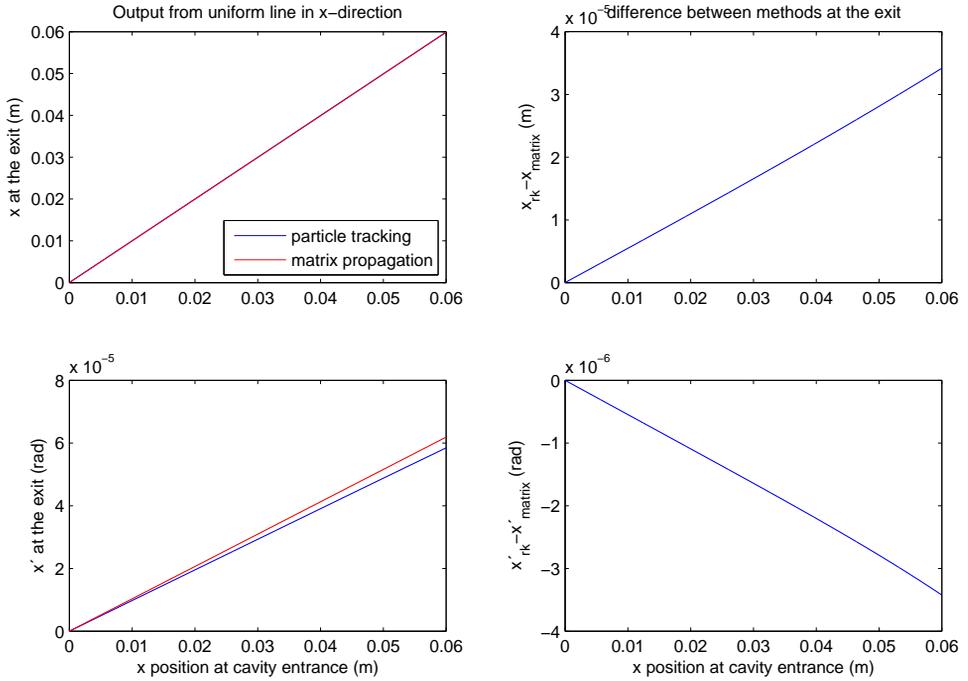


Figure 6.2: The two plots on the left show the output from both the matrix model and the particle tracking model. The output is hard to distinguish in the top left figure but a deviation in the x' -coordinate is visible between the methods at growing radius in the bottom left plot. The radius of the iris of the cavity is roughly 6 cm, meaning that particles further away than that will hit the walls. The two plots on the right show the difference between the methods. A difference in both position and divergence is clear, and the difference appear to be linear.

Another simulation has been performed with a uniform distribution in the z -direction. The results can be seen in Figure 6.4. The final position remains more or less the same relative to the synchronous particle, this is because the simulations are only over one cavity and the particles start at the same velocity as the synchronous particle. The difference in relative momentum, z' , is more pronounced. It is clear that the linear model does not hold up well when looking at longitudinal dynamics. Looking at the difference at the exit in the output of z and z' between the two methods, one can see not only a non-linear behavior, but also a non-symmetrical behavior, which is expected to be seen in the nonlinear model. Figure 6.4 shows how a particle that is slightly behind the synchronous particle at the cavity entrance has a slightly higher energy at the exit of the cavity.

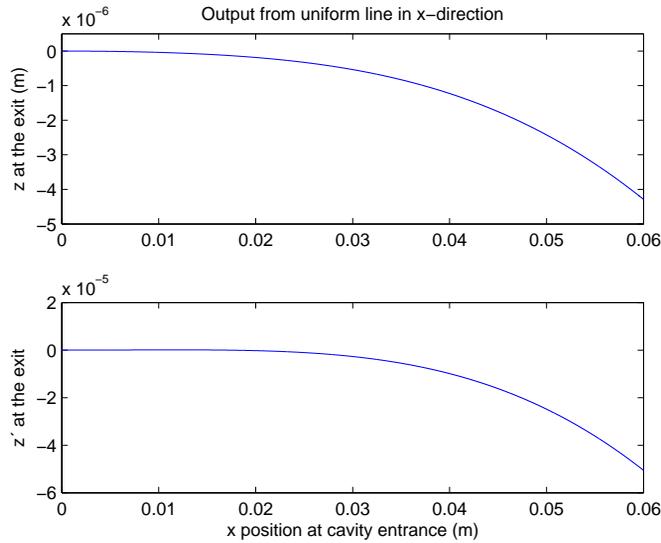


Figure 6.3: These two plots show how the position in z and relative momentum z' as a function of the distance the particle holds from the axis when passing through a cavity. It shows that particles further from the center will lose energy relative to the particles near the center.

6.1.3 Benchmarking in C and JAVA

A discussion is ongoing at ESS whether the online and offline model should be implemented in C or in JAVA. As grounds for deciding this, much of the code written is benchmarked in both languages. ESS is working with the software development firm Cosylab to construct the online model. Cosylab also help with the benchmarking of codes. The code for the RK2 method, after being decided to be superior to the semi-analytic method was benchmarked with the help of Ivo List from Cosylab. The results of the benchmarking can be seen in Figure 6.5. The results are obtained at different resolutions and clocked 100 times. The conclusion is that JAVA is roughly 60% slower than C [14]. The benchmarking also shows that the calculations only match up to 6 to 7 digits. This error is likely due to the different ways C and JAVA treat floating points in mathematical operations, e.g. finding the the square root.

