Michael Eder, BSc

Three-dimensional geometric integrator for charged particle orbits in toroidal fusion devices

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Technical Physics

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn. Winfried Kernbichler

Institute of Theoretical and Computational Physics

Co-Supervisor:
Dipl.-Ing. Dr.rer.nat. Christopher Albert, BSc

Graz, July 2018

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

In this master's thesis, a three-dimensional integrator for guiding center orbits of charged particles in toroidal fusion devices is described. The integrator is intrinsically designed to preserve the total energy and the perpendicular adiabatic invariant as constants of motion up to computer accuracy, and thus belongs to the class of geometric integrators. Furthermore, also the phase space volume is conserved.
The purpose of that integrator is to be used in Monte Carlo procedures to simulate the particle distribution function where a box counting method (within spacial cells) is used for the calculation of macroscopic plasma parameters. Such a computation is needed for the evaluation of plasma response currents as well as charges caused by external non-axisymmetric electromagnetic perturbations in tokamaks.
At the cost of orbit accuracy and exactness of time evolution, the integrator possesses about the same computation speed as a conventional high order adaptive ODE integrator, while already providing the particle's coordinates and velocities at the boundaries of spacial cells which is needed for the evaluation of the distribution function. Thus, it is more efficient than a direct solution of the equations of motion with a high order adaptive ODE integrator, where these quantities at the cell-boundaries have to be computed additionally. Moreover, it is less sensitive to inaccurate representation of the electromagnetic field, due to numerical inaccuracies resulting from statistical noise in the data.

Kurzfassung

In dieser Masterarbeit wird ein dreidimensionaler Integrator für die Gyrationszentrumstrajektorien von geladenen Teilchen in toroidalen Fusionskammern präsentiert. Der Integrator ist intrinsisch dafür entworfen, die Gesamtenergie, das magnetische Moment der geladenen Teilchen und das Phasenraum-Volumen bis auf Computer-Genauigkeit zu erhalten. Auf Grund dieser Eigenschaft gehört er zur Gruppe der geometrischen Integratoren.

Das Einsatzgebiet dieses Integrators sind Monte-Carlo-Simulationen zur Berechnung der Teilchen-Verteilungsfunktion. Diesbezüglich ist es notwendig, Teilchen-Koordinaten und -Geschwindigkeiten an den Rändern von dreidimensionalen Raumzellen zu berechnen. Mit Hilfe der Teilchen-Verteilungsfunktion können makroskopische Plasma-Parameter ermittelt werden. Im Speziellen wird eine derartige Simulation für die Berechnung von Plasma-Response-Strömen und -Ladungen auf externe, nichtaxisymmetrische, elektromagnetische Störungen in Tokamaks durchgeführt.

Auf Kosten der Trajektoriengenauigkeit und der Exaktheit der Zeitentwicklung besitzt dieser Integrator ungefähr die gleiche Rechengeschwindigkeit wie gewöhnliche adaptive ODE-Integratoren höherer Ordnung, wobei im Gegensatz zu diesen Teilchen-Koordinaten und -Geschwindigkeiten bereits an den Rändern von dreidimensionalen Raumzellen berechnet werden. Dadurch wird eine Ersparnis an Rechenleistung erreicht, die bei gewöhnlichen adaptiven ODE-Integratoren zusätzlich anfällt. Außerdem ist der Integrator weniger sensitiv für ungenaue Darstellung des elektromagnetischen Feldes auf Grund von numerischen Ungenauigkeiten, die von statistischem Rauschen in den zu Grunde liegenden Daten stammen.

Acknowledgements

My sincere gratitude goes to my supervisor Dr. Winfried Kernbichler who supported me with great ideas and suggestions. I am especially thankful for his advice while at the same time giving me the freedom to develop my own ideas.

I would like to express my gratitude to my co-supervisors Dr. Sergei Kasilov and Dr. Christopher Albert for making themselves available whenever I needed help. In particular, I am grateful for their competent answers to my countless questions as well as for their patience and motivation. I deeply appreciate their ongoing encouragement and support.

I especially would like to thank Dr. Sergei Kasilov for his great hospitality in Kharkov where I visited the Institute of Plasma Physics of the NSC KIPT and where I was able to collaborate with him. The joint research was supported by the OEAD within the framework of "Scientific and Technological Cooperation with Ukraine 2017-18, Project No. UA 04/2017". The results obtained from that collaboration had a fruitful influence on this thesis.

Finally, I would like to thank my parents, my sister and friends with all my heart for their continuous support and encouragement and most importantly Jenny for all her love and patience.

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under Grant Agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Contents

Chapter 1

Introduction

1.1 Background

In plasma physics, in particular in controlled fusion physics, kinetic theory is used to explain macroscopic properties of the plasma. Each particle of a plasma can be represented as a point in six-dimensional phase space ($\mathbf{r}$,$\mathbf{v}$). Due to the fact that the physical problem with an exact microscopic description of a plasma, consisting of a high number of interacting particles, is intractable, the microscopic theory was developed further to kinetic theory. In kinetic theory one applies statistical concepts and further sophisticated approximations in order to calculate macroscopic properties. The key quantity in kinetic theory is the particle distribution function f_α [1] which can be implicitly defined by the number of particles of kind α per unit volume in phase space near the point $z = (\mathbf{r}, \mathbf{v})$ at the time t

$$f_\alpha(\mathbf{r}, \mathbf{v}, t)\, \mathrm{d}^3 r \mathrm{d}^3 v. \tag{1.1}$$

Following [2], an evaluation of the distribution function and/or its moments by direct modelling of particle orbits in Monte Carlo simulations is widely used in plasma physics (see e.g. codes like EUTERPE [3, 4] or ASCOT [5]).
A key issue in such codes is an efficient algorithm for the calculation of trajectories (orbits) of non-interacting charged particles in complex (quasi-)stationary magnetic and electric fields. The reason for that is the high number of test particle orbits which is required to minimize the statistical error of such calculations, that scales inversely with the square root of the number of test particles.
This issue is especially important in global transport modelling (see e.g. [6]) where the profiles of plasma parameters are calculated self-consistently from test particle trajectories which have to be traced over the profile relaxation (confinement) time. In particular, such a computation is needed for the evaluation of plasma response currents as well as charges caused by external non-axisymmetric electromagnetic perturbations in tokamaks [7].

Within transport modelling, computation of stochastic test particle orbits [8] requires the solution of guiding center equations [9, 10], which is usually performed with the help of conventional adaptive high order ODE integrators.

Such adaptive high order ODE integrators have two big disadvantages: First, the particle's coordinates and velocities are naturally calculated at accuracy-adapted integration steps. In order to calculate the distribution function, these quantities are needed at the boundaries of defined cells, which involves further computational effort. Second, high order spline-interpolation of the field quantities can lead to unwanted oscillations in case of an inaccurate representation of the electromagnetic field (noise in the data). Further, these high oscillations in the field quantities can destroy the physical properties of the particle orbit.

For axisymmetric fusion devices an algorithm which is roughly an order of magnitude more efficient than a standard high order adaptive ODE integrator and which is less sensitive to noise in the field quantities was already developed [2]. However, this particular code is only a two-dimensional integrator for charged particle orbits. If one is interested in the particle motion inside an axisymmetric device with weak non-axisymmetric perturbations (e.g. tokamak with ELM-mitigation-field), an algorithm for a three-dimensional integrator must be developed.

Thus, one can formulate the following requirements for a three-dimensional integrator for charged particle orbits in toroidal fusion devices:

1. Particle orbit-coordinates and velocities must be computed at the boundaries of spacial three-dimensional cells.
2. The algorithm should be less sensitive to the accuracy of the representation of the electromagnetic field (noise in the data).
3. Computational effort should be less than in a conventional adaptive ODE integrators.

1.2 Overview

In chapter 2 such an integrator which satisfies the requirements from above is derived. In order to reduce computational effort, one neglects the exactness of time evolution and approximates field quantities with linear functions, which diminishes also the sensitivity to noise in the representation of the electromagnetic field. The obtained

integrator belongs to the class of geometric integrators, therefore it is called *3DGeoInt*.

In order to evaluate the distribution function by direct modelling of particle orbits, coordinates and velocities must be computed at the boundaries of spacial cells. Chapter 3 presents a simple method how space can be split into cells (tetrahedrons).

To calculate the particle orbit in a cell, the linearized field quantities for each and every cell must be provided. In chapter 4 is described how these quantities are computed and how the tetrahedron-specific algorithm-constants of the linearized set of equations of motion are calculated.

Chapter 5 is the main "numerical" part in this thesis. An algorithm for a computationally inexpensive calculation of the particle orbit inside a cell is developed. From given entry coordinates and velocities, the particle's exit coordinates and velocities at the respective point are computed by the Runge-Kutta method, applied in an iteration procedure.

The calculated particle orbits are presented in chapter 6 in a qualitative manner. An overview of the orbits of passing particles and trapped particles is given. Additionally, the bounce time and bounce frequency are computed. Finally, orbits of axisymmetric toroidal fusion devices with weak non-axisymmetric perturbation are calculated and qualitatively analyzed.

In order to assess the algorithm's accuracy and computation speed, several tests are executed in chapter 7. For that purpose *3DGeoInt* is compared to a conventional adaptive ODE integrator. Additionally, also the effect of noise in the field quantities is tested. Furthermore, the conservation of the canonical toroidal angular momentum is examined.

In Appendix A the linearized set of equations which is derived in chapter 2 is mathematically analyzed. The obtained results can further be used to optimize *3DGeoInt*.

Chapter 2

Derivation of the geometric integrator

2.1 Definition of a geometric integrator

Following [11], geometric integration is the numerical integration of a differential equation, while preserving one or more of its "geometric" properties exactly within computer accuracy. Many of these geometric properties are of crucial importance in physical applications: preservation of energy, momentum, angular momentum, etc. This definition is rather loose, whereas a subclass of geometric integrators is the class of symplectic integrators, which have a more restricted definition. Symplectic integrators [12, 13] possess, as a conserved quantity, a (slightly perturbed) Hamiltonian.
The particle orbits that are calculated with a geometric integrator do not necessarily satisfy Hamilton's equations of motion, as in the case of orbits computed with a symplectic integrator.

In order to obtain a particle distribution function in six-dimensional phase space, we are interested in the particle's coordinates and velocities at the boundaries of a cell at a certain time, as described in chapter 1. Hence, we must split the space into small cells.
In each cell we approximate the exact equations of motion by a set of linear ordinary differential equations, in order to be less sensitive to noise in the field quantities and to save computation power. Additionally, we will also approximate the exact time evolution by keeping the orbit's exact geometry to reduce computational effort.
This linear set has two exact properties: it conserves the total energy and the magnetic moment, thus the integrator belongs to the class of geometric integrators. Furthermore it is ensured, that the orbits are divergence free in coordinate space.

2.2 Guiding center equations in vector form and in curvilinear variables

The equations of guiding center motion with w and $J_\perp$, the total energy and the perpendicular adiabatic invariant (both invariants of motion), can be written in a rather simple symbolic equation, as derived in [10]:
The divergence free (Hamiltonian) form of drift orbits can be described by the lines of force of the effective Morozov-Solov'ev field [9],

$$\mathbf{B}^* = \nabla \times \mathbf{A}^*, \tag{2.1}$$

where the effective vector potential is expressed through the magnetic field parameters and the invariants of motion w (total energy) and $J_\perp$ (perpendicular adiabatic invariant) as following

$$\mathbf{A}^* = \mathbf{A} + \frac{v_\parallel}{\omega_c}\mathbf{B}, \qquad v_\parallel = \sigma\left(\frac{2}{m_\alpha}\left(w - J_\perp\omega_c - e_\alpha\Phi\right)\right)^{1/2}, \tag{2.2}$$

where $\mathbf{A}$, $\mathbf{B}$, $\omega_c = e_\alpha B/(m_\alpha c)$ are the usual vector potential, the magnetic field strength and the cyclotron frequency, respectively. Φ is the electrostatic potential, and σ, e_α, m_α and c are parallel velocity sign, particle charge and mass and the speed of light, respectively. Note that $\mathbf{A}^* = c\mathbf{P}/e_\alpha$ where $\mathbf{P}$ is the canonical momentum in guiding center approximation.

The big advantage of this form is, that $v_\parallel$ is a function of coordinates, already treating the invariants of motion $w = m_\alpha v^2/2 + e_\alpha\Phi$ and $J_\perp = m_\alpha v_\perp^2/(2\omega_c)$ as constants. Hence, derivatives in (2.1) are computed treating $v_\parallel$ as function of coordinates given by (2.2).

The equations of guiding center motion, using the the effective Morozov-Solov'ev field, are

$$\dot{\mathbf{r}} = v_\parallel\frac{\mathbf{B}^*}{B_\parallel^*}, \qquad B_\parallel^* = \mathbf{h}\cdot\mathbf{B}^*, \tag{2.3}$$

with $\mathbf{h} = \mathbf{B}/B$ being the unit vector along the magnetic field.
In curvilinear variables x^i equations of motion (2.3) are

$$\dot{x}^i = \frac{v_\parallel\varepsilon^{ijk}}{\sqrt{g}B_\parallel^*}\frac{\partial A_k^*}{\partial x^j}, \tag{2.4}$$

where g is a metric determinant and covariant components of effective vector potential A_k^* are expressed via covariant components of vector potential A_k and of the magnetic field B_k as follows

$$A_k^* = A_k + \frac{v_\parallel}{\omega_c} B_k. \tag{2.5}$$

Introducing the notation

$$\frac{v_\parallel^2}{2} = U = \frac{1}{m}\left(w - J_\perp \omega_c - e_\alpha \Phi\right) \tag{2.6}$$

in order to avoid difficulties with the sign of $v_\parallel$, we can write the equation of motion for $v_\parallel$ as

$$\dot{v}_\parallel = \frac{1}{v_\parallel} \dot{x}^i \frac{\partial U}{\partial x^i} \tag{2.7}$$

where $U = U(\mathbf{x})$ and $\dot{x}^i$ are given by (2.4). Treating $v_\parallel$ as an independent variable guiding center equations become a set of four equations

$$\begin{aligned} \frac{\mathrm{d}x^i}{\mathrm{d}t} &= \frac{1}{B_\parallel^* \sqrt{g}} \cdot \varepsilon^{ijk} \left(v_\parallel \frac{\partial A_k}{\partial x^j} + 2U \frac{\partial}{\partial x^j} \frac{B_k}{\omega_c} + \frac{B_k}{\omega_c} \frac{\partial U}{\partial x^j} \right), \\ \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} &= \frac{1}{B_\parallel^* \sqrt{g}} \cdot \varepsilon^{ijk} \frac{\partial U}{\partial x^i} \left(\frac{\partial A_k}{\partial x^j} + v_\parallel \frac{\partial}{\partial x^j} \frac{B_k}{\omega_c} \right). \end{aligned} \tag{2.8}$$

2.3 Geometric integrator: Neglect exactness of time evolution

We can formally introduce the orbit parameter τ, related to time by $\mathrm{d}t = B_\parallel^* \sqrt{g} \mathrm{d}\tau$, and the set of guiding center equations becomes

$$\begin{aligned} B_\parallel^* \sqrt{g} \dot{x}^i = \frac{\mathrm{d}x^i}{\mathrm{d}\tau} &= \varepsilon^{ijk} \left(v_\parallel \frac{\partial A_k}{\partial x^j} + 2U \frac{\partial}{\partial x^j} \frac{B_k}{\omega_c} + \frac{B_k}{\omega_c} \frac{\partial U}{\partial x^j} \right), \\ B_\parallel^* \sqrt{g} \dot{v}_\parallel = \frac{\mathrm{d}v_\parallel}{\mathrm{d}\tau} &= \varepsilon^{ijk} \frac{\partial U}{\partial x^i} \left(\frac{\partial A_k}{\partial x^j} + v_\parallel \frac{\partial}{\partial x^j} \frac{B_k}{\omega_c} \right). \end{aligned} \tag{2.9}$$

If τ is used as an independent variable of the ODE set, time evolution is obtained in the implicit form integrating the above equation for $t(\tau)$ with $B_\parallel^* \sqrt{g}$ being a known function of τ.

As it was described in chapter 1, a major goal of *3DGeoInt* is computational efficiency. Thus, one has to consider to make certain approximations at the cost of exactness. In order to reduce this exact equation set to a linear set we notice that the common factor $B_\parallel^* \sqrt{g}$ does not influence the geometry of the orbits, but only affects time evolution which is not needed very accurately. Therefore this factor can be set to a

constant within a cell, what makes implicit time dependence explicit.

Note, that due to this approximation, the exactness of time evolution is neglected and therefore, the particle motion doesn't satisfy Hamilton's equations of motion anymore. If one is interested in a symplectic integrator, that satisfies Hamilton's equation of motion, the implicitly time dependent differential equation $\mathrm{d}t = B_{\parallel}^{*}\sqrt{g}\mathrm{d}\tau$ has to be solved.

2.4 Approximate field quantities by linear functions

2.4.1 Introduction and effects

In order to reduce the integrator's sensitivity to noise in the field quantities, one can approximate the field quantities with linear functions in a cell, instead of a 5^{th} order spline interpolation, as it is done for convential adaptive ODE integrators [22, 23, 24]. A second benefit due to the approximation with linear functions occurs at the computation speed. Obviously a linear interpolation takes less computational effort than a 5^{th} order spline interpolation.

Hence, the field quantities will be approximated by piecewise linear functions with discontinuous first derivatives. The particle's motion in an axisymmetric device can be described by one single equation, due to the reduction of dimensionality. For such axisymmetric systems, discontinuous first derivatives in the field quantities will not have an effect on the physical properties. Briefly speaking, particle orbits will stay closed in the poloidal plane. It is assumed, that for our purpose the effects of discontinuous first derivatives in the field quantities in axisymmetric systems with weak non-axisymmetric perturbations, are small enough.

2.4.2 Linearized equation set in standard form

Approximating the quantities A_k, B_k/ω_c, Φ and ω_c by linear functions, equation (2.9) becomes a set of four linear equations. Note that quantities A_k, B_k/ω_c and ω_c are linearly interpolated independently from each other. Although they are linked with each other, mutual relations contain the metric tensor, so one can assume that this tensor is modified in such a way that those relations hold.
Denoting the extended set of variables with z^i where $z^i = x^i$ for $i = 1, 2, 3$ and $z^4 = v_{\parallel}$,

the linearized equation set (2.9) takes a standard form

$$\frac{\mathrm{d}z^i}{\mathrm{d}\tau} = a^i_l z^l + b^i, \tag{2.10}$$

where for $i, l = 1, 2, 3$ matrix elements are

$$\begin{aligned}
a^i_l &= \varepsilon^{ijk}\left(2\frac{\partial U}{\partial x^l}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^l}\frac{B_k}{\omega_c}\right) \qquad \text{for} \qquad 1 \le i, l \le 3, \\
a^i_4 &= \varepsilon^{ijk}\frac{\partial A_k}{\partial x^j} \qquad \text{for} \qquad 1 \le i \le 3, \\
a^4_l &= 0 \qquad \text{for} \qquad 1 \le l \le 3, \\
a^4_4 &= \varepsilon^{ijk}\frac{\partial U}{\partial x^i}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c},
\end{aligned} \tag{2.11}$$

and components of vector b^i are

$$\begin{aligned}
b^i &= \varepsilon^{ijk}\left(2U_0\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \left(\frac{B_k}{\omega_c}\right)_0\frac{\partial U}{\partial x^j}\right) \qquad \text{for} \qquad 1 \le i \le 3, \\
b^4 &= \varepsilon^{ijk}\frac{\partial U}{\partial x^i}\frac{\partial A_k}{\partial x^j}.
\end{aligned} \tag{2.12}$$

Here, quantities denoted with subscript zero mean the value at the origin of the coordinates,

$$U = U_0 + x^i\frac{\partial U}{\partial x^i}, \qquad \frac{B_k}{\omega_c} = \left(\frac{B_k}{\omega_c}\right)_0 + x^i\frac{\partial}{\partial x^i}\frac{B_k}{\omega_c}. \tag{2.13}$$

Furthermore it is possible to express a^4_4 as trace of a^i_l for $1 \le i, l \le 3$.
The diagonal elements of a^i_l are

$$\begin{aligned}
a^1_1 &= \varepsilon^{1jk}\left(2\frac{\partial U}{\partial x^1}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^1}\frac{B_k}{\omega_c}\right), \\
a^2_2 &= \varepsilon^{2jk}\left(2\frac{\partial U}{\partial x^2}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^2}\frac{B_k}{\omega_c}\right), \\
a^3_3 &= \varepsilon^{3jk}\left(2\frac{\partial U}{\partial x^3}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^3}\frac{B_k}{\omega_c}\right).
\end{aligned} \tag{2.14}$$

The trace of a^i_l is the sum over a^1_1, a^2_2 and a^3_3 and can be written as an implicit sum over i

$$\mathrm{tr}\{a^i_l\} = a^1_1 + a^2_2 + a^3_3 = \varepsilon^{ijk}(2\frac{\partial U}{\partial x^i}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \underbrace{\frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^i}\frac{B_k}{\omega_c}}_{\star}). \tag{2.15}$$

By using the anticyclic behaviour of the Levi-Civita symbol ($\varepsilon^{ijk} = -\varepsilon^{jik}$), the term

marked by $\star$ can be rewritten as

$$\varepsilon^{ijk}\frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^i}\frac{B_k}{\omega_c} = -\varepsilon^{jik}\frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^i}\frac{B_k}{\omega_c}. \tag{2.16}$$

Interchanging the names of i and j in the $\star$-term ($i \leftrightarrow j$), leads to the expression

$$\begin{aligned} \mathrm{tr}\{a_l^i\} &= \varepsilon^{ijk}\left(2\frac{\partial U}{\partial x^i}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} - \frac{\partial U}{\partial x^i}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c}\right) \\ &= \varepsilon^{ijk}\left(\frac{\partial U}{\partial x^i}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c}\right) \\ &= a_4^4 \qquad \text{for} \qquad 1 \leq i, l \leq 3. \end{aligned} \tag{2.17}$$

Chapter 3

Generating the mesh: Split space into tetrahedrons

For a first examination of the working principle of *3DGeoInt* a rather simple mesh is used.
Since *3DGeoInt* should calculate particle orbits for toroidal fusion devices, a torus-like space (hollow cylinder) has to be split into finite cells. For splitting a three-dimensional space into three-dimensional cells, tetrahedrons have been proved to be very practical. [21]

First of all, a hollow cylinder is split into hexahedrons, as can be seen in figure 3.1. A cylindrical coordinate system (R,φ,Z) is used to define hexahedron-cells with side lengths ΔR, $\Delta\varphi$ and ΔZ. This hexahedron-mesh fits to the geometry of the tokamak ASDEX Upgrade [18], but can be easily modified for any toroidal fusion device. The size of the cell-elements is crucial for the accuracy of all further calculations. One has to make a trade-off between calculation speed and accuracy.

In order to obtain tetrahedrons as cell-elements, the hexahedron is split into two prisms, as one can see in figure 3.2. Prism 2 can be handled the same way as prism 1 after the inversion of all axes. Each prism is then subsequently split into 3 tetrahedrons, as one can see as well in figure 3.2. The 3 tetrahedrons can be identified by a set of four vertices of the prism. Table 3.1 lists the tetrahedrons with their defining vertices of the prism. Furthermore, the prism vertices (1,2,3,4,5,6) are mapped to tetrahedron vertices (1,2,3,4) for each of the 3 tetrahedrons, as can be seen in table 3.2.
With this method of splitting the space into tetrahedron-cells it is guaranteed, that every face of a tetrahedron matches exactly the opposite face of the neighbouring cell with the same area and shape. In other words, every tetrahedron has only four neighbouring tetrahedrons.

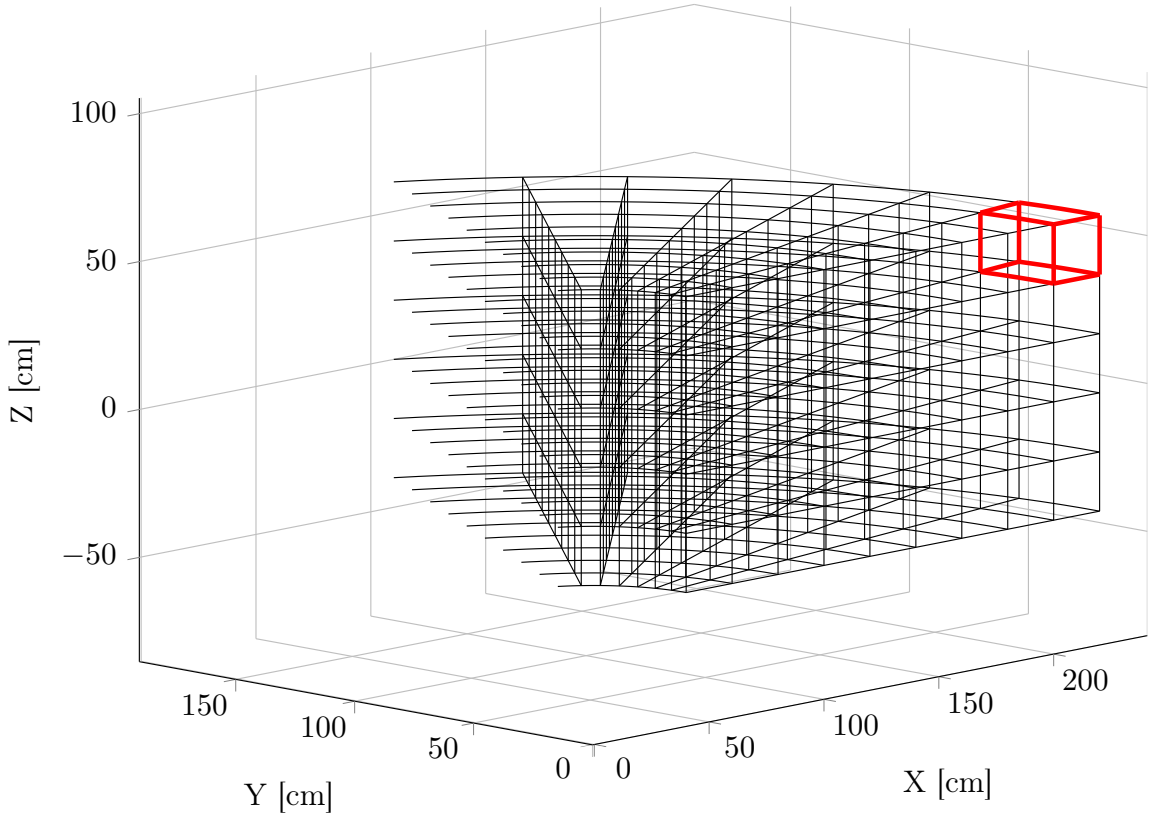
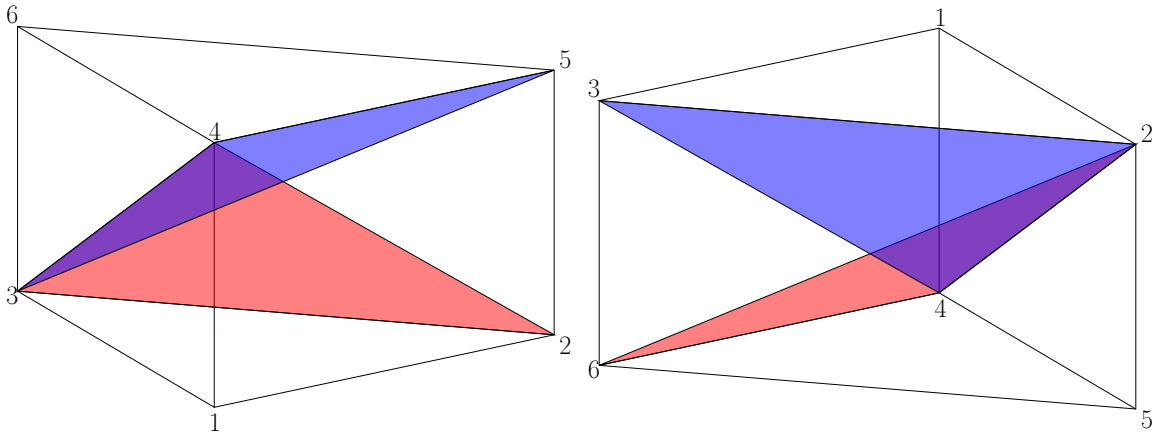


Figure 3.1: Hexahedron-mesh:
A torus-like space (hollow cylinder) is split into hexahedron-cells (red), with side lengths ΔR, $\Delta\varphi$ and ΔZ. The hollow cylinder is toroidally closed, but only a small section is depicted in order to make the visibility of the mesh better.



(a) Prism 1 (b) Prism 2

Figure 3.2: Tetrahedron-cells:
Each hexahedron of the mesh in figure 3.1 is split into two prisms, which are in each case subsequently split into three tetrahedrons.

Table 3.1: Identification of tetrahedrons from the vertices of the prism.

tetrahedron id	vertices of prism
①	1,2,3,4
②	2,3,4,5
③	3,4,5,6

Table 3.2: Mapping of prism vertices (1,2,3,4,5,6) to tetrahedron tetrahedron vertices (1,2,3,4) for each of the 3 tetrahedrons

①	
tetrahedron	prism
1	1
2	2
3	3
4	4

②	
tetrahedron	prism
1	2
2	3
3	4
4	5

③	
tetrahedron	prism
1	3
2	4
3	5
4	6

Every tetrahedron of the mesh can be described by four normal vectors n_i of four infinitely extended planes. Three of these planes go through the vertex with index-number 1. The fourth plane defines the face, that is exactly on the opposite side of vertex 1. Its normal distance to vertex 1 must be calculated.
Thus, a tetrahedron is fully described by four normal vectors n_i^α for the planes $\alpha = (1, 2, 3, 4)$ and additionally for the fourth plane the distance to the first vertex d^α. (Obviously three elements of d^α, which are the distance of three planes to vertex 1, are zero.) The location of the cell-element (tetrahedron) in space is given by the coordinates of vertex 1.
Per definition all normal vectors n_i^α are calculated in such a way, that they point to the inward of the tetrahedron.

Chapter 4

Calculate tetrahedron-specific quantities

4.1 Obtain the field quantities in the tetrahedron

The field quantities **A** (vector potential), **B** (magnetic field vector) and B (magnetic field modulus) at the mesh-points are given by a code [22, 23, 24], that uses a 5^{th} order spline interpolation for the field quantity data from an equilibrium file in the EFIT-format of a toroidal fusion device. Furthermore, also the poloidal magnetic flux Ψ is given at the mesh-points.

Since *3DGeoInt* is designed to integrate particle orbits for field quantities, that are approximated by linear functions in a cell, these linear functions have to be provided. In total the quantities A_k, B_k/ω_c, Φ and ω_c are independently from each other linearly interpolated in between the given quantities at the mesh-points. Thus, the respective quantity at the vertex number 1 and its gradient in the tetrahedron is required.

4.2 Calculate tetrahedron-specific matrix-elements a_l^i and the components of the vector b^i

In order to integrate the linear set of ordinary differential equations, the matrix-elements a_l^i and the components of the vector b^i need to be computed for every tetrahedron.

As explained in section (2.2), the parallel velocity $v_\parallel$ is a function of coordinates. In order to avoid difficulties caused by the sign of $v_\parallel$, the quantity U is introduced

$$U = \frac{v_\parallel^2}{2} = \frac{1}{m}\left(w - J_\perp \omega_c - e_\alpha \Phi\right), \tag{4.1}$$

where the total energy w and the perpendicular adiabatic invariant $J_\perp$ are constants of motion and the coordinate dependence of U lies in ω_c and Φ.
By inserting the constants of motion, given by the quantities at the particle's entry point into the tetrahedron ($w = mv^2_{\text{e.p.}}/2 + e_\alpha \Phi_{\text{e.p.}}$, $J_{\perp,\text{e.p.}} = mv^2_{\perp,\text{e.p.}}/(2\omega_{c,\text{e.p.}})$, U can be written as

$$U = \frac{1}{m}\Big(\underbrace{\frac{mv^2_{e.p.}}{2} + e_\alpha \Phi_{e.p.}}_{w} - \underbrace{\frac{mv^2_\perp}{2}}_{J_{\perp,\text{e.p.}}\cdot\omega_c} - e_\alpha \Phi \Big), \tag{4.2}$$

where $v_{e.p.}$ is the particle's absolute velocity at the entry point and $v_\perp$ is particle's gyration velocity around the guiding center perpendicular to the magnetic field. Respectively $v_{\perp,\text{e.p.}}$ is the gyration velocity at the entry point. Φ is the electrostatic potential and $\Phi_{e.p.}$ is respectively the electrostatic potential at the particle's entry point.
The particle's absolute velocity is given by

$$v = \sqrt{v_\parallel^2 + v_\perp^2}. \tag{4.3}$$

Some of the vector components of b^i contain the quantity U_0, which is U at the origin of the coordinates. $J_\perp$ is invariant all over the tetrahedron and ω_c is a linear function of the module of the magnetic field B. Thus, the particle's perpendicular kinetic energy at the origin of the coordinates can be calculated by using the quotient of the magnetic field module B_0 at the coordinate origin and $B_{\text{e.p.}}$ and the particle's entry point

$$\left.\frac{mv^2_\perp}{2}\right|_{x^i=0} = \underbrace{\frac{mv^2_{\perp,\text{e.p.}}}{2\omega_{c,e.p.}}}_{J_\perp,\text{e.p.}} \cdot \omega_{c,0} = \frac{m^2 v^2_{\perp,\text{e.p.}} c}{2eB_{e.p.}} \cdot \frac{eB_0}{mc} = \frac{mv^2_{\perp,\text{e.p.}}}{2B_{e.p.}} \cdot B_0, \tag{4.4}$$

where $\omega_{c,e.p.}$ is the cyclotron frequency at the entry point.

Due to the fact that also the electrostatic potential Φ is a linear function all over the tetrahedron

$$\Phi = \Phi_0 + x^i \frac{\partial \Phi}{\partial x^i}, \tag{4.5}$$

the quantity U_0 at the coordinate origin can be expressed by

$$U_0 = \frac{1}{m}\left(\frac{mv^2_{e.p.}}{2} + e_\alpha \Phi_{e.p.} - \frac{mv^2_{\perp,\text{e.p.}}}{2B_{e.p.}} \cdot B_0 - e_\alpha \Phi_0 \right). \tag{4.6}$$

Furthermore the gradient of the quantity U is

$$\frac{\partial U}{\partial x^i} = -\frac{v_{\perp,\text{e.p.}}^2}{2B_{e.p.}} \cdot \frac{\partial B}{\partial x^i} - \frac{e_\alpha}{m} \cdot \frac{\partial \Phi}{\partial x^i}. \tag{4.7}$$

The gradient of the electrostatic potential, namely the electric field, can be separated into components parallel and perpendicular to the magnetic field.

$$\frac{\partial \Phi}{\partial x^i} = -E_i = -E_{\parallel,i} - E_{\perp,i} \tag{4.8}$$

Therefore it is useful to introduce $\mathbf{h} = \mathbf{B}/B$ as the unit vector along the magnetic field, in order to express the electric field components

$$\begin{aligned} E_{\parallel,i} &= \left(-\frac{\partial \Phi}{\partial x^j} \cdot h^j\right) h_i \qquad \text{and} \\ E_{\perp,i} &= E_i - E_{\parallel,i} = -\frac{\partial \Phi}{\partial x^i} + \left(\frac{\partial \Phi}{\partial x^j} \cdot h^j\right) h_i. \end{aligned} \tag{4.9}$$

The electric field parallel to the magnetic field causes an acceleration of the particle's guiding center into the direction of the magnetic field. Whereas the electric field perpendicular to the magnetic field causes an $\mathbf{E} \times \mathbf{B}$-drift of the guiding center into the third direction, perpendicular to both the magnetic field and the respective electric field.
Thus, a decrease of the electrostatic potential energy $e\Phi$ in the direction of the magnetic field, namely an electric field $E_{\parallel,i}$, yields to an increase of the parallel kinetic energy and vice versa. Therefore, the parallel velocity $v_\parallel$ is modified.

Things are more complicated, if the particle doesn't move along the magnetic field, but drifts across, due to an electric field $E_{\perp,i}$. First of all we use a standard drift ordering, where the electric field is weak, such that the $\mathbf{E} \times \mathbf{B}$-drift velocity is of the same order as the magnetic drift velocity. In this ordering $\frac{\rho}{L} << 1$, where ρ is the Larmor radius and L is the spatial scale of the electromagnetic field. Thus, the kinetic energy of the $\mathbf{E} \times \mathbf{B}$-drift is quadratic in the Larmor radius and should be ignored in this ordering. Therefore the $\mathbf{E} \times \mathbf{B}$-drift doesn't change the electrostatic potential energy.

Furthermore one has to distinguish in between uniform and not uniform magnetic fields. In the first case, if the magnetic field is uniform, there is no magnetic field drift present and the cross-field drift is purely an $\mathbf{E} \times \mathbf{B}$-drift. Such a drift goes along contours of Φ by definition. The electrostatic potential Φ, respectively, stays constant

along the drift orbit. Thus, the electrostatic potential energy $e\Phi$ remains unchanged and the parallel velocity $v_\parallel$ is not modified.
If the magnetic field is not uniform, magnetic field drifts occur and such drifts do not need to go along contours of Φ. But the $\mathbf{E}\times\mathbf{B}$-drift crosses contours of B, and since $J_\perp$ is invariant, the changing magnetic field causes a change in the perpendicular gyration velocity $v_\perp$. Thus, a decrease of the electrostatic potential energy, yields to an increase of kinetic perpendicular gyration energy and vice versa. Once again the parallel velocity $v_\parallel$ is not modified.

If we insert the calculated quantities U_0, $\frac{\partial U}{\partial x^i}$ into equation (2.11) and (2.12), and use $\frac{B_k}{\omega_c} = h_k \cdot \frac{cm}{e_\alpha}$, we obtain for the matrix elements

$$
\begin{aligned}
a_l^i &= \varepsilon^{ijk}\left[\frac{2cm}{e_\alpha}\left(-\frac{v_{\perp,\text{e.p.}}^2}{2B_{e.p.}}\cdot\frac{\partial B}{\partial x^l}-\frac{e_\alpha}{m}\cdot\frac{\partial\Phi}{\partial x^l}\right)\frac{\partial}{\partial x^j}h_k\right.\\
&\quad\left.+\frac{cm}{e_\alpha}\left(-\frac{v_{\perp,\text{e.p.}}^2}{2B_{e.p.}}\cdot\frac{\partial B}{\partial x^j}-\frac{e_\alpha}{m}\cdot\frac{\partial\Phi}{\partial x^j}\right)\frac{\partial}{\partial x^l}h_k\right]\\
&= -\frac{v_{\perp,\text{e.p.}}^2}{2B_{e.p.}}\cdot\frac{cm}{e_\alpha}\cdot\underbrace{\varepsilon^{ijk}\left(2\frac{\partial B}{\partial x^l}\frac{\partial}{\partial x^j}h_k+\frac{\partial B}{\partial x^j}\frac{\partial}{\partial x^l}h_k\right)}_{\alpha_l^i}-c\cdot\underbrace{\varepsilon^{ijk}\left(2\frac{\partial\Phi}{\partial x^l}\frac{\partial}{\partial x^j}h_k+\frac{\partial\Phi}{\partial x^j}\frac{\partial}{\partial x^l}h_k\right)}_{\beta_l^i}
\end{aligned}
$$

$$
\begin{aligned}
&\text{for}\qquad 1\le i,l\le 3,\\
a_4^i &= \varepsilon^{ijk}\frac{\partial A_k}{\partial x^j}\qquad\text{for}\qquad 1\le i\le 3,\\
a_l^4 &= 0\qquad\text{for}\qquad 1\le l\le 3,\\
a_4^4 &= -\frac{v_{\perp,\text{e.p.}}^2}{2B_{e.p.}}\cdot\frac{cm}{e_\alpha}\cdot\operatorname{tr}\{\alpha_l^i\}-c\cdot\operatorname{tr}\{\beta_l^i\}\qquad\text{for}\qquad 1\le i,l\le 3,
\end{aligned}
\tag{4.10}
$$

and for the components of vector b^i

$$
\begin{aligned}
b^i &= \frac{cm}{e_\alpha} \cdot \varepsilon^{ijk} \left[\frac{2}{m} \left(\frac{m v_{e.p.}^2}{2} + e_\alpha \Phi_{e.p.} - \frac{m v_{\perp,\mathrm{e.p.}}^2}{2 B_{e.p.}} \cdot B_0 - e_\alpha \Phi_0 \right) \frac{\partial}{\partial x^j} h_k \right. \\
&\quad \left. + h_{k,0} \cdot \left(- \frac{v_{\perp,\mathrm{e.p.}}^2}{2 B_{e.p.}} \cdot \frac{\partial B}{\partial x^j} - \frac{e_\alpha}{m} \cdot \frac{\partial \Phi}{\partial x^j} \right) \right] \\
&= \frac{cm}{e_\alpha} \cdot \varepsilon^{ijk} \left[\left(v_{e.p.}^2 - \frac{v_{\perp,\mathrm{e.p.}}^2}{B_{e.p.}} \cdot B_0 \right) \frac{\partial}{\partial x^j} h_k - \frac{v_{\perp,\mathrm{e.p.}}^2}{2 B_{e.p.}} \cdot \frac{\partial B}{\partial x^j} \cdot h_{k,0} \right] \\
&\quad - c \cdot \varepsilon^{ijk} \left[2 \left(\Phi_0 - \Phi_{\mathrm{e.p.}} \right) \cdot \frac{\partial}{\partial x^j} h_k + \frac{\partial \Phi}{\partial x^j} \cdot h_{k,0} \right] \\
&\quad \text{for} \qquad 1 \leq i \leq 3, \\
b^4 &= \left(- \frac{v_{\perp,\mathrm{e.p.}}^2}{2 B_{e.p.}} \cdot \frac{\partial B}{\partial x^i} - \frac{e_\alpha}{m} \cdot \frac{\partial \Phi}{\partial x^i} \right) \cdot \varepsilon^{ijk} \frac{\partial A_k}{\partial x^j}
\end{aligned}
\tag{4.11}
$$

where $h_{k,0}$ is the unit vector along the magnetic field at the coordinate origin.