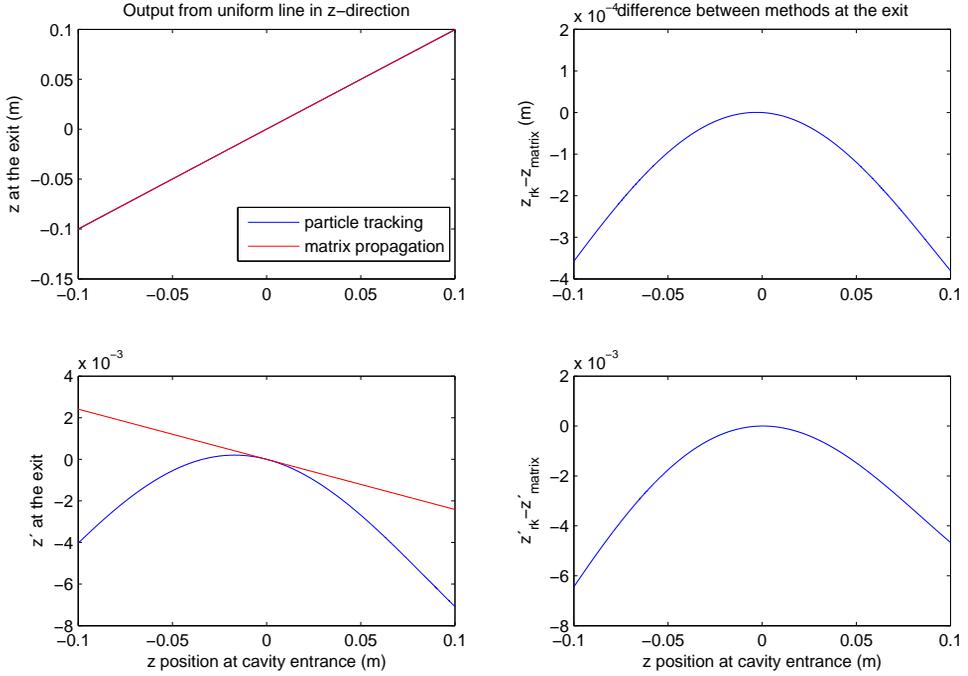


Figure 6.4: The two figures on the left show the output of a uniform distribution on the z -line, in the top left plot the output position is plotted against input position. The bottom left plot show the relative momentum at the cavity exit against initial position. The plots on the right show, at the top the difference in z , and at the bottom the difference in z' , between the linear and non-linear model.

6.2 Particle Simulations

A few different simulations have been run to showcase and validate the output of the model.

6.2.1 End to end simulation

End to end simulations refer to simulations that run through one or many parts of the accelerator. A part of the accelerator that is not represented in the implemented model is the accelerator optics. The optics, when discussing accelerators, usually refer to the use of quadrupole magnets to change particle paths. A quadrupole can be used to focus the particle beam in the x' or y' direction. A quadrupole magnet can however only be made to focus the beam in one plane at a time, when focusing in one plane the beam will defocus in the other plane. A

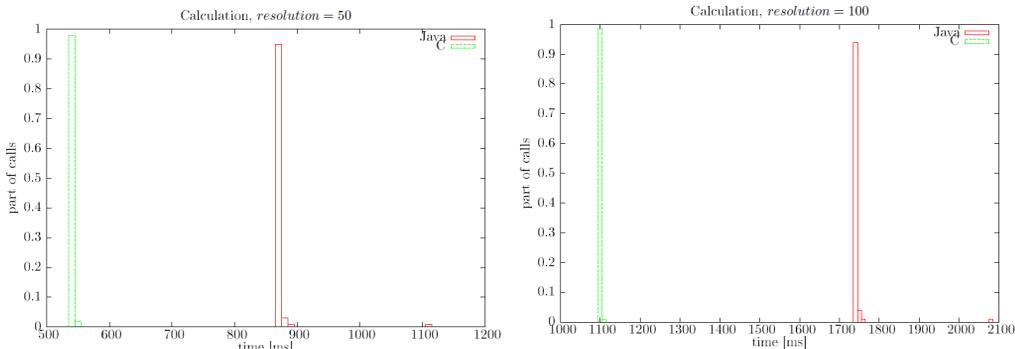


Figure 6.5: Comparison between the same code in JAVA and in C. The results show that the code is faster in C.

quadrupole magnet that focuses the beam in the horizontal plane is defined as a focusing magnet, and a quadrupole that focuses the beam in the vertical plane is defined as a defocusing magnet. An overall focusing effect can be achieved by setting up what is called FODO-lattices, combinations of FOcusing and DefOcusing quadrupoles [11].

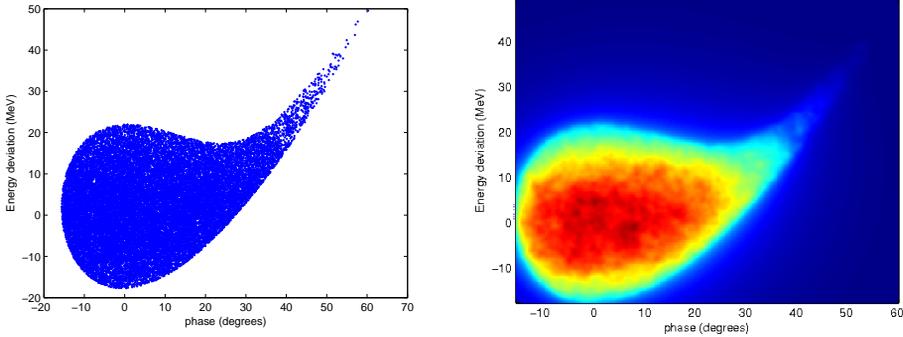
The high β -section of the ESS LINAC will contain in total 120 superconducting cavities. These cavities are placed in sections of 8 cavities before being interjected with FODO-magnets, all in a repeating pattern [1]. Since the optics is not included in the model, end to end simulations of off-axis particles are not possible without using other methods that model focusing. The on-axis particles are not affected by the optics, and that makes them ideal for end to end simulation.

A relevant property when designing an accelerator is the acceptance. Accepted particles are defined as the particles that make it through the entire length of the accelerator while still being close to the synchronous particle. This means that the accepted particles are at the entrance of the accelerator section, it does not refer to coordinates at the exit of it.

The accepted particles from a simulation of 100 000 particles in a two dimensional Gaussian distribution in the z, z' plane can be seen in Figure 6.6. For the simulation the synchronous phase was set to -15° .

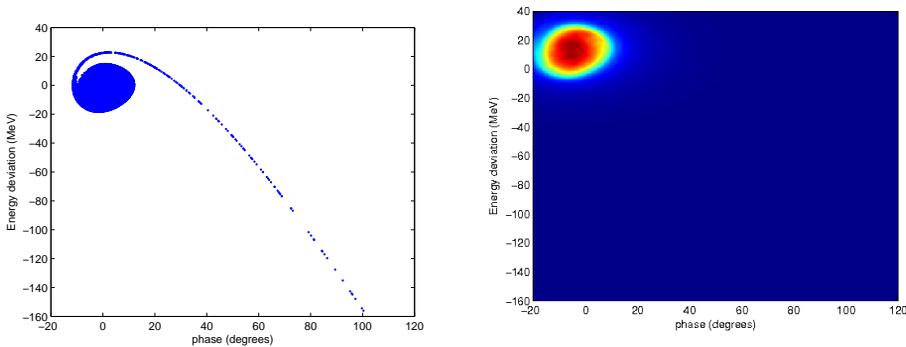
Figure 6.6 shows a behavior characteristic for LINACs known as the golf club, which further validates the model. The energy gained by the synchronous particle is close to the one seen in 1.2, a full correspondence is not expected since the synchronous phase is not -15° for the entire high β section.