Chapter 5

Calculation of the orbit in a tetrahedron

5.1 Introduction

The calculation of the orbit in a tetrahedron is done by the routine *pusher_ tetr_ orb.f90*. As one can see in section (3), the space has to be split into small cells (tetrahedrons) for our purpose. These tetrahedrons are given by four planes α, that are described by inward pointing normal vectors n_i^α, and the coordinates of the vertex with index 1. Three out of these four planes pass through this vertex with the index 1. Additionally for the fourth plane the distance to the first vertex d^α is given. (Obviously three elements of d^α are zero.)

A particle enters this tetrahedron at a given position $x^i_{e.p.}$ with the parallel velocity $v_{\parallel,e.p.}$ and the perpendicular velocity $v_{\perp,e.p.}$. Furthermore the respective plane (*iface*), through which the particle enters the tetrahedron, must be defined. The exact equations of motions are approximated in these tetrahedrons by a linear set of ordinary differential equations (2.9). The result of this initial value problem is the point of intersection $x^i(\tau)$ in between the orbit and the tetrahedron (the point, where the particle leaves the tetrahedron) and $v_\parallel(\tau)$ at this point. For the calculation of this implicitly τ-dependent set of ODEs the Runge-Kutta method is used. Usually, a numerical standard approach is to guess an integration time (in our case τ) and iteratively repeat the Runge-Kutta method, until the respective exit point is found. Since the goal of *3DGeoInt* is to calculate a vast amount of orbits, a high number of "orbit-pushings" through tetrahedrons has to be computed. Therefore the calculation of the orbit in a tetrahedron has to be as efficient as possible. In the following subsections the functional principle of the routine *pusher_ tetr_ orb.f90* is explained in detail.

5.2 Approximate solution of τ to pass the tetrahedron

To find the point of intersection in between the orbit and the tetrahedron (exit point), first of all one has to integrate the set of ODEs (2.9). Furthermore one has to find a way how to determine the point of intersection. The four faces of the tetrahedron are given by four infinitely extended planes, more specifically by their normal vectors n_i^α. To understand through which of these four planes the orbit leaves the tetrahedron, intersection points with all four planes have to be calculated.

As one can easily see in figure 5.1, the point of intersection that can be reached in the smallest positive "time" τ is the respective one.
As explained above, the calculation of the orbits should be as efficient as possible. That's why the better the guessed τ for the Runge-Kutta method can be approximated, the more efficient the integrator will be. The first goal is to find an approximate solution for the guessed τ.

5.2.1 Approximation to integrate the set of ODEs

The approach that is used to obtain a good guess for τ is to simplify the set of ODEs by setting the implicitly τ-dependent variables to constants of motion. The set of ODEs can then be analytically solved.

As one can see in section (2.2) the linear set of ordinary differential equations can be put in the standard form (2.10):

$$\frac{\mathrm{d}z^i(\tau)}{\mathrm{d}\tau} = a_l^i z^l(\tau) + b^i,$$

where $z^i(\tau) = x^i(\tau)$ for $1 \leq i \leq 3$ and $z^4(\tau) = v_\parallel(\tau)$.
If we consinder the equation of motion of $v_\parallel$ (2.7) in the standard form (2.10), $\frac{dv_\parallel}{d\tau}$ has the form

$$\frac{dv_\parallel(\tau)}{d\tau} = a_4^4 \cdot v_\parallel(\tau) + b^4. \tag{5.1}$$

Since A_k, $\frac{B_k}{\omega_c}$, Φ, and ω_c are independently linearly interpolated in the tetrahedron, a_4^4 and b^4 are constants. As a first approximation, we set $v_\parallel(\tau) = v_{\parallel,const.}$ all over the tetrahedron.

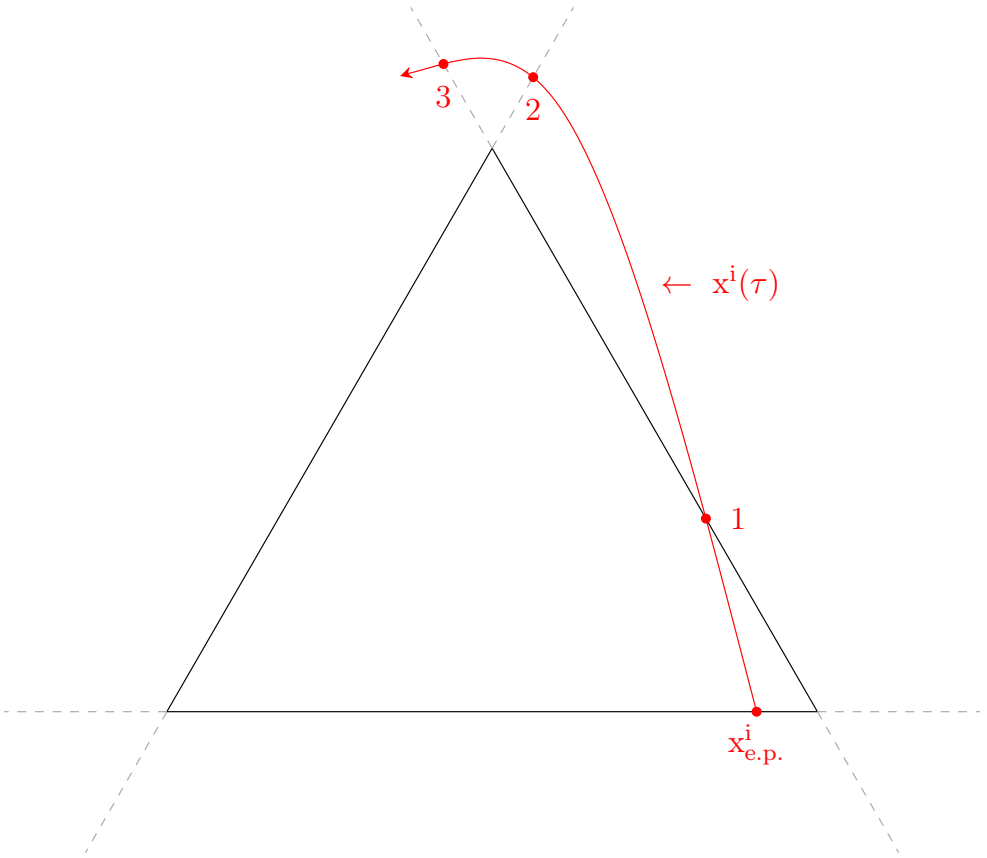


Figure 5.1: Intersections in between the orbit and the cell's faces:
For a better understanding, the 3D-cell (tetrahedron) is depicted as a 2D-triangle. The figure shows the particle's orbit, starting at $x^i_{\text{e.p.}}$, the particle's entry point into the tetrahedron. The cell is left by the particle at point ①, the intersection in between the orbit and one of the tetrahedron's faces. Further on, the particle will have more intersections with other planes at the points ② and ③, respectively. Obviously, the right exit point, through which the particle will leave the tetrahedron, is the intersection point that can be reached in the smallest positive "time" τ.

This approximation enables us to integrate equation (5.1) and one obtains

$$v_{\parallel}(\tau) = (a^4_4 \cdot v_{\parallel,const.} + b^4)\tau + v_{\parallel,e.p.}, \tag{5.2}$$

where $v_{\parallel,e.p.} = v_{\parallel}(\tau = 0)$ is the parallel velocity at the entry point of the tetrahedron.

The linear set of ODEs for the coordinates x^i in standard form is

$$\frac{\mathrm{d}x^i(\tau)}{\mathrm{d}\tau} = \frac{\mathrm{d}z^i(\tau)}{\mathrm{d}\tau} = a^i_l z^l(\tau) + b^i \qquad for \qquad 1 \leq i \leq 3. \tag{5.3}$$

By splitting the matrix multiplication $a^i_l z^l(\tau)$ into two terms $1 \leq l \leq 3$ and $l = 4$, one

gets

$$\frac{\mathrm{d}x^i(\tau)}{\mathrm{d}\tau} = a_l^i x^l(\tau) + a_4^i z^4(\tau) + b^i \qquad for \qquad 1 \leq i,l \leq 3, \tag{5.4}$$

where $z^4(\tau) = v_{\parallel}(\tau)$ can be replaced by the approximation for $v_{\parallel}(\tau)$, namely equation (5.2):

$$\frac{dx^i(\tau)}{d\tau} = a_l^i x^l(\tau) + a_4^i(a_4^4 \cdot v_{\parallel,const.} + b^4)\tau + a_4^i v_{\parallel,e.p.} + b^i \qquad for \qquad 1 \leq i,l \leq 3. \tag{5.5}$$

Once again it is necessary to set the τ-dependent variable constant in order to be able to integrate the differential equation: $x^l(\tau) = x^l_{const.}$. It is easy to analyze, which values can be taken by $x^l_{const.}$. Obviously only coordinates x^l that are inside the tetrahedron are allowed as constants.

The integration over $d\tau$ yields

$$x^i(\tau) = \frac{1}{2}a_4^i(a_4^4 \cdot v_{\parallel,const.} + b^4)\tau^2 + (a_4^i v_{\parallel,e.p.} + b^i + a_l^i x^l_{const.})\tau + x^i_{e.p.}, \tag{5.6}$$

where $x^i_{e.p.} = x^i(\tau = 0)$ are the coordinates of the particle's entry point into the tetrahedron.

The point of intersection in between the orbit and the tetrahedron must lie on one of the four planes that span the tetrahedron. By multiplying the normal vectors n_i^α of the 4 planes ($\alpha = 1 \ldots 4$) with the equation for $x^i(\tau)$, the result is the distance in between $x^i(\tau)$ and the plane α, that passes through the coordinate origin. The particle leaves the tetrahedron, when this distance is zero.

$$\underbrace{n_i^\alpha \cdot x^i(\tau)}_{dist \stackrel{!}{=} 0} = \frac{1}{2}n_i^\alpha \cdot a_4^i(a_4^4 \cdot v_{\parallel,const.} + b^4)(\tau^\alpha)^2 + n_i^\alpha \cdot (a_4^i v_{\parallel,e.p.} + b^i + a_l^i x^l_{const.})\tau^\alpha + n_i^\alpha \cdot x^i_{e.p.}. \tag{5.7}$$

Furthermore one hast to subtract d^α, since one of the four planes is not passing through the origin, but is in distance d from the origin. (Three elements of d^α are zero.)
τ^α is a vector with 4 τ-values that solves the quadratic equation for the respective

plane α. Thus one obtains four quadratic equations for τ^α:

$$\frac{1}{2}\underbrace{n_i^\alpha \cdot a_4^i(a_4^4 \cdot v_{\parallel,const.} + b^4)}_{a^\alpha}(\tau^\alpha)^2 + \underbrace{n_i^\alpha \cdot (a_4^i v_{\parallel,e.p.} + b^i + a_l^i x_{const.}^l)}_{b^\alpha}\tau^\alpha + \underbrace{n_i^\alpha \cdot x_{e.p.}^i - d^\alpha}_{c^\alpha} = 0 \tag{5.8}$$

$$\frac{1}{2}a^\alpha(\tau^\alpha)^2 + b^\alpha\tau^\alpha + c^\alpha = 0 \tag{5.9}$$

5.2.2 Estimation of $v_{\parallel,const.}$ and $x_{const.}^l$

In order to obtain an approximative analytical value of τ^α, that is as close as possible to the value, that can be later numerically calculated by an iterative use of the Runge-Kutta method, the following assumptions are made:

1. The parallel velocity $v_\parallel$ doesn't change while the particle is passing the tetrahedron.

   $v_{\parallel,const.} = v_{\parallel,e.p.}$

2. The Matrix $a_l^i = 0$ for $1 \leq i, l \leq 3$. This is equivalent to the assumption of setting $x_{const.}^l$ to the coordinate origin.

   $x_{const.}^l = 0$

5.2.3 Calculate the approximate solution of τ

The point of intersection in between the orbit and the tetrahedron is very likely the one, that can be obtained with the smallest τ of τ^α. If the real orbit passes the tetrahedron, definitely the respective plane is the one, that can be reached in the smallest positive "time" τ. But since we made approximations in order to be able to integrate the set of ODEs, this law is no longer guaranteed to be true. The constants $x_{const.}^l$ and $v_{\parallel,const.}$ are in reality dependent on τ and therefore the real orbit might leave the tetrahedron through a different face than the approximated orbit. Figure 5.2 depicts a situation, where $x^l(\tau)$ and $v_\parallel(\tau)$ are not constant during motion and thus, the approximated orbit differs from the real orbit.
Nevertheless, we will look for the smallest τ of τ^α to get an approximate solution of τ to pass the tetrahedron.

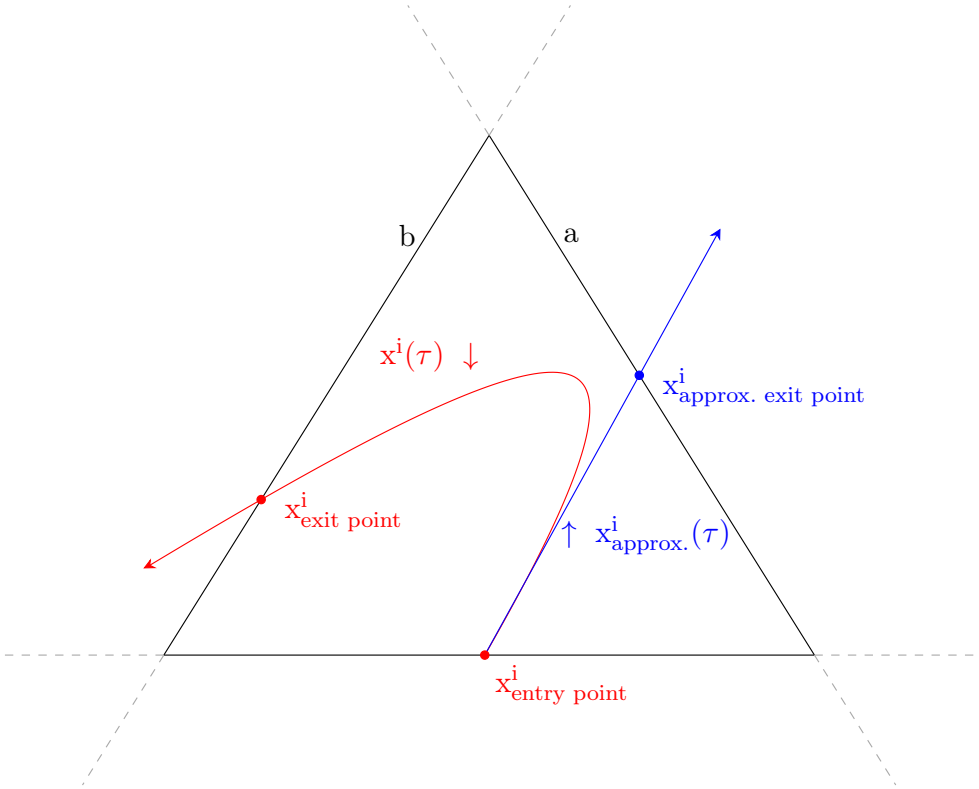


Figure 5.2: Influence of the approximation $x^l(\tau) = x^l_{const.}$ and $v_{\parallel}(\tau) = v_{\parallel,const.}$ on the orbit: For the real orbit (red) the quantities $x^l(\tau)$ and $v_{\parallel}(\tau)$ are not constants during motion, whereas they are ones for the approximated orbit (blue). Since the analytical approximation of integration variable τ is done for approximated quantities, the law that the right exit point is reached in the smallest positive "time" τ, is not guaranteed. For the approximated orbit, the intersection point with a plane that is reached in smallest positive τ is located at plane a, whereas the real exit point is located at plane b.

The general solutions of equation (5.9) for τ^α are

$$\begin{aligned} \tau_1^\alpha &= \frac{-b^\alpha + \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \qquad \text{and} \\ \tau_2^\alpha &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \end{aligned} \tag{5.10}$$

with the constants

$$\begin{aligned} a^\alpha &= n_i^\alpha \cdot a_4^i (a_4^4 \cdot v_{\parallel,e.p.} + b^4), \\ b^\alpha &= n_i^\alpha \cdot (a_4^i v_{\parallel,e.p.} + b^i) \quad \text{and} \\ c^\alpha &= n_i^\alpha \cdot x_{e.p.}^i - d^\alpha. \end{aligned} \tag{5.11}$$

One can easily show, that $c^\alpha \geq 0$. The quadratic equation coefficient c is zero, if the orbit leaves the tetrahedron through the same face where it entered (the tetrahedron). In all other cases, it is greater than zero. Mind the case, where the particle leaves the tetrahedron through the face, where the plane doesn't pass through the origin. In this case $n_i \cdot x_{e.p.}^i < 0$, but the respective $d < 0$, thus $c > 0$.
In principle one has to make sure, that the discriminant $(b^\alpha)^2 - 2a^\alpha c^\alpha$ is positive, otherwise no real solutions for orbit-tetrahedron-intersections can be found.