Figure 6.7 shows the phase space distribution of particles at the end of the high β -section. It shows a small tail of particles that are starting to fall behind.



- (a) The accepted particles displayed as dots at the entrance of the high- β section.
- (b) The particle density of the accepted particles at the entrance of the high- β section.

Figure 6.6: Figure displaying the distribution of accepted particles.



- (a) The accepted particles at the end of the high beta section displayed as dots.
- (b) The particle density of the accepted particles at the end of the high beta section.

Figure 6.7: Figure displaying the distribution of accepted particles at the end of the high beta section.

Further development and discussion

This chapter contains a discussion of the model. It also contains reflections on further work, and how the model can potentially be developed in the future.

7.1 The use of the model

The model presented here can simulate the behavior of 10^4 particles through the entire high beta section of the ESS LINAC, at a time on the order of a minute. That is most likely not fast enough for the online model. The online model could utilize the code in a useful manner, but would require small particle bunches for full simulations in reasonable times. It is hard to make valuable statistical predictions from simulations with few particles, such predictions would be the standard deviation of the particles or the correlations between the z -position and momentum deviation of the particles. In order to make simulations in real time with the method presented here, which is required by online model, one would have to know exactly which particles one is interested in and simulate just them. A way to decrease the number of particles required to simulate dynamics in real time is to choose particles on the envelope of the bunch. The envelope is typically defined as the standard deviation of the particles of the bunch. By taking the envelope of the bunch, for example in the z and z' -direction at the entrance of a section. This would give a two dimensional distribution, since the coordinates x , y , x' and y' would all be zero. The number of particles required to describe the envelope of the bunch are far fewer than the total number of particle in the bunch. It is however not clear if a particle that starts on the envelope of the bunch remains on the envelope throughout the simulation. Larger particle bunches can be used to simulate smaller parts of an accelerator at low times. This would for example be the case when simulating the behavior of particles passing the 8 cavities between the FODO sections. The model is fairly computationally heavy. In order to make a simulation of 10^6 particles through the entire high beta section at a resolution of 100 steps per cavity, a total number of $1.2 \cdot 10^{10}$ steps are required, where at each step a new energy and radial impulse is calculated. With a mid range laptop, as was used for this thesis, it takes more than an hour. With better hardware the run time can be reduced, but by how much remains to be seen. The model does however appear to accurately predict particle paths, and that will become useful when comparing and studying validity of other models.

7.2 Improving the parametrization

The parametrization introduced here in section 3.2 requires 15 parameters. In this section a possible way to reduce the number of parameters is presented. The idea is that instead of having a field subdivided into five sections that fit the actual field well, the fit should be made against the TTF function for a simpler field. The parametrization

$$A \cos\left(\frac{\pi x}{L}\right) + B \cos\left(\frac{3\pi x}{L}\right) \quad (7.1)$$

would still be used, but would describe the entire field without making subdivisions into several such functions. Instead of fitting Equation (7.1) to the field points, it would instead be fitted so it replicates the results of the TTF function defined by Equation (4.3). The fit could be made by first calculating the TTF using actual field points. The parameter L would be the same as assigned for section z_3 and A and B would be tuned so the energy gains of particles would be consistent with the TTF. The same extension of describing the field in the radial direction with Bessel functions would be made. If this was possible, the field and the code would be further simplified and the particle tracking could possibly be made faster. With such a fit it would be possible for the semi-analytic method to surpass the RK 2 method in accuracy and speed, since it is slowed down by having to consider which section it is in.

7.3 Improving the particle integration

As mentioned in section 6.1.3 the calculations match only up to 6 or 7 digits because of the low accuracy when using double precision [14]. This can possibly be mended by restructuring the code or reconsidering the order of some operations. The error is mostly pronounced when looking at radial divergence of particles. When calculating the radial divergence after each step, a small increase or decrease of the radial divergence is done. And when adding a small number to a large number information is lost in the decimal expansion of the small number, and a limit in accuracy is reached. For this reason the radial divergence is only solved to first order, this is because it is close to the maximum accuracy of the method at low resolution. It saves some computational time if the radial divergence is not considered to second order.

Another approach that can improve the computational speed would be to implement it in a parallel processing framework. The particle integration is structured so that it can simulate particles independently of each other. This means that the calculations can be performed in parallel. Collective effects such as space charge would still be a bottleneck.

7.4 Extending the code

In order to improve the model there are a few things that can be done. The main issue with the model as it stands now, which reduces the utility of its results, is the fact that it does not take particle-particle interaction and space charge effects

into account. There are different ways to implement models for particle interaction [4][15]. In order to make a good implementation of it, one would require that the particle tracking is solved over a grid in time, and not in space. It might seem like a big step, but the changes would be small in the RK framework. For example,

$$\frac{d\beta}{ds} = \frac{d\beta}{dt} \frac{dt}{ds} = \frac{1}{\beta c} \frac{d\beta}{dt} \Leftrightarrow \frac{d\beta}{dt} = \frac{cE_z}{\gamma^3 m_0} \quad (7.2)$$

and the suspected problem when increasing the phase would possibly cease to exist since one would no longer need to couple velocity to time in the same manner. This would hopefully allow for the successful implementation of higher order RK schemes. This could possibly be combined with a parallel process implementation. Where a few steps in time is performed for the particles on parallel processors before a bunch profile is calculated. The bunch profile would give impulse from particle-particle interactions as well as calculating the effect of the bunch shielding itself from the outside electric field.