One can show, that for the inequality

$$\begin{aligned} 0 < \tau_1^\alpha &< \tau_2^\alpha \\ 0 < \frac{-b^\alpha + \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} &< \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \end{aligned} \tag{5.12}$$

the following solutions exist:

$$a^\alpha < 0, \quad b^\alpha > 0 \quad \text{and} \quad \frac{(b^\alpha)^2}{2a^\alpha} < c^\alpha < 0 \tag{5.13}$$

Whereas for the inequality

$$\begin{aligned} 0 < \tau_2^\alpha &< \tau_1^\alpha \\ 0 < \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} &< \frac{-b^\alpha + \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \end{aligned} \tag{5.14}$$

the following solutions exist:

$$a^\alpha > 0, \quad b^\alpha < 0 \quad \text{and} \quad 0 < c^\alpha < \frac{(b^\alpha)^2}{2a^\alpha} \tag{5.15}$$

Furthermore one also has to consider the inequality

$$\begin{aligned} \tau_1^\alpha < 0 &< \tau_2^\alpha \\ \frac{-b^\alpha + \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} < 0 &< \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \end{aligned} \tag{5.16}$$

with the solutions

$$a^\alpha < 0 \quad \text{and} \quad c^\alpha > 0 \tag{5.17}$$

and the inequality

$$\begin{aligned} \tau_2^\alpha < 0 \quad &< \quad \tau_1^\alpha \\ \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} < 0 \quad &< \quad \frac{-b^\alpha + \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \end{aligned} \tag{5.18}$$

with the solutions

$$a^\alpha > 0 \quad \text{and} \quad c^\alpha < 0. \tag{5.19}$$

If the discriminant becomes zero ($c = \frac{(b^\alpha)^2}{2a^\alpha}$, $\tau^\alpha = -\frac{b^\alpha}{a^\alpha}$), or c^α becomes zero ($\tau^\alpha = -\frac{2b^\alpha}{a^\alpha}$) there are solutions for

$$\begin{aligned} a^\alpha > 0 \quad &, \quad b^\alpha < 0 \quad \text{or} \\ a^\alpha < 0 \quad &, \quad b^\alpha > 0. \end{aligned} \tag{5.20}$$

In our problem $c^\alpha \geq 0$, that's why we can neglect the solution τ_1^α. Furthermore the probability, that the discriminant is zero, is very low, thus we will also neglect this case. In conclusion the smallest τ can be found among the following solutions:

$$\begin{aligned} \tau^\alpha &= -\frac{2b^\alpha}{a^\alpha} \quad \text{for} \quad c = 0, \quad \text{or} \\ \tau^\alpha &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \quad \text{for} \quad a^\alpha < 0, \quad \text{or} \\ \tau^\alpha &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \quad \text{for} \quad a^\alpha > 0 \quad \text{and} \quad b^\alpha < 0. \end{aligned} \tag{5.21}$$

In order to calculate the smallest τ with the least computational power, one must compute the coefficients a^α, b^α and c^α for all four planes. If $c^\alpha > 0$, one has to validate, if the discriminant $(b^\alpha)^2 - 2a^\alpha c^\alpha$ is positive. Only those τ^α have to be calculated, where the discriminant is positive, and one can discard the calculation, if both a^α and b^α are positive. Inevitably τ^α for $c = 0$ must be computed.
Obviously the guessed τ is the smallest among the calculated positive values. We assume that the particle will leave the tetrahedron through the respective face α.

5.2.4 No approximated analytical positive solution for τ exists

Furthermore one has to consider a special case, where the analytical approximation for τ fails, and therefore no positive solution for τ exists. As described above, in order to obtain an approximative analytical value of τ, $v_{\parallel,const.}$ was set to $v_{\parallel,e.p.}$. Let's

consider an orbit that enters a certain face, turns around close to this face and leaves the tetrahedron through the same face as it entered. (See figure 5.3) According to the described analytical approximation, in this case the quadratic equation coefficient c is zero and the solution for τ can be obtained by

$$\tau = -\frac{2b}{a},$$

where both a and b depend on $v_{\parallel,const.}$.
The approximation $v_{\parallel,const.} = v_{\parallel,e.p.}$ is not valid anymore in this special case, because $v_{\parallel}(\tau)$ changes sign when the orbit turns around. Due to neglecting the change of sign in the approximation, τ yields a negative and therefore invalid result.
Since the considered orbit turns very close to the face, no approximate analytical solutions with the other faces exist. Whereas, a solution should only exist with the face, through which the particle entered.

This special case, where no approximated analytical positive solution for τ exists, will be treated separately in section (5.4).

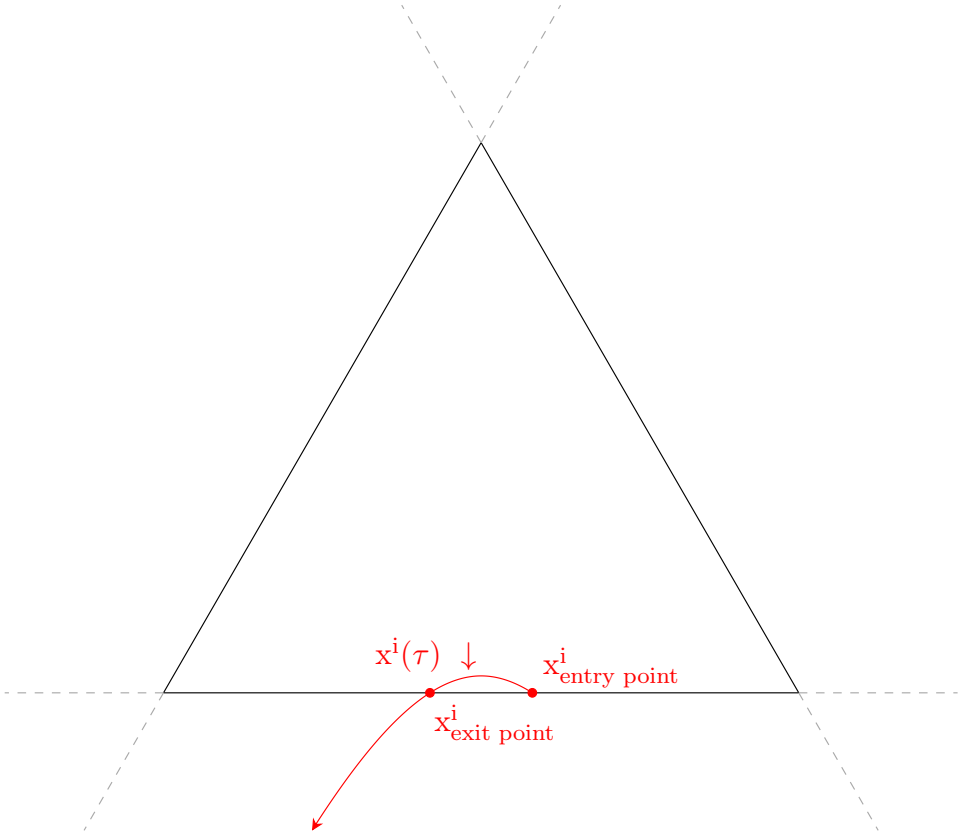


Figure 5.3: Orbits makes a turn close to the incoming face:
Since the approximation $v_{\parallel,const.} = v_{\parallel,e.p.}$ is not valid anymore, the analytical approximation can yield an invalid result.

5.3 Calculate the orbit with the Runge-Kutta method

The following section is loosely based on [14].

In order to obtain the particle's coordinates $x^i(\tau)$ and $v_\parallel(\tau)$ for a certain τ, the implicitly τ-dependent differential equation

$$\frac{\mathrm{d}z^i(\tau)}{\mathrm{d}\tau} = \underbrace{a^i_l z^l(\tau) + b^i}_{f(z^l(\tau))}, \tag{5.22}$$

where $x^i(\tau) = z^i(\tau)$ for $1 \leq i \leq 3$ and $v_\parallel(\tau) = z^4(\tau)$, has to be solved for the initial values $x^i_0 = x^i(\tau_0)$ and $v_{\parallel,0} = v_\parallel(\tau_0)$.
The subroutine *rk4*, which uses the classical 4^{th} order Runge-Kutta method, is used to solve this initial value problem numerically.

The classical 4th order Runge-Kutta method is a numerical solution method to solve initial value problems of ordinary differential equations. Hence, differential quotients are approximated by difference quotients. Obviously for non-linear functions, errors will occur necessarily, due to neglecting higher order terms of the Taylor series expansion. These errors can be reduced by combining various difference quotients. The classical 4^{th} order Runge-Kutta method is such a method, that compensates discretization errors up to the 3^{rd} derivative.

The set of ordinary differential equations

$$\frac{\mathrm{d}z^i(\tau)}{\mathrm{d}\tau} = f(z^l(\tau)), \quad z^l_0 = z^l(\tau_0) \tag{5.23}$$

is a 1^{st} order initial value problem. For a certain step-size h, the classical Runge-Kutta method for calculating the approximation value $u_{j+1} \approx z^i(\tau_{j+1})$ has the procedure function

$$\Phi(\tau_j, u_j, h, f) = \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4, \tag{5.24}$$

with the four difference quotients

$$\begin{aligned} k_1 &= f\left(\tau_j, u_j\right), \\ k_2 &= f\left(\tau_j + \frac{h}{2}, u_j + \frac{h}{2}k_1\right), \\ k_3 &= f\left(\tau_j + \frac{h}{2}, u_j + \frac{h}{2}k_2\right), \\ k_4 &= f\left(\tau_j + h, u_j + hk_3\right). \end{aligned} \tag{5.25}$$

Thus, the recursion equation to calculate the approximation is

$$\begin{aligned} u_{j+1} &= u_j + h \cdot \Phi(\tau_j, u_j, h, f) \\ &= u_j + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \qquad i = 0, 1, \ldots \end{aligned} \tag{5.26}$$

The difference quotient k_4 for $1 \leq i \leq 3$ gives the particle's approximate velocity v^i after the integration for the step h. The first iteration step is always calculated with $h = \tau$, for the guessed τ of the analytical approximation from the previous section. In order to achieve a certain accuracy for the particle's exit point of the tetrahedron, more iteration steps may be needed and the step size must be changed. The following section (5.5) will demonstrate, how to propose a new τ for the Runge-Kutta step size h.

Figure 5.4 depicts, how the classical 4^{th} order Runge-Kutta method uses four difference quotients to obtain an approximate value of e.g. $z(\tau) = z^1(\tau)$.

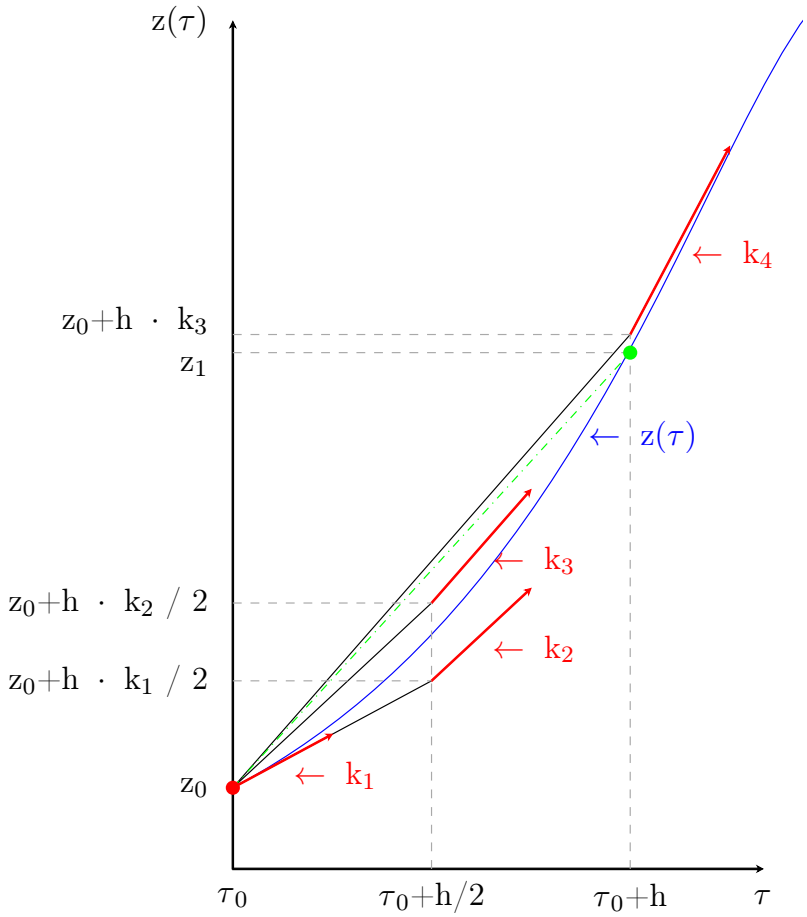


Figure 5.4: The classical 4^{th} order Runge-Kutta method combines four difference quotients (red) in order to obtain an approximation value (green) of the real function (blue) at the function value $\tau_0 + h$.

5.4 Calculate the orbit with the Runge-Kutta method, when no analytical approximation exists

As described in the previous section 5.2.4, the analytical approximation can fail for orbits that turn close to the face, through which the particle entered. By assuming that the particle leaves through the same face, the solution is given by

$$\tau = -\frac{2b}{a},$$

which yields a negative result in case of an approximation error.
In order to apply a numerical procedure to find the exit coordinates $x^i(\tau)$, the absolute

value of the equation above is taken as a first approximation

$$\tau_{\text{new}} = \left| -\frac{2b}{a} \right|. \tag{5.27}$$

By using the routine *rk4* the Runge-Kutta method is applied for the guessed τ_{new}. The normal velocity $v_{norm}(\tau_{\text{new}})$ regarding the assumed plane after the Runge-Kutta iteration can be calculated by multiplying the particle's velocity $v^i(\tau_{\text{new}})$, that is obtained with the Runge-Kutta method, with the plane's normal vector n_i:

$$v_{norm}(\tau_{\text{new}}) = n_i \cdot v^i(\tau_{\text{new}}) \tag{5.28}$$

The plane's normal vector is pointing to the inward of the tetrahedron, thus $v_{norm}(\tau_{\text{new}})$ is positive, if the particle moves inwards the tetrahedron, and negative, when it moves outwards.

The aim is to find the location $x^i(\tau)$, where $v_{norm}(\tau)$ becomes negative. In other words: the location after the orbit has turned. Therefore $v_{norm}(\tau_{\text{new}})$ is calculated after the first Runge-Kutta iteration for τ_{new}. If $v_{norm}(\tau_{\text{new}})$ is positive, a new τ is guessed for a consecutive Runge-Kutta iteration:

$$\tau_{\text{new}} := 2 \cdot \tau_{\text{new}} \tag{5.29}$$

This procedure is repeated until $v_{norm}(\tau_{\text{new}})$ becomes negative or the maximum number of iterations is exceeded.

5.5 Iteration procedure

5.5.1 Convergence criterion

After the Runge-Kutta method has been used to calculate the coordinates $x^i(\tau)$ of the orbit and $v_{\parallel}(\tau)$ for the analytically approximated guess for τ, one is interested in calculating the exit point (intersection with the plane) to a certain accuracy. Therefore one has to define the relative accuracy ϵ. Since the volume of each tetrahedron is not necessarily the same, the absolute accuracy can be calculated by

$$\text{dist}_{\text{min}} = \epsilon \cdot d, \tag{5.30}$$

where d is the nonzero element of d^α, namely the distance between the coordinate origin and the one plane, that doesn't pass through the origin.

Our goal is to propose a new τ and compute another Runge-Kutta calculation. Then one repeats this procedure until the distance in between the exit point $x^i(\tau)$ and the respective plane α is smaller than $\text{dist}_{\min}$.

$$n_i \cdot x^i(\tau) \overset{!}{<} \text{dist}_{\min} \tag{5.31}$$

In case that this convergence criteria cannot be fulfilled, the respective plane is not the correct plane, through which the particle leaves the tetrahedron. The plane was initially proposed, by assuming that the particle will leave the tetrahedron through the face that can be reached in the smallest positive τ. But since τ was approximated, the estimation for the appropriate face might have been wrong. For non-converging cases, a new approximation for τ with another face has to be made.

5.5.2 Calculate a new τ with Newton's method

For a clearer understanding, let's set τ after the initial Runge-Kutta iteration to τ_0 and $x^i(\tau_0)$ to x_0^i. Therefore we will try to suggest a new τ with Newton's method to integrate the orbit with the Runge-Kutta method from $x^i(\tau_0)$ to $x^i(\tau)$. The suggested face, through which the particle leaves the tetrahedron, is represented by n_i.

The distance between the particle's position and the plane after the initial Runge-Kutta iteration is $n_i \cdot x_0^i$. We are looking for a position $x^i(\tau)$ depending on τ, for which this distance becomes zero.

$$\underbrace{n_i \cdot x^i(\tau)}_{F(\tau)} \overset{!}{=} 0 \tag{5.32}$$

If we make a first degree Taylor series expansion, one obtains

$$F(\tau) = F(\tau_0) + F'(\tau_0) \cdot (\tau - \tau_0) = F(\tau_0) + F'(\tau_0)\tau - F'(\tau_0)\tau_0 \overset{!}{=} 0. \tag{5.33}$$

By setting τ_0 to 0 and rearranging the equation to express τ explicitly, the result is

$$\tau = -\frac{F(\tau_0)}{F'(\tau_0)}, \tag{5.34}$$

where $F(\tau_0)$ is the initial distance to the plane and $F'(\tau_0)$ is the normal velocity $v_{\text{norm}}(\tau_0)$, that is obtained from the precedent Runge-Kutta iteration:

$$\tau = -\frac{n_i \cdot x_0^i}{v_{\text{norm}}(\tau_0)} \tag{5.35}$$

The Runge-Kutta method is applied for this new τ. Afterwards the distance $F(\tau)$ in between the resulting position $x^i(\tau)$ to the plane is compared with the distance $F(\tau_0)$ in between $x^i(\tau_0)$ and the plane. There are two possibilities: Either the distance to the plane became smaller with the use of Newton's method and the consecutive calculation of a Runge-Kutta iteration, or the distance became larger. If the distance became smaller (see figure 5.5), Newton's method and Runge-Kutta will be repeated until the convergence criterion is fulfilled or a certain number of iteration steps is exceeded. If the distance became larger, a quadratic approach is taken into consideration (see figure 5.6).

$$\begin{aligned} \underbrace{n_i \cdot x^i(\tau)}_{F(\tau)\text{: new distance}} &< \underbrace{n_i \cdot x^i(\tau_0)}_{F(\tau_0)\text{: old distance}} \quad \rightarrow \quad \text{Newton's method} \\ n_i \cdot x^i(\tau) &> n_i \cdot x^i(\tau_0) \quad \rightarrow \quad \text{quadratic approach} \end{aligned} \tag{5.36}$$

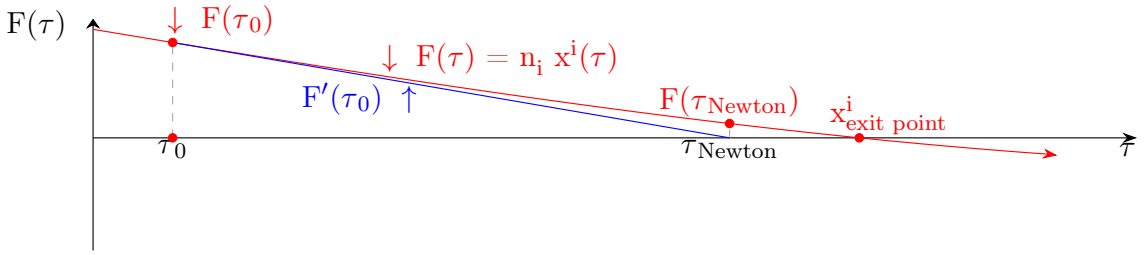


Figure 5.5: The use of Newton's method with a subsequent calculation of a Runge-Kutta iteration yielded to a smaller distance to the plane $F(\tau_{\text{Newton}})$, than the initial distance $F(\tau_0)$, before the iteration step.

5.5.3 Calculate a new τ with a quadratic approach

Let us assume, that the distance in between the orbit and the face, through that it is assumed that the particle will leave the tetrahedron, has a quadratic behaviour. As one can easily see in figure 5.6 (a), (b), (c) and (d) there are four different quadratic behaviour possibilities, for that reason the use of Newton's method will result in a larger distance $F(\tau_{\text{Newton}})$ to the plane than the previous iteration step $F(\tau_0)$.

In this case we will make a second degree Taylor series expansion and set it to zero in order to obtain the root.

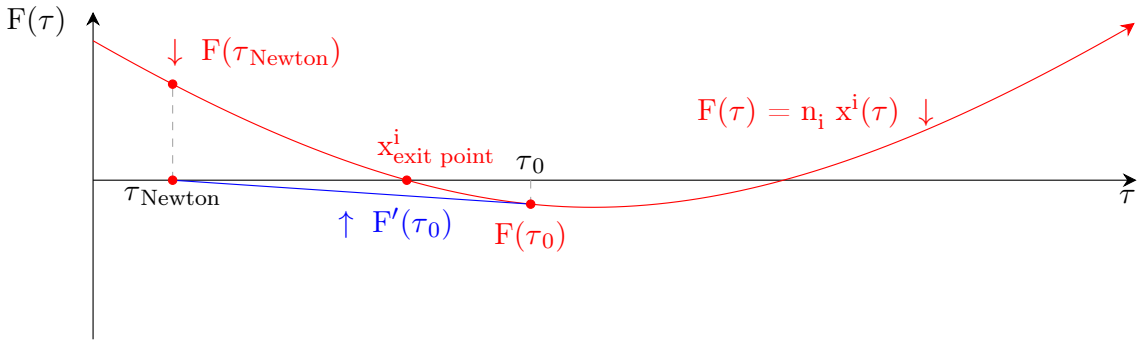
$$F(\tau) = F(\tau_0) + F'(\tau_0) \cdot (\tau - \tau_0) + \frac{1}{2} F''(\tau_0) \cdot (\tau - \tau_0)^2 \stackrel{!}{=} 0. \tag{5.37}$$

By setting τ_0 to 0 the problem turns into the quadratic equation

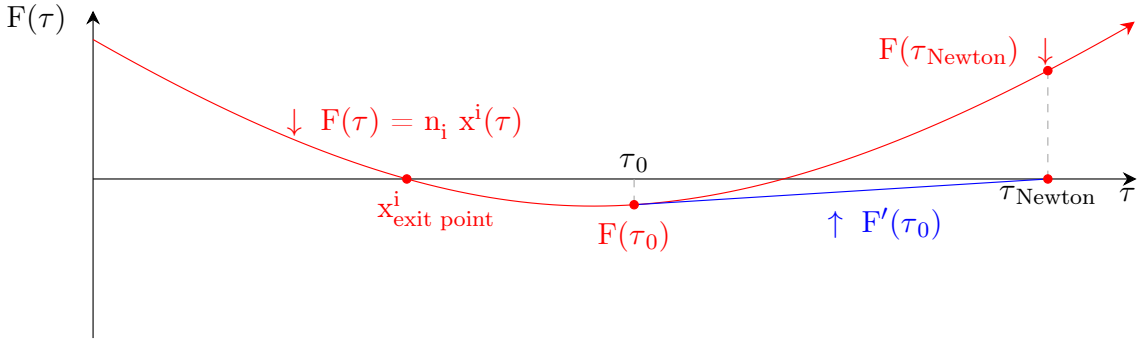
$$\underbrace{\frac{1}{2} F''(\tau_0)}_{a} \tau^2 + \underbrace{F'(\tau_0)}_{b} \tau + \underbrace{F(\tau_0)}_{c} = 0, \tag{5.38}$$

with the quadratic equation coefficients a, b and c.

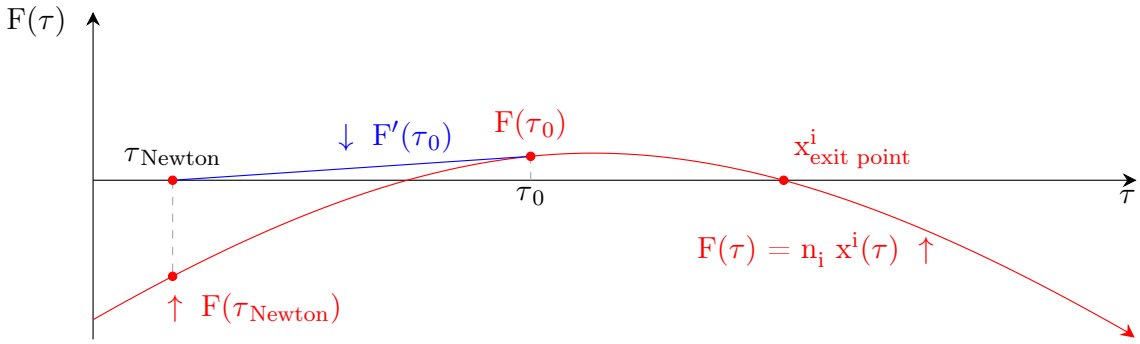
In order to obtain the quadratic equation coefficients b and c, one can use the same procedure that was used for Newton's method. Since we don't know the second derivative $F''(\tau_0)$ of the distance-function, we have to search for another way how to calculate the remaining coefficient a. Therefore we will make use of $F(\tau_{\text{Newton}})$, that was calculated at the previous iteration step and that had resulted in $F(\tau_{\text{Newton}}) > F(\tau_0)$.



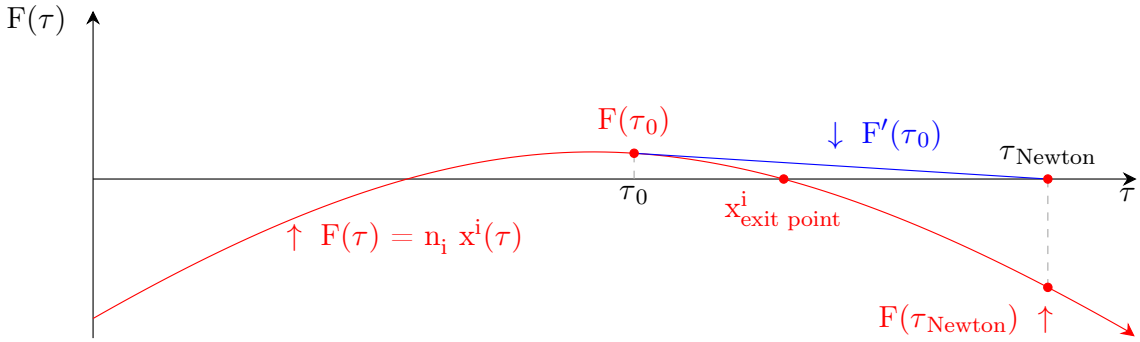
(a) τ_{Newton} is negative and $F(\tau_{\text{Newton}})$ is positive.



(b) τ_{Newton} is positive and $F(\tau_{\text{Newton}})$ is positive.



(c) τ_{Newton} is negative and $F(\tau_{\text{Newton}})$ is negative.



(d) τ_{Newton} is positive and $F(\tau_{\text{Newton}})$ is negative.

Figure 5.6: Quadratic behaviour of the distance in between the orbit and the assumed exit face:
There are four possibilities, why the use of Newton's method will result in a larger distance. Figure (a) and (b) show a positive sign of the quadratic function, whereas in figure (c) and (d) the sign of the quadratic function is negative. Furthermore it has to be distinguished, if the extrema of the quadratic function is overshot or undershot. The distinction in between these four cases results in different combinations of the sign of τ_{Newton} and $F(\tau_{\text{Newton}})$.

We will start by evaluating the quadratic equation at τ_{Newton}

$$F(\tau_{\text{Newton}}) = a \cdot \tau_{\text{Newton}}^2 + b \cdot \tau_{\text{Newton}} + c \tag{5.39}$$

and rearrange this equation to the missing quadratic equation coefficient a

$$a = \frac{F(\tau_{\text{Newton}}) - b \cdot \tau_{\text{Newton}} - c}{\tau_{\text{Newton}}^2}. \tag{5.40}$$

Furthermore we will express the first derivative $F'(\tau_0)$ by τ_{Newton} and $F(\tau_0)$

$$b = F'(\tau_0) = -\frac{F(\tau_0)}{\tau_{\text{Newton}}} = -\frac{c}{\tau_{\text{Newton}}}. \tag{5.41}$$

By inserting this expression into the equation above, the quadratic equation coefficient a simplifies to

$$a = \frac{F(\tau_{\text{Newton}})}{\tau_{\text{Newton}}^2}. \tag{5.42}$$

The quadratic equation for τ can then be solved by

$$\begin{aligned}
\tau_{1,2} &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\
&= \frac{\frac{F(\tau_0)}{\tau_{\text{Newton}}} \pm \sqrt{\left(\frac{F(\tau_0)}{\tau_{\text{Newton}}}\right)^2 - 4 \cdot \frac{F(\tau_{\text{Newton}})}{\tau_{\text{Newton}}^2} \cdot F(\tau_0)}}{2 \cdot \frac{F(\tau_{\text{Newton}})}{\tau_{\text{Newton}}^2}} \\
&= \frac{\tau_{\text{Newton}} \cdot \left(F(\tau_0) \pm \sqrt{F(\tau_0)^2 - 4 \cdot F(\tau_{\text{Newton}}) \cdot F(\tau_0)}\right)}{2 \cdot F(\tau_{\text{Newton}})} \\
&= \underbrace{\frac{\tau_{\text{Newton}}}{2 \cdot F(\tau_{\text{Newton}})}}_{h} \cdot \left(F(\tau_0) \pm \sqrt{F(\tau_0)^2 - 4 \cdot F(\tau_{\text{Newton}}) \cdot F(\tau_0)}\right)
\end{aligned} \tag{5.43}$$

One has to pay attention to the sign of τ_{Newton}, if one looks for a specific solution of the quadratic equation τ_1 or τ_2. In one of the simplification steps above, $\frac{1}{\tau_{\text{Newton}}^2}$ was excluded from the square root and as a result τ_{Newton} is subsequently multiplied with the square root. Thus, $\pm$ change sign, if τ_{Newton} is negative.

As one can see in figure 5.6, quadratic behaviour has to be separated into a first case $a > 0$ (figure 5.6 (a) and (b)), where the quadratic equation coefficient a is positive, and a second case $a < 0$ (figure 5.6 (c) and (d)), where a is negative. In the case that $a > 0$, the particle comes from a random point and leaves the tetrahedron on a quadratic trajectory, whereas in the case that $a < 0$, the particle entered the tetrahedron through the same face as it leaves the tetrahedron. The "left" root has to be chosen in the case that $a > 0$, since this is the one, where the particle exits the tetrahedron. The "right" root has to be chosen for $a < 0$ instead, because in this case, the "left" root is the one, where the particle entered. Both cases ($a > 0$ and $a < 0$) can be easily distinguished, by the sign of $F(\tau_{\text{Newton}})$. Furthermore, in both cases the solution is also dependent on the sign of τ_{Newton}.

One can save computation power if h is calculated first.

$$h = \frac{\tau_{\text{Newton}}}{2 \cdot F(\tau_{\text{Newton}})} \tag{5.44}$$

Before the solution τ for the quadratic approach is calculated, one has to examine, if

the solution even exists. Therefore the discriminant of the quadratic equation must be calculated

$$D = F(\tau_0)^2 - 4 \cdot F(\tau_{\text{Newton}}) \cdot F(\tau_0). \tag{5.45}$$

Table (5.1) shows the solutions of the quadratic equation for τ depending on the signs of $F(\tau_{\text{Newton}})$ and h.
If $D > 0$, a real solution exists and the appropriate solution of table (5.1) is calculated.

Table 5.1: Solutions of the quadratic equation for τ depending on the signs of $F(\tau_{\text{Newton}})$ and h

$\sigma(F(\tau_{\text{Newton}}))$ ... sign of $F(\tau_{\text{Newton}})$
$\sigma(h)$ ... sign of h

$\sigma(F(\tau_{\text{Newton}}))$	$\sigma(h)$	τ
+1	+1	$h \cdot \left(F(\tau_0) - \sqrt{D}\right)$
	-1	$h \cdot \left(F(\tau_0) + \sqrt{D}\right)$
-1	+1	$h \cdot \left(F(\tau_0) + \sqrt{D}\right)$
	-1	$h \cdot \left(F(\tau_0) - \sqrt{D}\right)$

Further on, a Runge-Kutta iteration is computed with this τ. Afterwards the whole convergence procedure is repeated: If the convergence criterion is not yet fulfilled, once again Newton's method is applied, and so on.

Whereas if $D < 0$, no real solution for the quadratic equation exists. This means that the orbit makes a turn close to the assumed exit face and that's why the particle doesn't leave the tetrahedron through this face. Thus, another face must include the point, through which the particle leaves the tetrahedron.

5.5.4 Calculate a new τ with a new analytical approximation

Consecutive iterations of Newton's method and/or a quadratic approach showed, that the orbit makes a turn close to the assumed face. Therefore the orbit didn't converge at the assumed exit face, that was proposed by the analytical approximation. The reason is, that the plane was initially proposed, by assuming that the particle will leave the tetrahedron through the face that can be reached in the smallest positive τ. But since τ was approximated, the assumption for the appropriate face could yield a wrong result. In such cases, where the orbit didn't converge, the assumption is obviously wrong.

In a next step a new analytical approximation for another face is made, excluding the previous face. Therefore one defines a logical array *allowed_faces* and sets the appropriate logical value to *false*, if the discriminant of the quadratic approach is negative. In a similar manner as in a previous section, one has to solve a quadratic equation, to obtain an approximation for τ. Once again the smallest τ can be found among the following solutions

$$\begin{aligned}
\tau^\alpha &= -\frac{2b^\alpha}{a^\alpha} \quad \text{for} \quad c=0, \quad \text{or} \\
\tau^\alpha &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \quad \text{for} \quad a^\alpha < 0, \quad \text{or} \\
\tau^\alpha &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \quad \text{for} \quad a^\alpha > 0 \,\text{and}\, b^\alpha < 0,
\end{aligned} \tag{5.46}$$

where a solution α is excluded, if the respective value in *allowed_faces* is *false*.

In comparison to the initial analytical approximation, the quadratic equation coefficients are different. After an unsuccessful attempt to converge at the previous face, the particle's location is now x_0^i, close to the forbidden face. Instead, at the initial analytical approximation, the particle's location was $x_{e.p.}^i$. Thus, the quadratic equation coefficients are

$$\begin{aligned}
a^\alpha &= n_i^\alpha \cdot a_4^i (a_4^4 \cdot v_{\parallel,e.p.} + b^4), \\
b^\alpha &= n_i^\alpha \cdot (a_4^i v_{\parallel,e.p.} + b^i) \quad \text{and} \\
c^\alpha &= n_i^\alpha \cdot x_0^i - d^\alpha.
\end{aligned} \tag{5.47}$$

Obviously, for the smallest positive τ among the solutions of eq. (5.46), the routine *rk4* is used to compute a Runge-Kutta calculation. The respective face α is the new assumed exit face. Afterwards the whole convergence procedure is repeated.

5.6 Error diagnostic and troubleshooting

5.6.1 Introduction to possible errors

In the previous section, the exit coordinates $x^i(\tau)$ of the particle and the parallel velocity $v_\parallel(\tau)$ at this exit point were searched. Every tetrahedron is spanned by four planes, that are described by four normal vectors n_i^α. An orbit was considered to be converged, when the distance in between the exit point $x^i(\tau)$ and the assumed plane

α was smaller than $dist_{min}$.

With this definition of convergence, it is not guaranteed, that the "convergence-point" is also the appropriate exit point. Furthermore it is even possible, that under some conditions, the particle's entry point is erroneously chosen as the "convergence-point". Therefore two additional conditions have to be introduced to guarantee, that the chosen "convergence-point" is also the real exit point:

1. The orbit must not pass through another plane, before it reaches the exit point.
2. The particle's velocity vector must point outwards the tetrahedron at the exit point.

In the following subsections possible errors of the convergence procedure are explained in detail and solutions to troubleshoot these errors are introduced.

5.6.2 Orbit converged at the wrong plane

Figure 5.7 depicts an example, how the analytical approximation for the appropriate exit face can fail. A wrong plane was proposed by the analytical approximation and the orbit converged at this proposed plane. To reach this convergence-point, it is necessary for the orbit to cross the right plane before.

To calculate the distance in between the particle's position and a certain plane α, one simply has to multiply the plane's normal vector n_i^α with the particle's coordinates x^i. Obviously, the plane that doesn't pass through the coordinate-origin has to be treated separately by subtracting d.

$$dist^\alpha = n_i^\alpha \cdot x^i - d^\alpha \tag{5.48}$$

Since all normal vectors are pointing inwards the tetrahedron, one can use the sign of this product to understand, if the particle is inside or outside the tetrahedron respective to this plane.

$$\begin{aligned} dist^\alpha > 0 \quad &\rightarrow \quad \text{Particle is inside the tetrahedron respective to plane } \alpha. \\ dist^\alpha < 0 \quad &\rightarrow \quad \text{Particle is outside the tetrahedron respective to plane } \alpha. \end{aligned} \tag{5.49}$$

When an orbit is considered to be converged at a plane, the sign of the distance in between particle's coordinates and respective plane might be plus or minus. Therefore

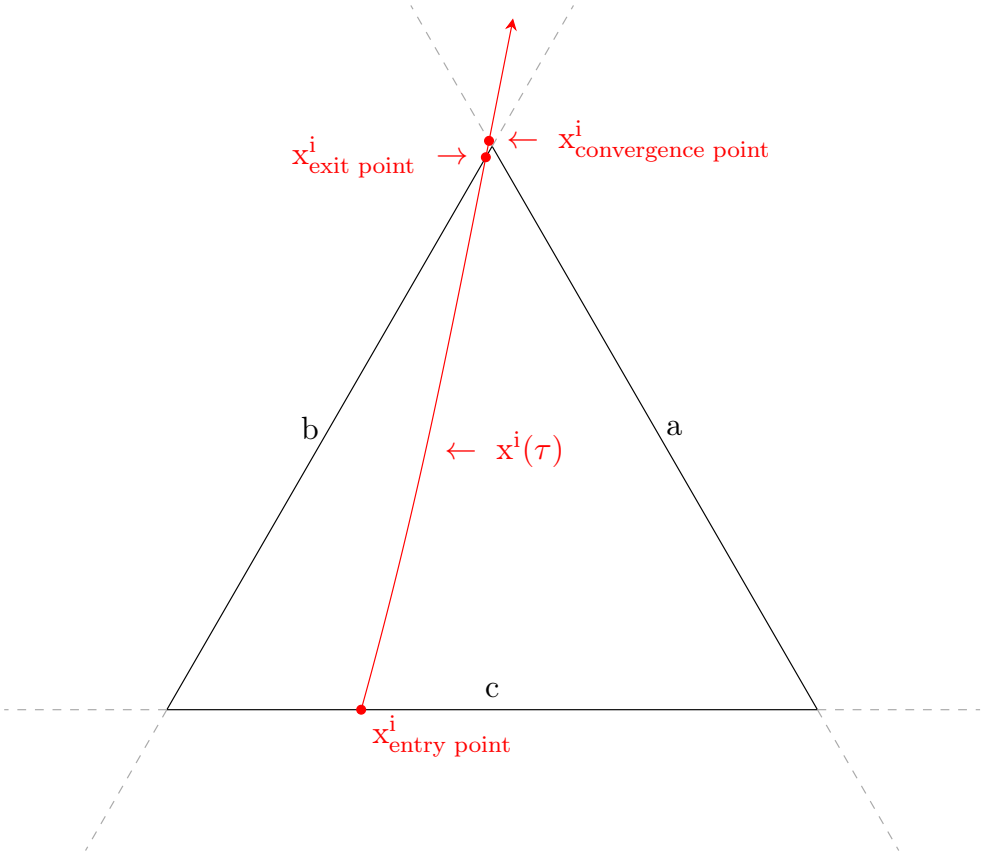


Figure 5.7: Orbit converged at the wrong plane:
The particle's exit point is very close to one of the tetrahedrons vertices. Initially the analytical approximation proposed, that the particle will leave the tetrahedron through face *a*, but the real orbit passes through face *b*. Thus, the orbit converged at the wrong plane.

only the sign of the distance in between the particle's coordinates and the other three planes has to be examined. In the Fortran code the four planes have the indices $\alpha = 1, \ldots, 4$. To examine only the three other planes, if the plane with a certain index must not be examined, the following code is very helpful,

```
do i=1,3
  j=iface_new+i
  j=modulo(j-1,4)+1
end do
```

where *iface_new* is the index of the plane that must not be considered, and j adopts the remaining three indices.