References

- [1] S. Peggs, *ESS Technical Design Report*, April 2013.
- [2] M. Galassi *et al*, *GNU Scientific Library Reference Manual*, 3rd Ed., 2011.
- [3] W.E. Fischer, *Application of Spallation Neutron Sources* Paul Scherrer Institut, Villigen, Switzerland, Proceedings of EPAC, 2002.
- [4] E. Laface, *ESS Linac Simulator: A first benchmark with TraceWin*, Proceedings of IPAC, Shanghai, China, 2013.
- [5] A. Einstein, *Zur Elektrodynamik bewegter Körper*, Annalen der Physik 17: 891, 1905; English translation: G.B. Jeffery, W. Perrett, *On the Electrodynamics of Moving Bodies*, 1923.
- [6] D.J. Griffiths, *Introduction to electrodynamics*, 3rd ed., Prentice Hall. 1999.
- [7] T.P. Wangler *RF Linear Accelerators*, 2nd. ed., WILEY-VCH Verlag GmbH Co, 2008.
- [8] K.S. Krane, *Introductory Nuclear Physics*, 2nd. ed. John Wiley Sons, Inc, 1988
- [9] M. Eshraqi, G. Franchetti, A. Lombardi, *Emittance control in rf cavities and solenoids*, Physical Review Special Topics - Accelerators and Beams, 2009
- [10] G. Sparr, A. Sparr *Kontinuierliga system*, edition 2:6 , Replika Press Pvt Ltd, India 2009
- [11] K. Wille, *The Physics of Particle Accelerators*, Oxford University Press, 2000.
- [12] S. Endre, M David, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.
- [13] D. Uriot, *TraceWin*, CEA Saclay DSM/Irfu/SACM/LEDA CEN Saclay 91191 Gif sur Yvette, February 15th 2013.
- [14] I. List, *Benchmark Cavity model in C vs Java*, Cosylab for ESS, September 6, 2013.
- [15] K.Y. Ng, *The transverse Space-Charge force in tri-gaussian distribution*, Fermilab-TM-2331-AD, 2005

Field parametrization

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*THIS VERSION ASSUMES THAT ONE KNOWS THE LENGTH OF THE CAVITY
AND EQUIDISTNACE BETWEEN FIELD POINTS*/
/*If the field is given with (x,E(x)) it can be added in the line
where the field values are read and let those define the
position vector instead of the vector constructed*/

#define pi 3.14159265359
int main(int argc, char **argv)
{
    FILE *input_file; /* this is the file name */
    FILE *output_file; /*this will be the file i write to*/
    double *position; /* this is the Z position from the file */
    double *field; /* this is the E field from the file */
    int i;
    int number_elements; /*nr of field points*/
    int number_of_zeros=0;
        /*this program assumes even number of zeros*/
    double l=0;
    double x_zero[20]; /*for the x-value and index*/
    int x_zero_i[20]; /*is the index of the x-value
        to the left of the zero*/
    double XTf[2][2]; /*matrices used for least square fitting*/
    double XTf[2]; /*vector for least square fitting*/

    double A_mid,B_mid,l_mid; /*Parameters*/
    double A_init,B_init,l_init;
    double A_final,B_final,l_final;
    double sigma_init,p_init;
```

```

double sigma_final, p_final;

double a, b, c; /*Parameters for second order fit*/
double max_i, max_f, field_max_i, field_max_f;
    /*position of first and last field maximum,
    value at those positions*/

int max_i_i, max_f_i; /*index of initial and final maximas*/
int i1, i2; /*variables for keeping track of indices*/
double length=1.515;
double dx;
double df_i, df_f; /*derivative value*/
char buffer[1024]; /* this is a dummy buffer */

if((input_file=fopen(argv[1], "r"))==NULL)
    /* if the input file cannot be open the program quits*/
{
    printf("Please specify an input file.\n");
    return -1;
}

number_elements=0;

/* fgets read one line at time and put the content into the buffer.
for the moment we are not interested into the content but we
want just to count the number of lines, so, for every read we
increase the number_elements */
while(fgets(buffer, 1024, input_file) != NULL)
{
    number_elements++;
}
dx=length/(number_elements-1); /*splits up the interval*/

position=(double *)malloc(sizeof(double)*number_elements);
field=(double *)malloc(sizeof(double)*number_elements);

rewind(input_file);

/* let 's fill the vectors */

for (i=0; i<number_elements; i++)
{
    if (fscanf(input_file, "%lf", &field[i])==EOF)
    {
        printf("Something is going wrong.\n");
        return -1;
    }
}

```

```

}
/* creates the position grid*/
for (i=0;i<number_elements;i++)
{
    position[i]=dx*i-length/2;
}
fclose(input_file);
/*Now that the computer has the position and field it finds
the zeros of the field to get the L in cos(pi*x/L)*/
for (i=0;i<number_elements-1;i++)
{
    if ((field[i]<0&&field[i+1]>0)|| (field[i]>0&&field[i+1]<0))
    {
        x_zero[number_of_zeros]=position[i+1]-field[i+1]*
        (position[i+1]-position[i])/(field[i+1]-field[i]);
        /*linear extrapolation to find points where axis is cut*/
        x_zero_i[number_of_zeros]=i; /*saves index before the zero*/
        number_of_zeros+=1;
    }
}

/*for (i=1;i<number_of_zeros;i++)
{
    l+=x_zero[i]-x_zero[i-1];
}
l/=number_of_zeros-1;
printf("L is %.5E \n",l);*/
l_mid=(x_zero[number_of_zeros-1]-x_zero[0])/(number_of_zeros-1);
/*finding the l in cos(pi*x/l) for the middle section*/

/*Implementation of the least square method*/
for (i=x_zero_i[0]+1;i<x_zero_i[number_of_zeros-1];i++)
{
    XTX[0][0]+=pow(cos(pi*position[i]/l_mid),2);
    XTX[1][1]+=pow(cos(3*pi*position[i]/l_mid),2);
    XTX[0][1]+=cos(3*pi*position[i]/l_mid)*
        cos(pi*position[i]/l_mid);
    /*symmetry gives that XTX[1][0] is not needed*/
    XTf[0]+=cos(pi*position[i]/l_mid)*field[i];
    XTf[1]+=cos(3*pi*position[i]/l_mid)*field[i];
}
A_mid=(XTX[0][0]*XTf[0]+XTX[0][1]*XTf[1])/
        (XTX[0][0]*XTX[1][1]-XTX[0][1]*XTX[0][1]);
B_mid=(XTX[0][1]*XTf[0]+XTX[1][1]*XTf[1])/
        (XTX[0][0]*XTX[1][1]-XTX[0][1]*XTX[0][1]);

```