If $dist^\alpha > 0$ for all three planes α, then it is guaranteed, that the right plane was chosen to be the plane, through which the particle leaves the tetrahedron. Although

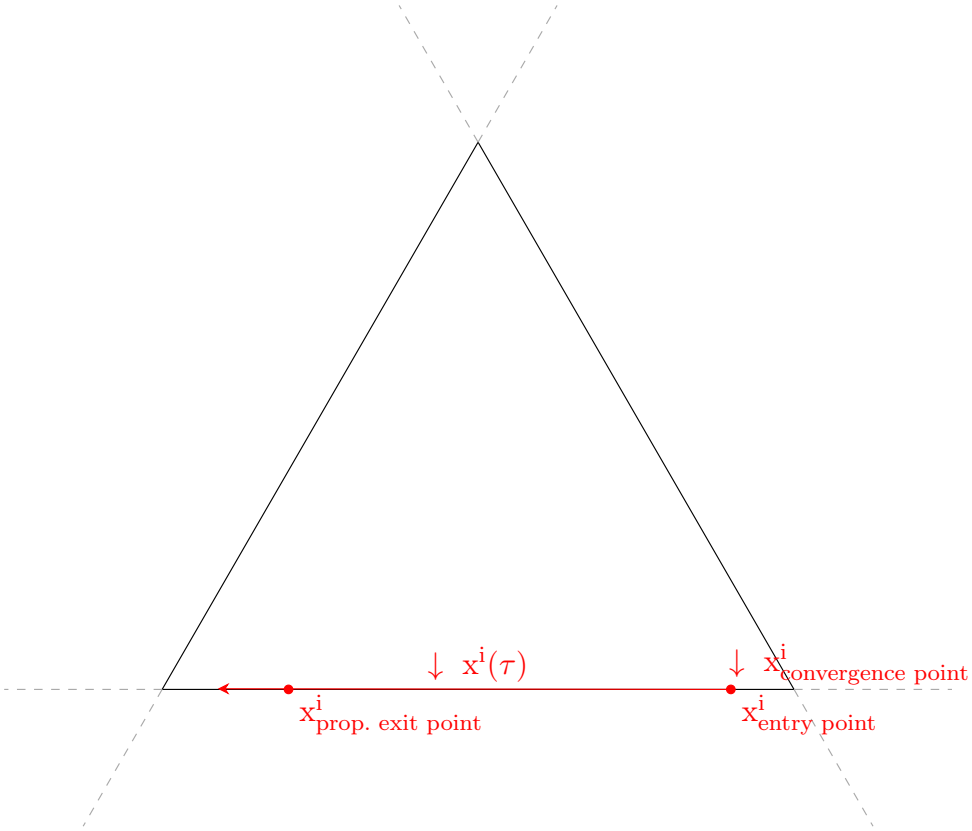
it is not guaranteed yet, that the appropriate exit point was chosen.

If $dist^{\alpha} < 0$ for only one plane out of the three planes, the chosen exit face is wrong, because the particle is outside the tetrahedron respective to this plane. That means that the orbit has crossed this plane first, before it converged at the assumed plane. Therefore troubleshooting for such a case has to be introduced.

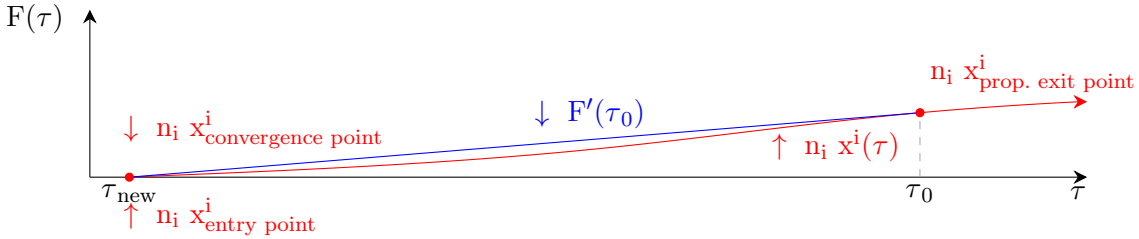
5.6.3 The particle's velocity vector points inwards the tetrahedron at the exit point (positive $v_{\mathbf{norm}}$)

There exists a special case, in which the orbit converges at the right plane, but the convergence-point is not the real exit point of the orbit. Figure 5.8 depicts a case, in which the orbit is almost tangential to a plane. According to the analytical approximation the particle should leave the tetrahedron through the respective face at a certain τ. When the coordinates of the real orbit at τ are calculated with the Runge-Kutta method, it turns out, that this particle doesn't leave the tetrahedron at this position. The reason is, that either τ was approximated too small for the assumed exit point, or in reality the particle leaves through another face.
Furthermore in this special case, the normal velocity $v_{\mathrm{norm}}(\tau)$ after the Runge-Kutta iteration points inwards the tetrahedron, because of the almost tangential behaviour. The subsequent convergence procedure will try to use Newton's method to decrease the distance in between the particle's coordinates and the respective plane. Since the orbit is almost tangential, the "new" point, that is proposed by Newton's method, is very close to the entry point and indeed the distance decreases. That's why this iteration step seems useful for to the convergence procedure. Finally, when the convergence criterion is fulfilled, the orbit converges "at/around" the coordinates of the entry point. Obviously the normal velocity $v_{\mathrm{norm,e.p.}}$ at the entry point is positive, since the particle moves inwards the tetrahedron at this position. To identify such a case, one has to calculate the normal velocity $v_{\mathrm{norm}}(\tau)$ of the particle at the exit point and examine, if it is positive or negative.

If it can be guaranteed, that the right exit face is chosen (see previous section) and if additionally the normal velocity at the convergence point is negative (the velocity vector points outwards the tetrahedron), one can assume to have found the right exit point.
Whereas, if the velocity vector points inwards the tetrahedron (positive v_{norm}) at the convergence-point, the right exit point has not been found. Thus, troubleshooting for such a case has to be introduced as well.



(a) The orbit is almost tangential to one of the cell's faces.



(b) Newton's method finds a convergence point, that is not the right exit point.

Figure 5.8: In the depicted special case (a), the orbit is almost tangential to one of the cell's faces. Furthermore the analytical approximation proposed a wrong exit point. Since the orbit has not turned towards the respective face, the derivative at the proposed exit point is positive. That's why Newton's method (b) finds a convergence point that is close to the particle's entry point into the tetrahedron. Obviously the normal velocity at this convergence point is positive.

5.6.4 Troubleshooting for orbits with a wrong exit point

As it was described in the previous sections, the convergence-point doesn't match with the right exit point, if either the orbit converged at the wrong plane, or v_{norm} is positive at the exit point. If the convergence procedure yielded a wrong result, the following steps have to be applied. The detection of errors and the troubleshooting

itself is part of the same code.

First of all one starts with a logical array *allowed_faces* with four values, where all values are set *true* by default. If it turns out during the error detection process, that the exit through a certain plane is not allowed, because another plane has been crossed before, the respective value of this logical array is set to *false*. Another logical variable *finish* is introduced, that is by default *true*.

The whole detection and troubleshooting code consists of a loop over all four planes. First of all, it is examined, if the orbit converged at the right plane. (The distances in between the particle's coordinates and the other three planes are positive). If the orbit didn't converge at the right plane, *finish* and the respective value of *allowed_faces* are set to *false*. Further on, in case of proper convergence, the normal velocity at the exit point is examined. Once again *finish* and the respective value of *allowed_faces* are set to *false*, if v_{norm} is positive at the exit point.
If no errors were detected, namely *finish* is *true*, the loop is immediately exited and the right exit point has been found.

Orbit converged at the wrong plane:
For the wrongly converged orbits, a new exit face is proposed. Since the distance in between the particle's coordinates and another plane is negative, if a certain face had been crossed before, the respective face is assumed to be the right one. But this face can only be taken into consideration, if the respective value of *allowed_faces* is *true*, otherwise another allowed face is chosen.

Normal velocity at the convergence point is positive:
The normal velocity at the convergence point v_{norm} can be positive, if the orbit is almost tangential to a certain plane. The orbit has the possibility, either to make a slight turn and to exit through the plane, to that it is almost tangential, or to exit through another face.
That's why a new analytical approximation is made, in which forbidden faces are excluded. One uses the routine *rk4* to push the orbit to the new assumed face. If no analytical approximation exists, one assumes, that the orbit turns close to the face and exits through the same face as it entered. In this case, one proceeds as explained in subsection (5.4).

Depending on which error occurred, either the orbit has been pushed to another

face, or the coordinates $x^i(\tau)$ remain the same and just the assumed exit face was changed. As shown in subsection (5.5), one continues with the usual convergence procedure: Newton's method or a quadratic approach. If the discriminant in the quadratic equation of the quadratic approach is negative, meaning that the particle doesn't exit through this face, the respective value of *allowed_faces* is set to *false* and this loop-step is cycled.
If the orbit converged at the new assumed face, a new iteration in the loop is started, beginning with the error detection.

At some step in the loop, the right exit point might have been found and one exits the loop, since *finish* remains *true*. In such a case, the described troubleshooting procedure was successful. Nevertheless, there is still the possibility, that this procedure fails. If all four values of *allowed_faces* become *false*, the troubleshooting procedure was not able to find the proper exit point.

5.6.5 Troubleshooting failed: Last line of defense

Up to this point, the aim of the explained algorithm was to be as computationally inexpensive as possible. It turns out, that there are special cases, in which the inexpensive algorithm fails, and one has to take into account to use a computationally more expensive code. This "last line of defense" should be used in as few cases as possible, but it guarantees to find the right exit point.

Example for failed troubleshooting procedure:
For instance let us consider a case, where the initial analytical approximation yielded a wrong exit point. Further on the error could be detected, but not solved.
As one can see in figure 5.9, the particle leaves the tetrahedron through face a close to one of its vertices and makes a turn right after the exit face had been passed. After the orbit's turn, the plane b is crossed, but outside the tetrahedron. The initial analytical approximation proposed plane b as the exit face. The convergence procedure was successful and the orbit converged at this particular plane (convergence point 1). Since the convergence-point was outside the tetrahedron, the error was detected and the plane, that was crossed by the orbit first (plane a), was considered as a new proposal for the exit face.
The troubleshooting procedure then applies Newton's method or the quadratic approach until the orbit converges at the assumed plane. In our case Newton's method seems to be useful, since the distance decreases. As described above, if the orbit converged at the wrong plane, the coordinates of the convergence-point $x^i(\tau)$ are

considered as "starting"-coordinates x_0^i for the troubleshooting procedure and just another face is proposed for a consecutive convergence procedure. What makes this case so special is, that the orbit had turned before it converged at the second plane (b). When Newton's method is applied from the coordinates of convergence point 1, for the plane, that had been crossed first (b), another "wrong" convergence point 2 is found at the proper face. This particular convergence point is in fact the orbit's third crossing with a plane. Since this convergence-point is also outside the tetrahedron, the troubleshooting procedure will consider this face as not allowed. In fact, the proper face has been found, but not the right exit point. As a result all faces will be forbidden and therefore the troubleshooting fails.

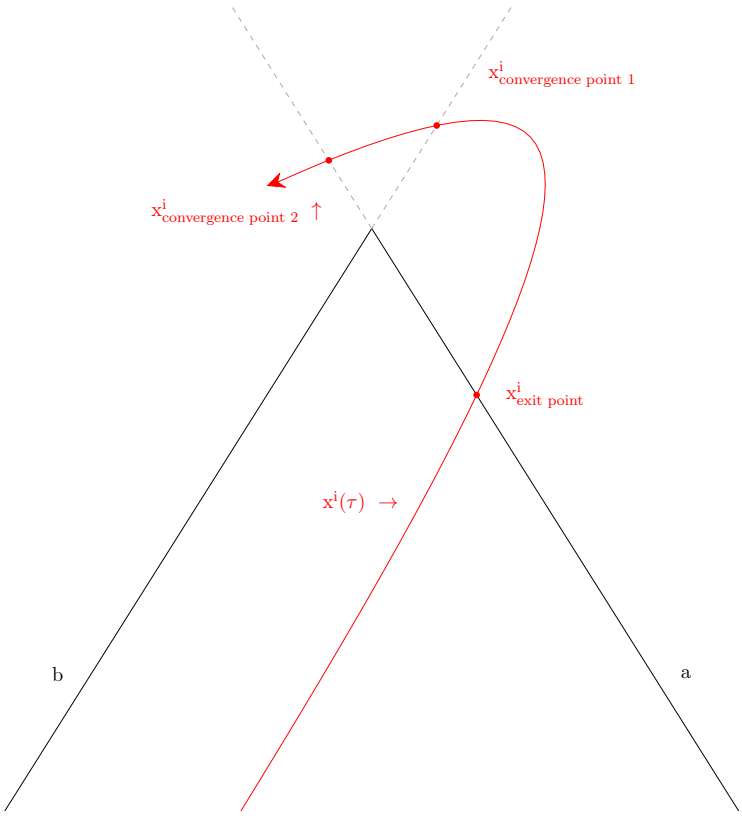


Figure 5.9: The standard troubleshooting procedure fails:
Convergence point 1 is detected as a wrong exit point. The following convergence procedure (Newton's method) finds convergence point 2, which is as well a wrong exit point. The reason, why the standard troubleshooting procedure fails, is that the orbit makes a turn right after the proper exit face had been passed, but before the orbit converged at the wrong plane.

Functionality principle of the last line of defense code:
First of all the orbit that has been calculated up to this point, is considered to be useless, since the convergence-point can be at a random location. That's why, one starts from the beginning with the orbit calculation, by setting the coordinates of the orbit to the particle's entry point into the tetrahedron $x^i_{\text{e.p.}}(\tau_0)$.
The functionality principle, that is used further on for the last line of defense code, is the bisection method. As a first task, it is necessary to integrate the orbit, until the tetrahedron is left by the particle. For that reason an analytical approximation like in the very beginning is made. The only difference is, that almost tangential cases, as explained in subsection (5.6.3), are excluded from the possible results. This increases the probability, that the orbit will cross a plane.
For this first guess of τ one Runge-Kutta iteration is calculated and the distance to the four planes is evaluated. In most of the cases, the particle should have left the tetrahedron, but still there are some special cases, where the distance to all four planes is positive, meaning that the particle is still inside the tetrahedron. This can happen, if the particle didn't leave through the face, that was proposed by the analytical approximation. To assure, that the particle leaves the tetrahedron, a loop is used with consecutive Runge-Kutta iterations, where the guessed τ is doubled for every iteration, until the distance to one of the four planes becomes negative.

Afterwards, it can be guaranteed, that the particle left the tetrahedron, the actual bisection method begins. The last Runge-Kutta iteration for τ, integrated the orbit from a point inside the tetrahedron to a point outside the tetrahedron. By setting

$$\tau_{\text{new}} = -\frac{\tau}{2} \tag{5.50}$$

for the next Runge-Kutta iteration, the orbit is integrated towards the inside of the tetrahedron. As a next step, one has to examine, if the new point is inside or outside the tetrahedron. The location of the particle, if inside or outside, determines the sign of the consecutive iteration step

$$\begin{aligned} \underset{\alpha}{\text{all}}\,(\text{dist}^\alpha) > 0 \qquad &\rightarrow \qquad \tau_{\text{new}} = \frac{|\tau|}{2} \qquad \text{(particle is inside)} \\ \underset{\alpha}{\text{any}}\,(\text{dist}^\alpha) < 0 \qquad &\rightarrow \qquad \tau_{\text{new}} = -\frac{|\tau|}{2} \qquad \text{(particle is outside)}, \end{aligned} \tag{5.51}$$

where $dist^\alpha$ is the distance to the four planes ($\text{dist} = n^\alpha_i x^i(\tau)$).
This procedure is repeated in a loop, until only one of the four elements of $dist^\alpha$ is negative, namely the particle is only outside the tetrahedron respective to one of the planes.

The usual convergence procedure with Newton's method or the quadratic approach, as explained in subsection (5.5), is used to find the right exit point up to a certain accuracy. Finally, one must approve, that the particle is not outside the respective other three faces and that the normal velocity vector points outside the tetrahedron, otherwise the last line of defense as a troubleshooting method has failed.

5.7 Final processing

After a successful numerical integration of the particle orbit and convergence of the trajectory at the right exit point, the particle's coordinates $x^i(\tau)$ and parallel velocity $v_\parallel(\tau)$ are given. For the functionality of the algorithm, also the perpendicular velocity $v_\perp(\tau)$ at the exit point is needed. Since $J_\perp = \frac{mv_\perp^2}{2\omega_c}$ is by definition of the integrator an invariant of motion and the magnetic field modulus B is a linear function inside the tetrahedron, the perpendicular velocity $v_\perp(x^i(\tau))$ at the exit point can be obtained by

$$v_\perp(x^i(\tau)) = \sqrt{\frac{2J_\perp \omega_c(x^i(\tau))}{m}} = \sqrt{\frac{2J_\perp eB(x^i(\tau))}{m^2 c}}. \tag{5.52}$$

Chapter 6

Evaluation of particle orbits calculated by 3DGeoInt

The following chapter is loosely based on the books [1], [15] and [16].

6.1 Physical set-up for orbit integration

For the calculations the Gaussian-cgs units are used.

As test particles for the orbit integration we choose nuclei of fully ionized Helium-atoms (α-particles). The mass of one α-particle consists of twice the proton mass ($m_p = 1.6726 \cdot 10^{-24}$ g) and twice the neutron mass ($m_n = 1.6749 \cdot 10^{-24}$ g), thus $m_\alpha = 6.6446 \cdot 10^{-24}$ g. Furthermore the speed of light $c = 2.9979 \cdot 10^{10}$ cm/s and the electron charge $e = 4.8032 \cdot 10^{-10}$ Fr are needed as constants.

The initial kinetic energy ($E_{kin} = \frac{m_\alpha v^2}{2}$) of an α-particle is set to 3 keV, which is $4.8066 \cdot 10^{-9}$ erg. Therefore, the particle's absolute velocity is set to

$$v = \sqrt{\frac{E_{kin}}{m_p + m_n}}. \tag{6.1}$$

To clearly specify the starting condition for the orbit integrator, also the pitch parameter at the starting point

$$\lambda_0 = \cos(\chi_0) = \frac{v_{\parallel,0}}{v_0}, \tag{6.2}$$

where χ_0 is the pitch angle, $v_{\parallel,0}$ the parallel velocity and v_0 the absolute velocity at the starting point, must be defined for every calculation.

6.2 Qualitative evaluation of orbits for axisymmetric devices

For a first qualitative evaluation of the orbits of the three-dimensionally geometric integrator *3DGeoInt*, the field quantities from the equilibrium in the EFIT-format from the tokamak *ASDEX Upgrade* [18] (shot *g26884*, at 4300 ms) are used.

The magnetic field of an axisymmetric fusion device consists of two axisymmetric components: A strong toroidal magnetic field generated by external coils and a much weaker poloidal field generated by the toroidal plasma current. In the following configuration the magnetic field is constant along the toroidal φ-axis.
The electrostatic potential Φ can be manually set to an arbitrary function. In our case, Φ is a linear function of the magnetic flux Ψ

$$\Phi = C \cdot \Psi, \tag{6.3}$$

where C is a constant. The constant C is chosen in such a way, that the electrostatic potential energy is in the order of the initial kinetic energy of the particle. For our calculation, the constant C is set to $\frac{1}{3 \cdot 10^7}$, since Ψ is in the order of 10^7.
With this choice of a linear dependence of Φ on Ψ, the electric field $\mathbf{E} = -\nabla\Phi$ is always perpendicular to the magnetic flux surface, and thus, $\mathbf{E} \times \mathbf{B}$-drifts occur only along the magnetic flux surface.

In the following sections the kinetic behaviour of the orbits is shown, by various set-ups of the pitch angle χ_0 at the starting point and different functions for the electrostatic potential Φ.

6.2.1 Passing particle

As a first example, one can see the orbit of a passing particle in three dimensions in figure 6.1. In an axisymmetric fusion device, the passing particles are almost confined to magnetic flux surfaces. If one makes a cut in the three dimensional space along a constant toroidal angle φ, one obtains a Poincaré cut. The orbits in the poloidal plane $\varphi = 0$ stay exactly closed for axisymmetric devices. Figure 6.2 depicts the Poincaré-cut for $\varphi = 0$. As one can easily see, the orbit of the particle is confined.

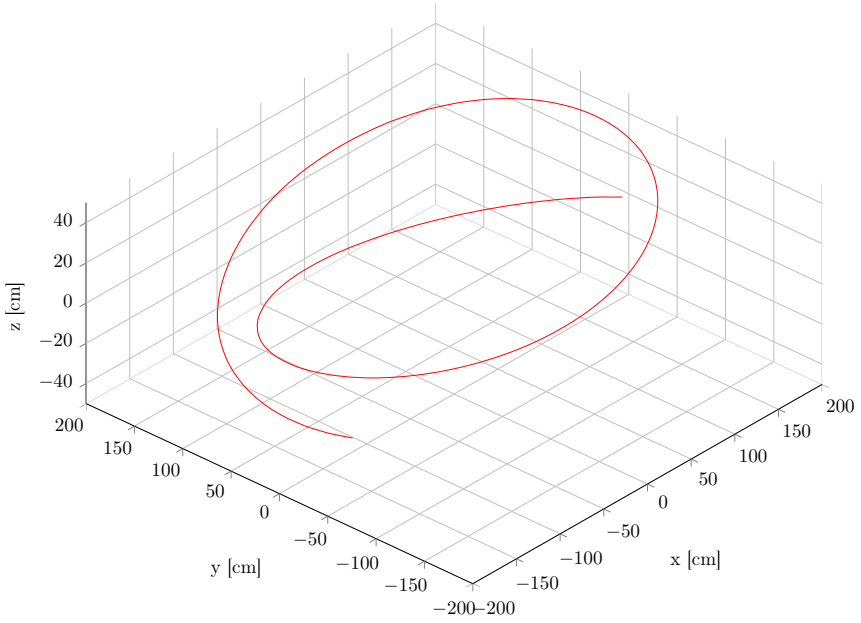


Figure 6.1: Example of a passing particle:
Starting position: $x^i = (R = 192.97 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$, pitch angle: $\chi_0 = 2.1$, electrostatic potential: $\Phi = 0$
The pitch angle at the starting position is chosen in such a way, that no magnetic mirror effect occurs.