```

/*Find the first and last maximas to fit the first section*/
for (i=2;i<x_zero_i[0];i++)
{
  if ((fabs(field[i])>fabs(field[i-2]))
    &&(fabs(field[i])>fabs(field[i-1]))
    &&(fabs(field[i])>fabs(field[i+1]))
    &&(fabs(field[i])>fabs(field[i+2])))
    /*take many points to reduce fails for uneven field*/
    {
      max_i_i=i;
    }
}
for (i=x_zero_i[number_of_zeros-1];i<number_elements-2;i++)
{
  if ((fabs(field[i])>fabs(field[i-2]))
    &&(fabs(field[i])>fabs(field[i-1]))
    &&(fabs(field[i])>fabs(field[i+1]))
    &&(fabs(field[i])>fabs(field[i+2])))
    /*take many points to reduce noise fails*/
    {
      max_f_i=i;
    }
}
i=max_i_i;
/*finding the max with a second order approx*/
a=((field[i+1]-field[i])
/(position[i+1]-position[i])-(field[i]-field[i-1])
/(position[i]-position[i-1]))/(position[i+1]-position[i-1]);

b=(field[i]-field[i-1])
/(position[i]-position[i-1])-a*(position[i]+position[i-1]);

max_i=-b/(2*a); /*maximum point*/
c=field[i]-a*position[i]*position[i]-b*position[i];
field_max_i=a*max_i*max_i+b*max_i+c;
df_i=(field[x_zero_i[0]+1]-field[x_zero_i[0]])
/(position[x_zero_i[0]+1]-position[x_zero_i[0]]);
l_init=2*(x_zero[0]-max_i);
/*fit a function A*cos(pi*(x-x0)/L')+B*cos(3*pi*(x-x0)/L')
to the interval from max_f_i to x_zero, where the translation
by x0 makes sure there is a maxima att the maxima of the field*/
A_init=3*field_max_i/4-df_i*l_init/(4*pi);
B_init=field_max_i/4+df_i*l_init/(4*pi);

/*finding the x at max around i=max_f_i with a second order approx*/
i=max_f_i;
a=((field[i+1]-field[i])/(position[i+1]-position[i]))

```

```

-(field [i]-field [i-1])/(position [i]-position [i-1]))
/(position [i+1]-position [i-1]);

b=(field [i]-field [i-1])
/(position [i]-position [i-1])-a*(position [i]+position [i-1]);
max_f=-b/(2*a);
c=field [i]-a*position [i]*position [i]-b*position [i];
field_max_f=a*max_f*max_f+b*max_f+c;
df_f=(field [x_zero_i[number_of_zeros-1]+1]
-field [x_zero_i[number_of_zeros-1]])
/(position [x_zero_i[number_of_zeros-1]+1]
-position [x_zero_i[number_of_zeros-1]]);

l_final=2*(max_f-x_zero [number_of_zeros-1]);
A_final=3*field_max_f/4+l_final*df_f/(4*pi);
/* Fitting derivatives and amplitudes*/
B_final=field_max_f/4-l_final*df_f/(4*pi);
field_max_f=A_final+B_final;

/*The initial and final section parametrization*/
/*choose indices that are at 80% and
50% of the distance from the peak to the end*/
i1=0.8*max_i_i;
i2=0.5*max_i_i;
p_init=log(log(field_max_i/field [i1])
/log(field_max_i/field [i2]))
/log((max_i-position [i1])/(max_i-position [i2]));
sigma_init=(max_i-position [i1])
/pow(log(field_max_i/field [i1]),1/p_init);
/*now to do the same for the last exponential*/
i1=max_f_i+0.2*(number_elements-1-max_f_i);
i2=max_f_i+0.5*(number_elements-1-max_f_i);
p_final=log(log(field_max_f/field [i1])
/log(field_max_f/field [i2]))
/log((max_f-position [i1])/(max_f-position [i2]));
sigma_final=(position [i1]-max_f)
/pow(log(field_max_f/field [i1]),1/p_final);
/*
Print parameters to the file specified as the second input
*/
if(((output_file=fopen(argv [2], "w"))==NULL)
/* if the input file cannot be open the program quits*/
{
printf("Please specify an input file.\n");
return -1;
}

```

```
}
fprintf(output_file,
"%5E %5E %5E %d \n", A_mid, B_mid, l_mid, number_of_zeros);
fprintf(output_file,
"%5E %5E %5E \n", A_init, B_init, l_init);
fprintf(output_file,
"%5E %5E %5E \n", A_final, B_final, l_final);
fprintf(output_file,
"%5E %5E %5E \n", sigma_init, p_init, position[0]);
fprintf(output_file,
"%5E %5E %5E \n", sigma_final, p_final,
position[number_elements-1]);
fclose(output_file);

/*Free memory as a last move*/
free(position);
free(field);
return 0;
}
```

Runge Kutta integration

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <gsl/gsl_sf_bessel.h>
/*
**
**
**
THIS VERSION INTEGRATES THROUGH MANY CAVITIES
**
**
**
*/
#define speed_of_light 299792458.0
#define mass 938.272046
#define charge 1.602176565e-19
#define pi 3.14159265359
#define frequency 704.42E+6
double x;
double y;
double z;
double sx;
double sy;
double sz;
double beta_s_i;
double beta_s_f;
double A_mid,B_mid,L_mid;
int nr_of_zeros;
double A_init ,B_init ,L_init ;
double A_final ,B_final ,L_final ;
double sigma_init ,p_init ,x_init ;
double sigma_final ,p_final ,x_final ;
double dx;
double omega=frequency*2*pi;
```

```

int resolution;
double x_1,x_2,x_3,x_4;
double *x_p,*y_p,*z_p,*sx_p,*sy_p,*sz_p;
double phi,beta;
int j;
int number_of_particles;

double gamma2beta(double gamma)
{
    return sqrt(1-1/pow(gamma,2));
}

double beta2gamma(double beta)
{
    return sqrt(1/(1-pow(beta,2)));
}

double gamma2energy(double gamma)
{
    return gamma*mass;
}

double energy2gamma(double energy)
{
    return energy/mass;
}
double beta_new(double beta, double dW)
{
    double gamma;
    double energy;
    gamma=beta2gamma(beta);
    energy=gamma2energy(gamma)+dW;
    gamma=energy2gamma(energy);
    return gamma2beta(gamma);
}
double beta2sz(double beta_s)
{
    return (beta*sqrt(1-beta_s*beta_s)/(beta_s*sqrt(1-beta*beta))-1);
}
double sz2beta(double beta_s)
{
    double variable;
    variable=(sz+1)*(sz+1)*beta_s*beta_s/(1-beta_s*beta_s);
    return sqrt(variable/(1+variable));
}

```

```

}

double dbeta_ds(double beta_p, double Ez)
{
    return (Ez/(beta_p*pow(beta2gamma(beta_p),3)*mass));
}

double Af_x(double x_pos)
/*Gives field value at x_pos that relates to the A coeff,
used in E_z and B_theta*/
{
    if ((x_pos<=x_1)&&(x_pos>=x_init))
    {
        return A_init*exp(-pow((x_1-x_pos)/sigma_init , p_init));
    }
    /*First cosine part, starts from 0*/
    else if ((x_pos<=x_2)&&(x_pos>x_1))
    {
        return A_init*cos(pi*(x_1-x_pos)/L_init);
    }
    /*Central part*/
    else if ((x_pos<=x_3)&&(x_pos>x_2))
    {
        return A_mid*cos(pi*(x_pos)/L_mid);
    }
    /*Last Cosine section, argument goes to zero at x_4*/
    else if ((x_pos<=x_4)&&(x_pos>x_3))
    {
        return A_final*cos(pi*(x_4-x_pos)/L_final);
    }
    /*last exponential section*/
    else if ((x_pos<x_final)&&(x_pos>x_4))
    {
        return A_final*exp(-pow((x_pos-x_4)/sigma_final , p_final));
    }
    else
    {
        return 0;
    }
}

double Bf_x(double x_pos)
/*Gives field value at x_pos that relates to the B coeff,
used in E_z and B_theta*/
{
    if ((x_pos<=x_1)&&(x_pos>=x_init))
    {
        return B_init*exp(-pow((x_1-x_pos)/sigma_init , p_init));
    }
}

```

```

}
/*First cosine part , starts from 0*/
else if ((x_pos<=x_2)&&(x_pos>x_1))
{
    return B_init*cos(3*pi*(x_1-x_pos)/L_init);
}
/*Central part*/
else if ((x_pos<=x_3)&&(x_pos>x_2))
{
    return B_mid*cos(3*pi*(x_pos)/L_mid);
}
/*Last Cosine section , argument goes to zero at x_4*/
else if ((x_pos<=x_4)&&(x_pos>x_3))
{
    return B_final*cos(3*pi*(x_4-x_pos)/L_final);
}
/*last exponential section*/
else if ((x_pos<x_final)&&(x_pos>x_4))
{
    return B_final*exp(-pow((x_pos-x_4)/sigma_final , p_final));
}
else
{
    return 0;
}
}
double Adf_x(double x_pos)
/*Gives field value at x_pos that relates to the A coeff ,
used in E_r*/
{
    if ((x_pos<=x_1)&&(x_pos>=x_init))
    {
        return A_init*exp(-pow((x_1-x_pos)/sigma_init , p_init))
            *pow((x_1-x_pos)/sigma_init , p_init -1)*p_init/sigma_init;;
    }
    /*First cosine part , starts from 0*/
    else if ((x_pos<=x_2)&&(x_pos>x_1))
    {
        return pi*A_init*sin(pi*(x_1-x_pos)/L_init)/L_init;
    }
    /*Central part*/
    else if ((x_pos<=x_3)&&(x_pos>x_2))
    {
        return -pi*A_mid*sin(pi*(x_pos)/L_mid)/L_mid;
    }
    /*Last Cosine section , argument goes to zero at x_4*/
    else if ((x_pos<=x_4)&&(x_pos>x_3))

```

```

    {
        return pi*A_final*sin(pi*(x_4-x_pos)/L_final)/L_final;
    }
    /*last exponential section*/
    else if ((x_pos<x_final)&&(x_pos>x_4))
    {
        return -A_final*exp(-pow((x_pos-x_4)/sigma_final , p_final))
            *pow((x_pos-x_4)/sigma_final , p_final-1)*p_final/sigma_final;
    }
    else
    {
        return 0;
    }
}
double Bdf_x(double x_pos)
/*Gives field value at x_pos that relates to the B coeff
, used in E_r*/
{
    if ((x_pos<=x_1)&&(x_pos>=x_init))
    {
        return B_init*exp(-pow((x_1-x_pos)/sigma_init , p_init))
            *pow((x_1-x_pos)/sigma_init , p_init-1)*p_init/sigma_init;;
    }
    /*First cosine part, starts from 0*/
    else if ((x_pos<=x_2)&&(x_pos>x_1))
    {
        return 3*pi*B_init*sin(3*pi*(x_1-x_pos)/L_init)/L_init;
    }
    /*Central part*/
    else if ((x_pos<=x_3)&&(x_pos>x_2))
    {
        return -3*pi*B_mid*sin(3*pi*(x_pos)/L_mid)/L_mid;
    }
    /*Last Cosine section, argument goes to zero at x_4*/
    else if ((x_pos<=x_4)&&(x_pos>x_3))
    {
        return 3*pi*B_final*sin(3*pi*(x_4-x_pos)/L_final)/L_final;
    }
    /*last exponential section*/
    else if ((x_pos<x_final)&&(x_pos>x_4))
    {
        return -B_final*exp(-pow((x_pos-x_4)/sigma_final , p_final))
            *pow((x_pos-x_4)/sigma_final , p_final-1)*p_final/sigma_final;
    }
    else
    {
        return 0;
    }
}

```

```

}
}

```

```

int integration(double phi_s_i, double phi_s_f
, double beta_s_i, double beta_s_f)
/*This function simulates all particles ,
with the synchronous particle as reference
through one cavity*/
{
    double r;
    double Ez, Er, Bo, Ez_s;
    double Af1, Bf1, Adf1, Bdf1, Af2, Bf2, Adf2, Bdf2;
    /*field values needed for RK2 at f(x(t), t) and f(x+df/2, t+dt/2)*/
    int i, k, j;
    double x_pos=x_init; /*set the starting point*/
    double dbeta_1, beta0;
    double dsr; /*radial impulse*/
    double domega, dphi_s;
    /*omega/(c*beta) can be interpreted as spatial frequency*/
    double k_1_m, k_3_m, k_1_f, k_3_f;
    double k_1, k_3;
    double beta_s, phi_s, beta_s2, phi_s2;

    k_1=sqrt(pow(pi/L_init,2)-pow(2*pi*frequency/speed_of_light,2));
    k_3=sqrt(pow(3*pi/L_init,2)-pow(2*pi*frequency/speed_of_light,2));
    k_1_m=sqrt(pow(pi/L_mid,2)-pow(2*pi*frequency/speed_of_light,2));
    k_3_m=sqrt(pow(3*pi/L_mid,2)-pow(2*pi*frequency/speed_of_light,2));
    k_1_f=sqrt(pow(pi/L_final,2)-pow(2*pi*frequency/speed_of_light,2));
    k_3_f=sqrt(pow(3*pi/L_final,2)-pow(2*pi*frequency/speed_of_light,2));

    beta_s=beta_s_i;
    phi_s=phi_s_i;
    j=0;
    for (i=0; i<resolution; i++)
    {
        if (x_pos>x_2)
        {
            k_1=k_1_m;
            k_3=k_3_m;
        }
        if (x_pos>x_3)
        {
            k_1=k_1_f;
            k_3=k_3_f;
        }
    }
}