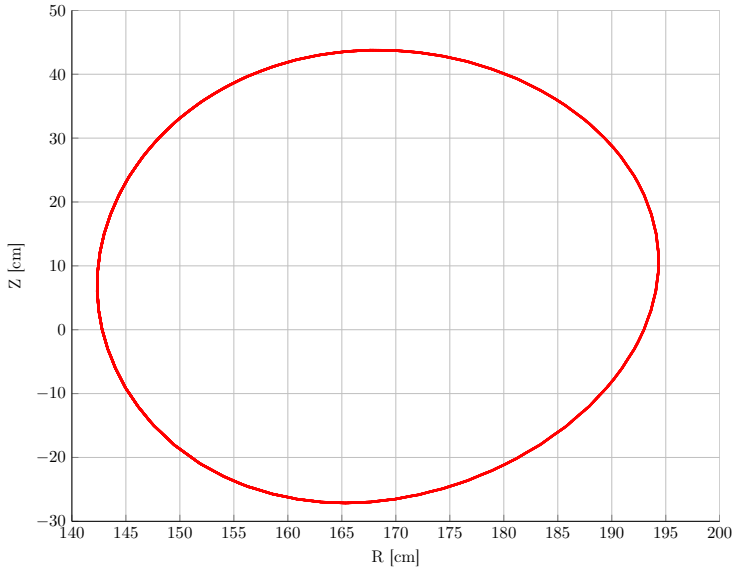


Figure 6.2: Poincaré-cut of a passing particle at $\varphi = 0$:
Starting position: $x^i = (R = 192.97 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$, pitch angle: $\chi_0 = 2.1$, electrostatic potential: $\Phi = 0$
The plot clearly depicts that for an axisymmetric system the orbit is exactly closed in the poloidal plane.

6.2.2 Trapped particle

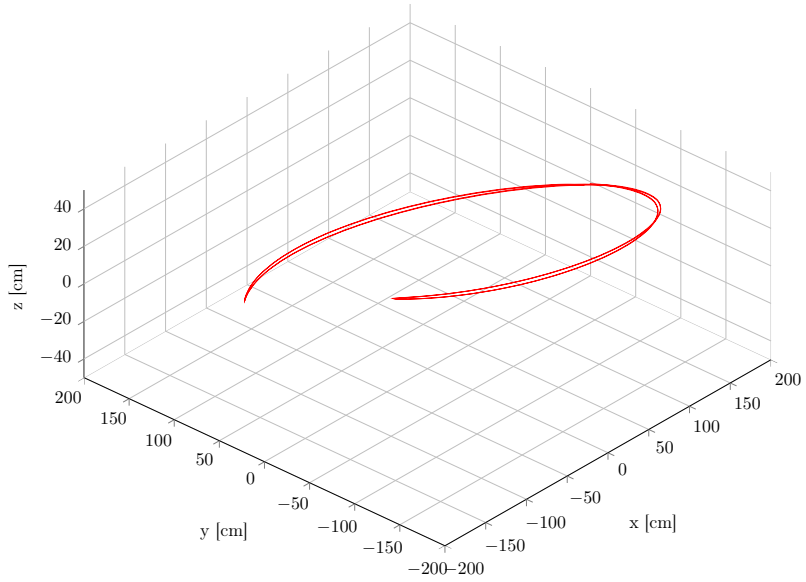


Figure 6.3: Example of a trapped particle:
Starting position: $x^i = (R = 192.97 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$, pitch angle: $\chi_0 = 2.0$, electrostatic potential: $\Phi = 0$
The pitch angle at the starting position is chosen in such a way, that the magnetic mirror effect occurs.

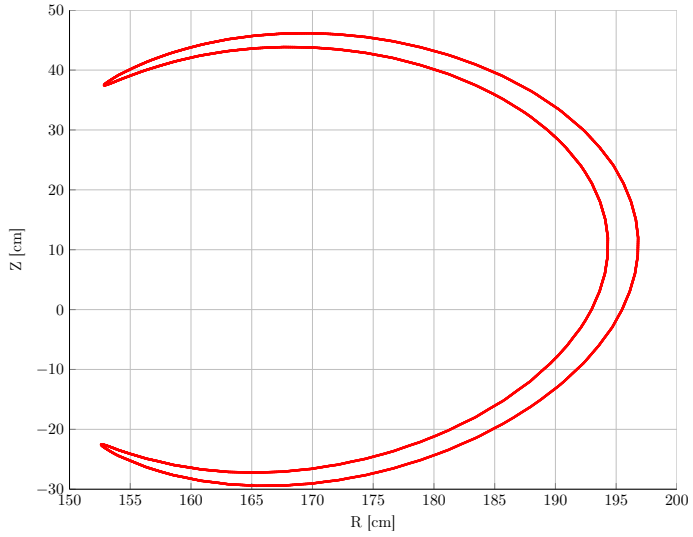


Figure 6.4: Poincaré-cut of a trapped particle at $\varphi = 0$:
Starting position: $x^i = (R = 192.97 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$, pitch angle: $\chi_0 = 2.0$, electrostatic potential: $\Phi = 0$
The plot clearly depicts that for an axisymmetric system the orbit is exactly closed in the poloidal plane.

One can see the three dimensional plot of the orbit of a trapped particle in figure 6.3. Due to the magnetic mirror effect, $v_{\|}$ changes sign, the particle is reflected and the orbit has the shape of a banana. Furthermore, the banana precesses toroidally along the φ-axis, because of the magnetic and electric drifts. Figure 6.4 shows the Poincaré-cut of a trapped particle at $\varphi = 0$. The Poincaré-cut of the precessing banana is shaped like a banana as well. In an axisymmetric fusion device the orbits are closed.

By introducing an electrostatic potential, the toroidal banana precession frequency Ω_{tor} is increased (see section 6.3 and [19]). One can see a strongly precessing banana in the three dimensional plot of a trapped particle in figure 6.5.

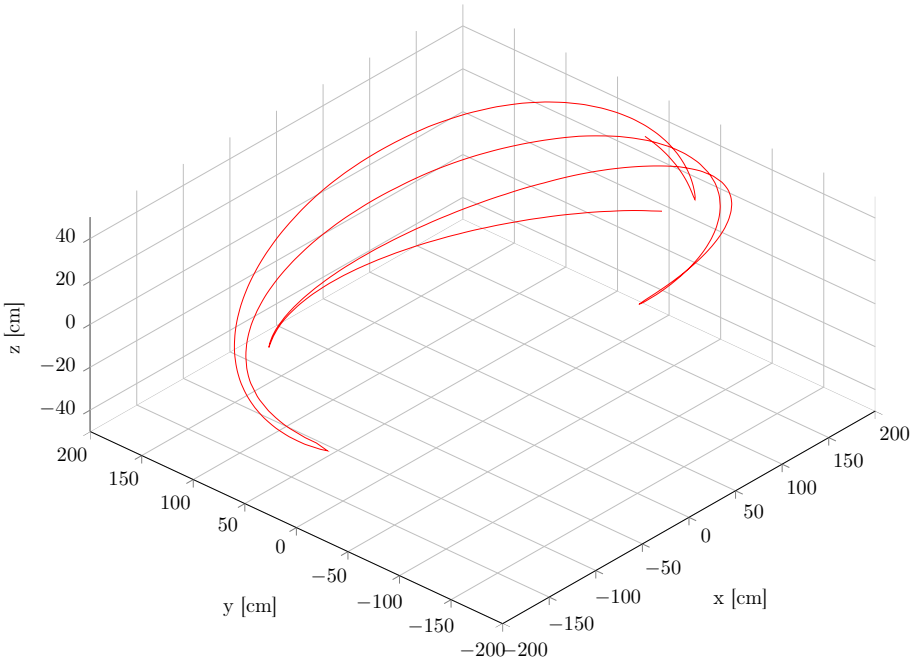


Figure 6.5: Toroidally precessing banana orbit:
The starting condition of the particle is the same as in figure 6.3, but additionally also an electrostatic potential $\Phi = 7 \cdot C \cdot \Psi$ is turned on. This causes an increase of the toroidal banana precession frequency Ω_{tor}.

6.2.3 Trapped and passing particles due to the pitch-parameter

In the previous plots, the orbits of trapped and passing particles were depicted. If a particle is trapped or can pass, depends on its ratio of $v_\perp$ to $v_\parallel$. Thus, the pitch parameter at the starting position λ_0 determines the kind of orbit. Figure 6.6 depicts 4 orbits of particles with exactly the same starting positions and the same total energy. The electrostatic potential is set to zero for this calculation. One can see 3 banana orbits and 1 passing orbit, only differing on the pitch parameter at the starting position.

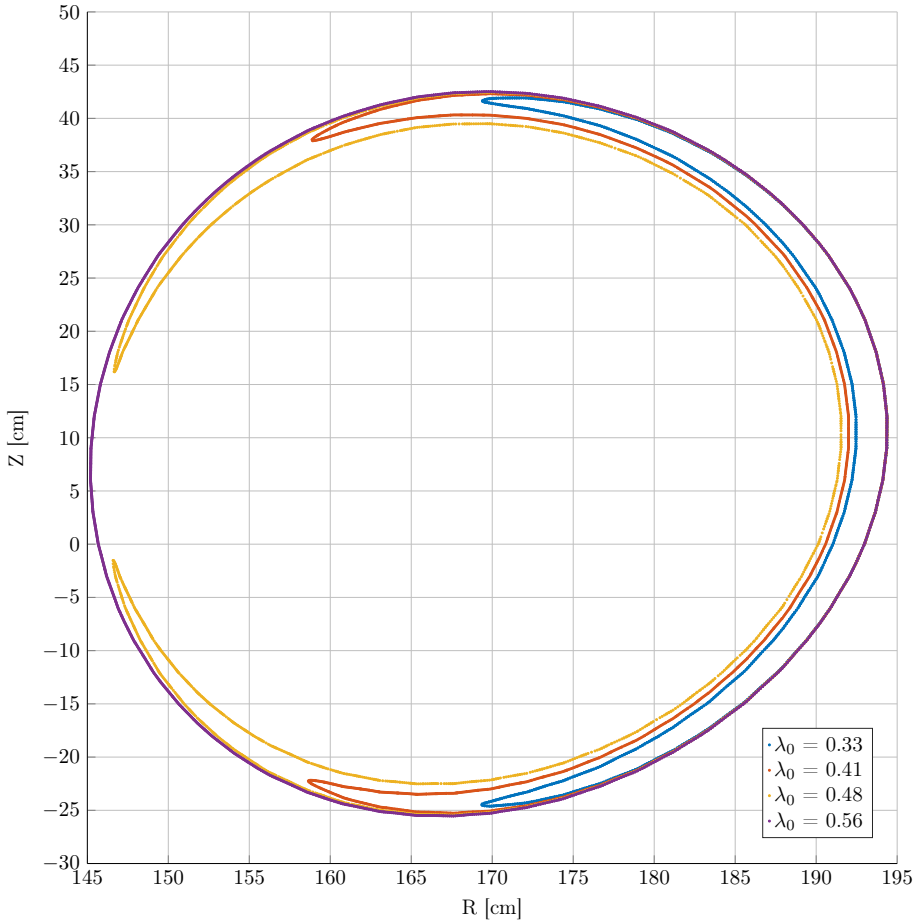


Figure 6.6: Trapped and passing particles:
Starting position: $x^i = (R = 193\text{ cm}, \varphi = 0, Z = 0\text{ cm})$, electrostatic potential $\Phi = 0$
The 4 particles have exactly the same starting position and the same total energy, they only differ on pitch parameter at the starting position λ_0.

6.2.4 Magnetic guiding center drifts

The drift velocity of the guiding center $\mathbf{v}_d$ for zero electric field can be written as

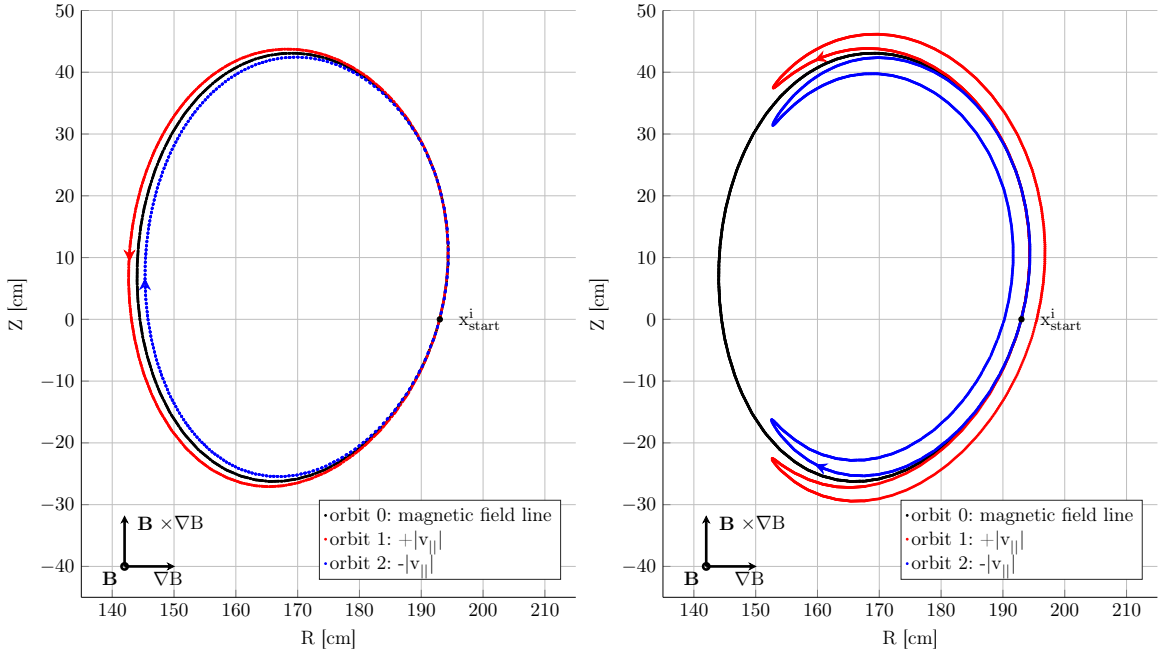
$$\mathbf{v}_d = \frac{c\mu}{e}\frac{\mathbf{B}\times\nabla B}{B^2} + \frac{cmv_\parallel^2}{eR_c^2}\frac{\mathbf{R}_c\times\mathbf{B}}{B^2}, \tag{6.4}$$

where $\mathbf{R}_c$ is the curvature radius of the magnetic field line. The particular drifts of the guiding center are the ∇B *drift* and the *curvature drift*. Both drifts are inversely proportional to the squared magnetic field modulus B.
Furthermore in force-free magnetic fields ($\mathbf{j} \parallel \mathbf{B}$) the ∇B *drift* and the *curvature drift* can be combined to

$$\frac{c\mu}{e}\frac{\mathbf{B}\times\nabla B}{B^2} + \frac{cmv_\parallel^2}{eR_c^2}\frac{\mathbf{R}_c\times\mathbf{B}}{B^2} = \frac{c\mu}{e}\left(1+\frac{2v_\parallel^2}{v_\perp^2}\right)\frac{\mathbf{B}\times\nabla B}{B^2} \tag{6.5}$$

If one increases manually the magnetic field modulus B in the fusion device, the magnetic guiding center drifts decrease, and the guiding center follows the magnetic field line.

Figure 6.7 shows the orbits of (a) two passing and (b) two trapped particles with the same starting position. For orbit 0, the magnetic field modulus B was manually increased up to an order, so that the magnetic drifts disappear and the particle follows the magnetic field line. Orbits 1 and 2 feel the unchanged magnetic field, but they differ in the sign of $v_\parallel$. One can see the influence of the magnetic guiding center drifts on the particle orbits. Depending on the sign of $v_\parallel$, the particle drifts either inside or outside the magnetic flux surfaces.



(a) Passing particle

(b) Trapped particle

Figure 6.7: Influence of the magnetic guiding center drifts on the particle orbits:

(a) Passing particle:
Starting position: $x^i = (R = 192.97$ cm, $\varphi = 0$, $Z = 0$ cm), pitch angle orbit 1: $\chi_{0,1} = 0.9$, pitch angle orbit 2: $\chi_{0,2} = 2.2$, electrostatic potential: $\Phi = 0$

(b) Trapped particle :
Starting position: $x^i = (R = 192.97$ cm, $\varphi = 0$, $Z = 0$ cm), pitch angle orbit 1: $\chi_{0,1} = 1.1$, pitch angle orbit 2: $\chi_{0,2} = 2.0$, electrostatic potential: $\Phi = 0$

To compute the orbit 0 along the magnetic field line, the magnetic field modulus was manually increased up to an order until the magnetic drifts vanish. The sign of $v_\parallel$ at the starting position defines, if the particle drifts inside (blue) or outside (red) the magnetic flux surface.

6.2.5 Trapped-passing particle boundary due to the electrostatic potential Φ

With the introduction of an electrostatic potential Φ that is a linear function of the magnetic flux Ψ, obviously also the electrostatic potential energy $e\Phi$ is linear in Ψ. If a particle drifts through flux surfaces, e.g. due to the magnetic guiding center drifts, the electrostatic potential energy $e\Phi$ increases or decreases. Since the particle's total energy w is conserved, the kinetic energy must change, because of the cross field drift.

Figure 6.8 depicts the influence of the electrostatic potential Φ on the particle's orbit. At a certain electrostatic potential Φ, the loss of parallel kinetic energy is high enough to cause the magnetic mirror effect. The particle's parallel velocity $v_{\parallel}$ changes sign, meaning it is reflected. A former passing particle becomes a trapped particle.