```

```

Af1=Af_x(x_pos);
Bf1=Bf_x(x_pos);
Af2=Af_x(x_pos+dx/2);
Bf2=Bf_x(x_pos+dx/2);
Adf2=Adf_x(x_pos+dx/2);
Bdf2=Bdf_x(x_pos+dx/2);

Ez=(Af1+Bf1)*sin(phi_s); /*field value at x_pos, phi*/
dbeta_1=dbeta_ds(beta_s,Ez); /*f(x(t),t)*/
Ez=(Af2+Bf2)
*sin(phi_s+dx*omega/(2*(beta_s+dx*dbeta_1/2)*speed_of_light));
dbeta_1=dbeta_ds(beta_s+dx*dbeta_1/2,Ez); /*f(x+df/2,t+dt/2)*/
dphi_s=omega/((beta_s+dbeta_1*dx/2)*speed_of_light);
beta_s2=beta_s+dx*dbeta_1;
phi_s2=phi_s+dx*dphi_s;

for(k=0;k<number_of_particles;k++)
{
  x=x_p[k]-z_p[k]*sx_p[k]; /*finding x-pos at x_pos*/
  y=y_p[k]-z_p[k]*sy_p[k]; /*finding y-pos at x_pos*/
  z=z_p[k];
  sx=sx_p[k];
  sy=sy_p[k];
  sz=sz_p[k];
  beta=sz*beta(beta_s);
  /*Calculates the initial velocity of the particle*/
  phi=phi_s-z*omega*sqrt(1+sx*sx+sy*sy)
  /(beta*speed_of_light);
  /*a particle with position z relative
  to the synchronous enters with a phase diff of phi*/
  r=sqrt(x*x+y*y); /*the radial position*/

  Ez=(Af1*gsl_sf_bessel_J0(k_1*r)
  +Bf1*gsl_sf_bessel_J0(k_3*r))*sin(phi);
  /*E_z-Field at ((r,x,t)*/
  dbeta_1=dbeta_ds(beta,Ez);
  domega=dx*sqrt(1+sx*sx+sy*sy)
  *omega/(2*(beta+dx*dbeta_1/2)*speed_of_light);
  /*Fields at ((r,x,t)+d(r,x,t)/2)*/
  Ez=(Af2*gsl_sf_bessel_J0(k_1*r)
  +Bf2*gsl_sf_bessel_J0(k_3*r))*sin(phi+domega);
  Er=-(Adf2*gsl_sf_bessel_J1(k_1*r)/k_1
  +Bdf2*gsl_sf_bessel_J1(k_3*r)/k_3)*sin(phi+domega);
  Bo=(Af2*gsl_sf_bessel_J1(k_1*r)/k_1
  +Bf2*gsl_sf_bessel_J1(k_3*r)/k_3)*cos(phi+domega);

  dsr=(Er-omega*beta*Bo/speed_of_light)

```

```

        *dx/(mass*beta2gamma(beta)*beta*beta);
        /*Radial impulse*/
    dbeta_1=dbeta_ds(beta+dbeta_1*dx/2,Ez);
        /*velocity increase*/

    phi+=dx*sqrt(1+sx*sx+sy*sy)*omega
        /((beta+dbeta_1*dx/2)*speed_of_light);
        /*phase increase*/

    beta0=beta;
    beta=beta+dx*dbeta_1;
        /*increasing velocity*/

    x+=sx*dx; /*increasing the transverse position*/
    y+=sy*dx;
    r=sqrt(x*x+y*y);
    if (r!=0)
    {
        sx=sx*beta0/beta+dsr*x/r;
        sy=sy*beta0/beta+dsr*y/r;
    }
    z=-beta*speed_of_light*(phi-phi_s2)
        /(omega*sqrt(1+sx*sx+sy*sy));
        /*recalculating z by the phi_s at the end*/
    sz=beta2sz(beta_s2);
        /*recalculating z by the beta_s at the end*/
    x_p[k]=x+z*sx;
    y_p[k]=y+z*sy;
    z_p[k]=z;
    sx_p[k]=sx;
    sy_p[k]=sy;
    sz_p[k]=sz;

}
j++;
x_pos+=dx;
beta_s=beta_s2;
phi_s=phi_s2;
}
return 0;

}
int main(int argc, char **argv)
{
    FILE *input_file; /* this is the file name */
    FILE *input_particles;
    FILE *output_file;
    FILE *input_cavities;

```

```
double beta_s_i;
double phi_max;
double phi_s_f;
double phi_s_i;
int i;
int k;
double Ez,Er,Bo;
int number_of_cavities;
char buffer[1024]; /* this is a dummy buffer */
double r;
double mean_x,mean_y,mean_z;
double var_x,var_y,var_z;

if ((input_file=fopen(argv[1], "r"))==NULL)
    /* if the input file cannot be open the program quits*/
{
    printf("Please specify an input file.\n");
    return -1;
}

if (fscanf(input_file,
"%lf %lf %lf %d", &A_mid, &B_mid, &L_mid, &nr_of_zeros)==EOF)
{
    printf("Something is going wrong.\n");
    return -1;
}
if (fscanf(input_file,
"%lf %lf %lf", &A_init, &B_init, &L_init)==EOF)
{
    printf("Something is going wrong.\n");
    return -1;
}
if (fscanf(input_file,
"%lf %lf %lf", &A_final, &B_final, &L_final)==EOF)
{
    printf("Something is going wrong.\n");
    return -1;
}
if (fscanf(input_file,
"%lf %lf %lf", &sigma_init, &p_init, &x_init)==EOF)
{
    printf("Something is going wrong.\n");
    return -1;
}
if (fscanf(input_file,
"%lf %lf %lf", &sigma_final, &p_final, &x_final)==EOF)
{
```

```

    printf("Something is going wrong.\n");
    return -1;
}

fclose(input_file);

printf("Give resolution: ");
scanf("%d",&resolution);
printf("\n");
dx=(x_final-x_init)/resolution;

/*These variables mark which sections of the field*/

x_1=-L_mid*(nr_of_zeros-1)/2-L_init/2;
x_2=-L_mid*(nr_of_zeros-1)/2;
x_3=L_mid*(nr_of_zeros-1)/2;
x_4=L_mid*(nr_of_zeros-1)/2+L_final/2;

/*reading in nr of particles*/

if ((input_particles=fopen(argv[2], "r"))==NULL)
{
    printf("Please specify an input file.\n");
    return -1;
}
number_of_particles=0;

while(fgets(buffer, 1024, input_particles) != NULL)
{
    number_of_particles++;
}
rewind(input_particles);
x_p=(double *) malloc(sizeof(double)*number_of_particles);
y_p=(double *) malloc(sizeof(double)*number_of_particles);
z_p=(double *) malloc(sizeof(double)*number_of_particles);
sx_p=(double *) malloc(sizeof(double)*number_of_particles);
sy_p=(double *) malloc(sizeof(double)*number_of_particles);
sz_p=(double *) malloc(sizeof(double)*number_of_particles);
for (i=0;i<number_of_particles;i++)
{
    if (fscanf(input_particles, "%lf %lf %lf %lf %lf %lf",
        &x_p[i], &y_p[i], &z_p[i], &sx_p[i], &sy_p[i], &sz_p[i])==EOF)
        {

```

```

        printf("Something is going wrong.\n");
        return -1;
    }
}

if ((output_file=fopen(argv[3], "w"))==NULL)
{
    printf("Please specify an input file.\n");
    return -1;
}

if ((input_cavities=fopen(argv[4], "r"))==NULL)
{
    printf("Please specify an input file.\n");
    return -1;
}
number_of_cavities=0;

while(fgets(buffer, 1024, input_cavities) != NULL)
{
    number_of_cavities++;
}
rewind(input_cavities);

printf("number of cavities=%d \n", number_of_cavities);
j=0;

for (k=0;k<number_of_cavities;k++)
{
    if (fscanf(input_cavities, "%lf %lf %lf %lf"
, &beta_s_i, &beta_s_f, &phi_s_i, &phi_s_f)==EOF)
    {
        printf("Something is going wrong.\n");
        return -1;
    }
    /*integrates all particles through one cavity at a time*/
    printf("Cavity nr:%d\n", k);
    integration(phi_s_i, phi_s_f, beta_s_i, beta_s_f);
}
for (i=0;i<number_of_particles;i++)
{
    fprintf(output_file,
        "%.15E %.15E %.15E %.15E %.15E %.15E \n",
        x_p[i], y_p[i], z_p[i], sx_p[i], sy_p[i], sz_p[i]);
}
}

```

```
free(x_p);
free(y_p);
free(z_p);
free(sx_p);
free(sy_p);
free(sz_p);
fclose(input_particles);
fclose(output_file);
fclose(input_cavities);

return 0;
}
```

List of Figures

1.1	The picture show an artist rendition of the completed ESS research facility.	2
1.2	The figure describes the layout of the linac and gives the energies between sections. The red sections are in room temperature. The blue sections are at a temperature of only a few kelvin, they hold superconducting components. LEBT is the low energy beam transport, MEBT is the Medium EBT and finally the High EBT followed by room for potential upgrades [1].	3
3.1	Plotted field data of the electric field on the z -axis, as a function of distance in a high- β cavity. The field data is taken from measurements of the manufacturer at Soleil in France.	10
3.2	The figure shows the field amplitude in MV/m, of the electric field in the z -direction at a phase corresponding to a maximal amplitude. The field is that of a high- β cavity. The figure shows how the field is subdivided into five sections, from z_1 to z_5 . In each of these sections a different set of parameters will be used to describe the shape of the field.	12
3.3	The parametrization is shown in red while the field that it is derived from is shown in blue. The second plot show the difference between the parametrization and the actual field data, normalized over the maximum amplitude of the field. This indicate to what degree the expansion in only two cosine terms can approximate the field. The largest error is present in the sections z_1 and z_5	14
3.4	A quiver plot of the E_z -field and a potential plot, both derived from the parametrization. This is only displayed for the middle cosine section, z_3 in Figure 3.2. For a superconducting cavity the arrows in the quiver-plot should be perpendicular to the cavity walls. This also means that for a field in a superconductor the outline of the cavity walls seen as a potential line. The plot gives an indication of where the walls are.	15
3.5	A plot showing the E_z and E_r -field. The E_r -field is shown at a radius $r = 0.01$ since E_r is 0 at $r = 0$	16
3.6	The field amplitude of E_z on the central axis, inside a spoke cavity.	18

3.7	The top figure show how close the parametrization replicate the spoke field. The bottom figure show the relative error between the spoke field data and the parametrization, it is normalized over the maximum field amplitude.	19
4.1	The energy gained by a particle as a function of its phase relative to the electric field when entering the cavity.	22
6.1	Comparison of the two methods implemented for particle tracking. The blue line marks the Runge Kutta and the red line marks the semi-analytical method. The second order Runge Kutta comes out as superior when looking at time it takes to run the simulation.	32
6.2	The two plots on the left show the output from both the matrix model and the particle tracking model. The output is hard to distinguish in the top left figure but a deviation in the x' -coordinate is visible between the methods at growing radius in the bottom left plot. The radius of the iris of the cavity is roughly 6 cm, meaning that particles further away than that will hit the walls. The two plots on the right show the difference between the methods. A difference in both position and divergence is clear, and the difference appear to be linear.	34
6.3	These two plots show how the position in z and relative momentum z' as a function of the distance the particle holds from the axis when passing through a cavity. It shows that particles further from the center will lose energy relative to the particles near the center.	35
6.4	The two figures on the left show the output of a uniform distribution on the z -line, in the top left plot the output position is plotted against input position. The bottom left plot show the relative momentum at the cavity exit against initial position. The plots on the right show, at the top the difference in z , and at the bottom the difference in z' , between the linear and non-linear model.	36
6.5	Comparison between the same code in JAVA and in C. The results show that the code is faster in C.	37
6.6	Figure displaying the distribution of accepted particles.	38
6.7	Figure displaying the distribution of accepted particles at the end of the high beta section.	38



LUND
UNIVERSITY

<http://www.eit.lth.se>