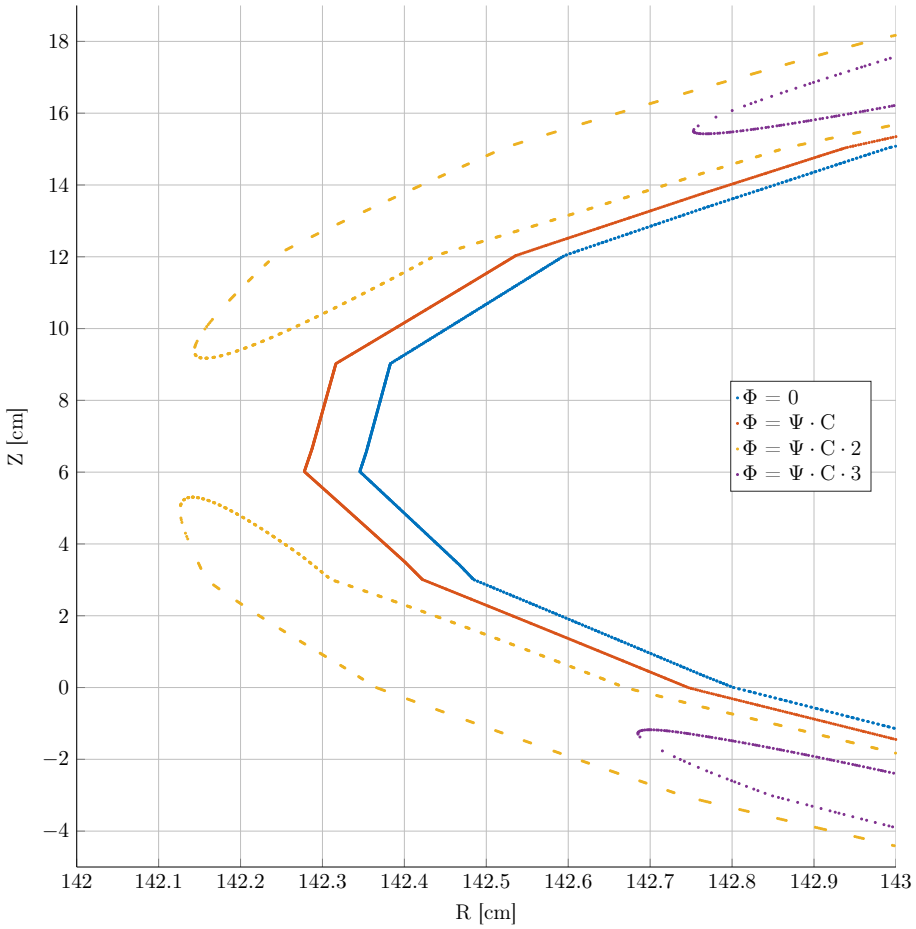


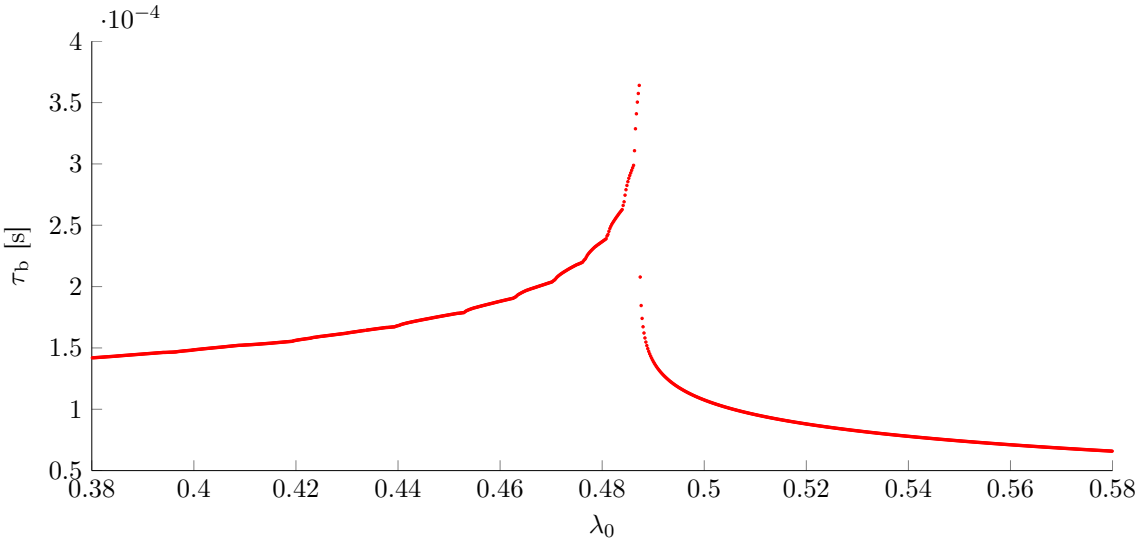
Figure 6.8: Influence of the the electrostatic potential Φ on the particle's orbit: Starting position: $x^i = (R = 192.97$ cm, $\varphi = 0$, $Z = 0$ cm), pitch angle: $\chi_0 = 2.1$ The electrostatic potential Φ is a linear function of the magnetic flux Ψ. The figure depicts the Poincaré cut for four particle orbits with the same starting position and same χ_0. Due to cross field drifts, the particle's kinetic energy decrease, while the electrostatic potential energy increases. A passing particle becomes a trapped particle at a certain electrostatic potential Φ, when $v_{\parallel}$ changes sign.

6.3 Bounce time τ_b and bounce frequency ω_b

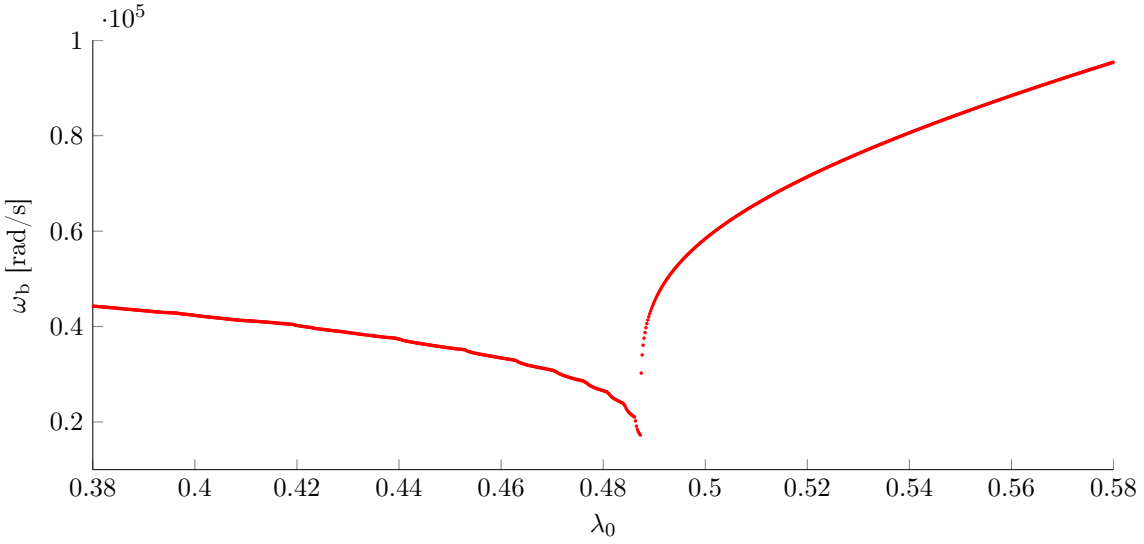
Universally, one can define the bounce time as the return time to the point with exactly the same (R,Z) coordinates.

In order to calculate the bounce time of a trapped particle, one follows its trajectory from a certain location (R,φ,Z), until the particle passes both banana-tips and reaches the same (R,Z) coordinates. In the meantime, the banana precessed in toroidal direction, as one can see in figure 6.5, meaning the location differs in φ.
For a passing particle the bounce time is the period of a poloidal motion. The displacement during the bounce time in the toroidal direction for passing particles is approximately $2 \cdot \pi \cdot q$, where q is the safety factor. The exact toroidal displacement of passing particle differs from this number, but only in a small term linear in the Larmor radius. This small term is of the same order (and has the same origin, due to the cross-field guiding center drift) as the displacement for trapped particles.

Figure 6.9a depicts the bounce time τ_b versus the pitch parameter λ_0 at the starting position. One can see on the left side of the plot, the bounce time for trapped particles. As λ_0 increases, also the bounce time τ_b increases, due to the larger size of the banana (see figure 6.6). It is clearly visible, that at the transition from a trapped particle to a passing particle, the bounce time has a singularity, meaning that the particle stands still at the point of transition. On the right side of figure 6.9a the bounce time for passing particles is depicted. By contrast, the bounce time increases, with decreasing pitch parameter, due to the particle's parallel deceleration when passing the "magnetic mirror".
Figure 6.9b shows the bounce frequency, which is calculated by $\omega_b = \frac{2\pi}{\tau_b}$.
In both figures one can see small oscillations, that are very likely caused by the cell size of *3DGeoInt*.



(a) Bounce time τ_b



(b) Bounce frequency ω_b

Figure 6.9: Bounce time (a) and bounce frequency (b) of trapped and passing particles: Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
In figure (a), τ_b is the bounce time for every particle orbit averaged over 200 bouncing periods. At the transition from trapped particles (left side of the plot) to passing particles (right side of the plot), a singularity (particle stands still) is clearly visible. The bounce frequency in figure (b) is calculated by $\omega_b = \frac{2\pi}{\tau_b}$.

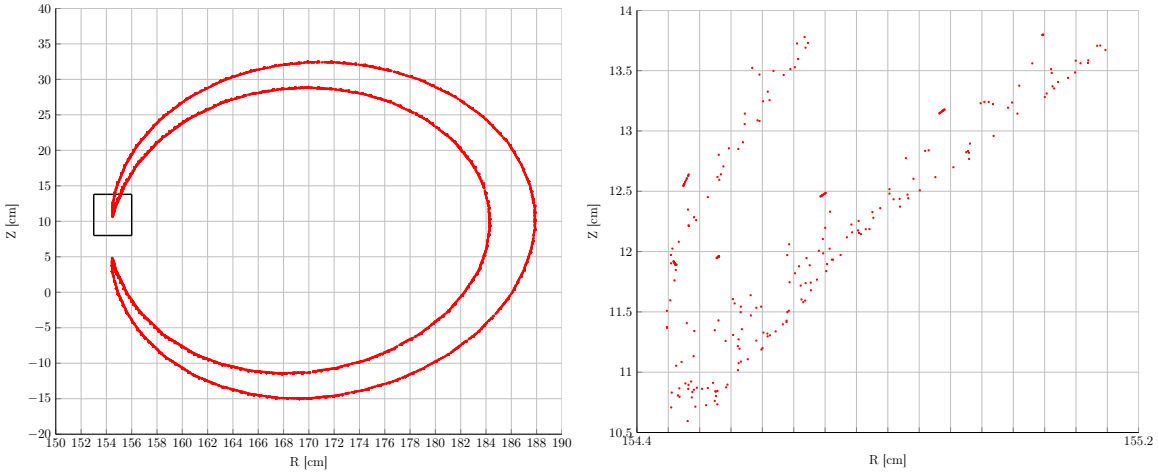
6.4 Evaluation of orbits under non-axisymmetric perturbation

For the demonstration of the working functionality of *3DGeoInt* only axisymmetric systems were examined in the previous sections. One of the main reasons to develop *3DGeoInt* is to calculate particle orbits for axisymmetric devices with weak non-axisymmetric perturbations. Such three-dimensional perturbations are for instance created by coils, that are used for the mitigation of edge localized modes (ELMs) [20]. The field quantities with a weak non-axisymmetric perturbation, that are used for the orbit integration with *3DGeoInt*, are obtained from [17].

6.4.1 Single trapped particle

Figure 6.10a depicts the Poincaré cut of a trapped particle under a non-axisymmetric perturbation. For non-axisymmetric systems the orbit is not necessarily confined anymore in the poloidal plane. Already a first glimpse at the plot of 6.10a shows, that under the given physical condition the orbit is not exactly closed. The enlargement of

the banana tip in 6.10b, clearly shows the non-confinement in the poloidal plane.



(a) Overview with enlargement indication

(b) Enlargement of banana tip

Figure 6.10: Poincaré cut of a trapped particle orbit under non-axisymmetric perturbation:
Starting position: $x^i = (R = 186 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$, pitch angle: $\chi_0 = 2.0$, electrostatic potential: $\Phi = 0$
The enlargement of the banana tip in figure (b) depicts, that the particle orbit is not exactly closed anymore under a non-axisymmetric perturbation.

6.4.2 Ensemble of non-interacting passing particles

As contrasted with section 6.4.1, where only a single trapped particle was examined, an ensemble of non-interacting passing particles with various starting positions in the fusion device is analysed. In particular, we are interested, how a weak non-axisymmetric perturbation is influencing the particle orbits.
Figure 6.11 depicts the Poincaré cut of many passing particle orbits, that are started from different positions. As one can clearly see, the particle orbits are definitely affected by the non-axisymmetric perturbation, especially towards the wall of the fusion device. The particle orbits are not necessarily closed anymore in the poloidal plane and behave chaotically. Furthermore, magnetic islands appear plainly visible. In order to examine the behaviour in vicinity of the walls better, the density of orbits was increased.

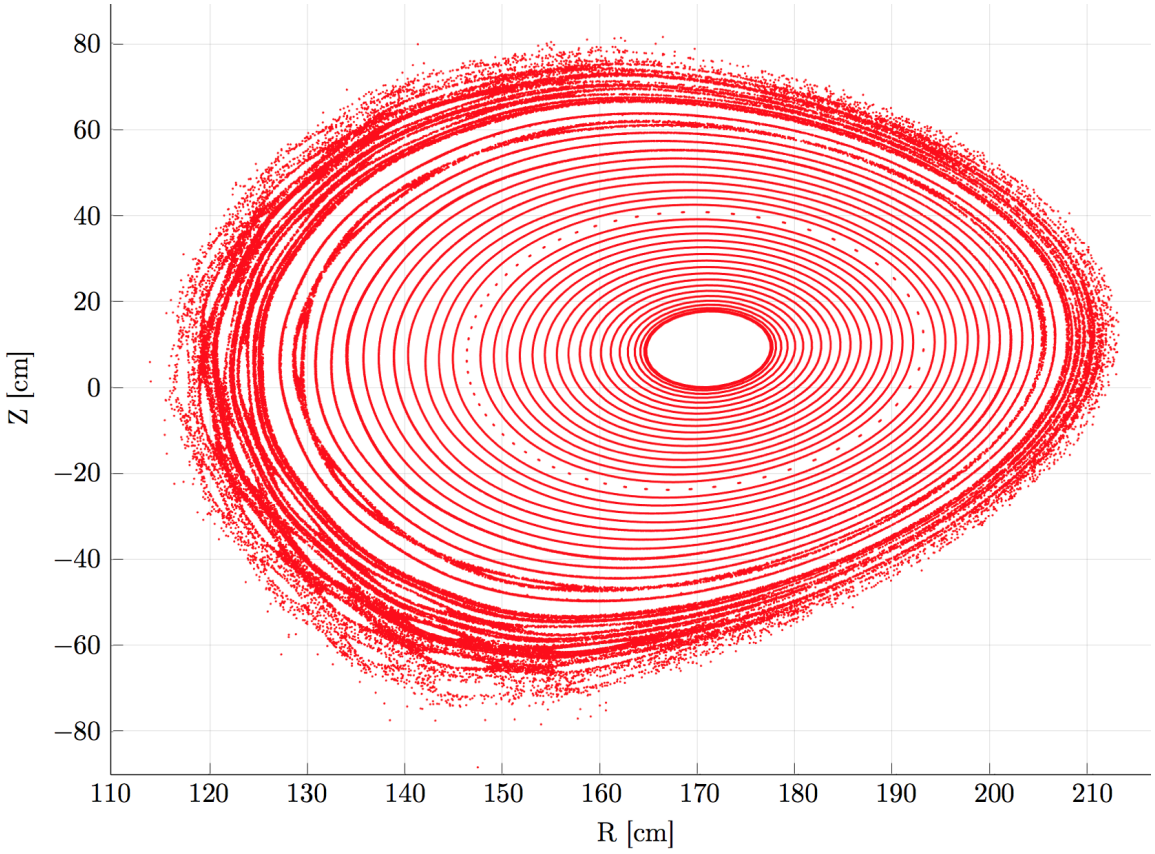


Figure 6.11: Ensemble of passing particles under non-axisymmetric perturbation:
Pitch parameter: $\lambda_0 = 0.9$
Towards the wall of the fusion device, chaotic behaviour of particle orbits and magnetic islands can bee seen.

Chapter 7

Benchmarking

7.1 Accuracy of *3DGeoInt*

The following section evaluates the accuracy of physical quantities which were calculated by *3DGeoInt*. For that purpose, the conservation of the toroidal angular momentum, which is an invariant of motion in axisymmetric systems, is examined. Furthermore the electrical component of the toroidal precession frequency is compared to an analytical expression. Finally, the influence of the mesh-size on these quantities is analysed.

7.1.1 Conservation of canonical toroidal angular momentum

Following [2], in case of an axisymmetric system, the guiding centre motion is fully integrable, because there exist three integrals of motion, which determine each orbit in the 5D phase space: the total energy w, the magnetic moment μ and the canonical toroidal angular momentum p_φ. They are respectively given by

$$\begin{aligned} w &= \frac{m\left(v_\perp^2 + v_\parallel^2\right)}{2} + e\Phi, \\ \mu &= \frac{mv_\perp^2}{2B} \quad \text{and} \\ p_\varphi &= mv_\parallel \frac{B_\varphi}{B} + \frac{e}{c} A_\varphi, \end{aligned} \tag{7.1}$$

where B_φ is the co-variant toroidal component of the magnetic field and A_φ is the co-variant component of the vector potential. For transport modelling it is of primary importance, that the invariants (7.1) are accurately conserved.
As it is described in section 2, the conservation of the total energy w and the magnetic moment μ are guaranteed up to the computer accuracy by definition of the geometric integrator. The conservation of the canonical toroidal angular momentum p_φ has not been used for the algorithm of the geometric integrator, thus one has to assess, if p_φ remains a constant of motion.

Figure 7.1 depicts the canonical toroidal angular momentum p_φ versus the number of passes through a cell for 6 different starting positions. As one can easily see in the plot, p_φ seems to remain constant. The mesh (see chapter 3), that was used for the orbit calculation, has the following number of hexahedrons: $n_R \times n_Z \times n_\varphi = 85 \times 100 \times 20$. For 10^7 passes through tetrahedrons (6 per 1 hexahedron), the relative change of p_φ is in the order of 10^{-7}. In average the relative change of p_φ in a cell is in the order of 10^{-14}. Thus, in conclusion the *3DGeoInt* conserves the invariants of motion up to a sufficient accuracy.

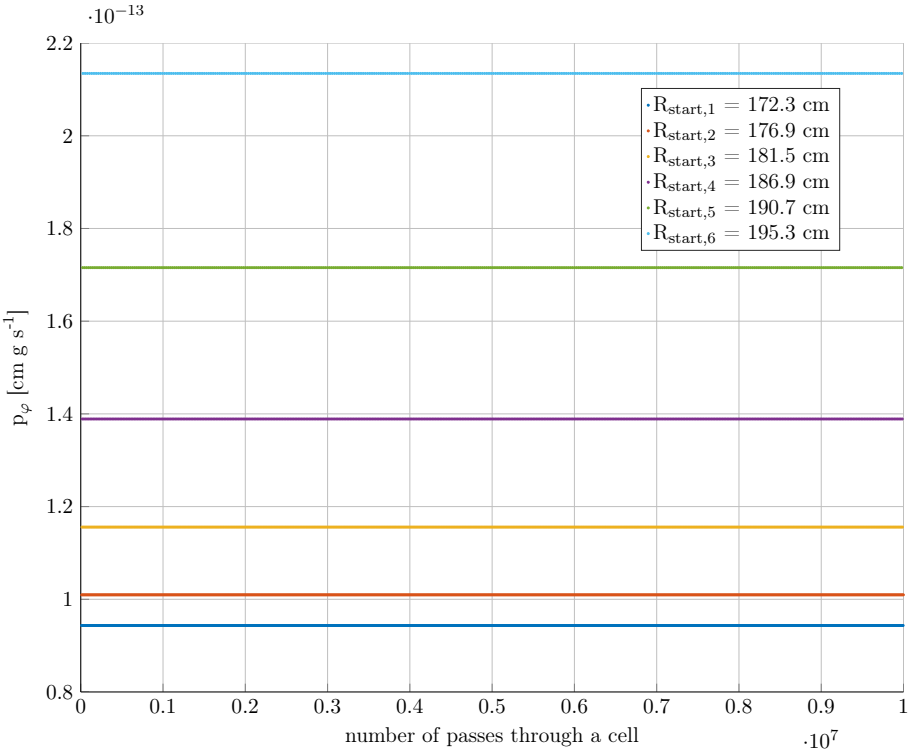


Figure 7.1: Canonical toroidal angular momentum p_φ vs. the number of passes through a cell:
Starting positions for all 6 particles: $x^i = (\varphi = 0,\ Z = 0\,\text{cm})$. R_{start} can be taken from the plot's legend.
Total number of hexahedrons in the mesh: $n_R \times n_Z \times n_\varphi = 85 \times 100 \times 20$.
The canonical toroidal angular momentum p_φ is in average conserved up to the relative change in the order of 10^{-14} for a pass through a cell.

7.1.2 Toroidal precession frequency Ω_{tor}

As mentioned in section 6.3, after the particle has moved through one bounce period, meaning that the same (R,Z) coordinates are reached again, the particle is toroidally displaced by the angle $\Delta\varphi$. The toroidal displacement per bounce time is called the

toroidal precession frequency

$$\Omega_{tor} = \frac{\Delta\varphi}{\tau_b}. \tag{7.2}$$

According to [19] the toroidal precession frequency for trapped particles can be decomposed into a magnetic and electric term

$$\Omega_{tor} = \Omega_{tE} + \Omega_{tB}, \tag{7.3}$$

where Ω_{tE} is the toroidal precession frequency caused by electric guiding center drifts and Ω_{tB} the respective drift frequency caused by magnetic guiding center drifts. Furthermore, for the electric toroidal precession frequency a very simple analytical formula is given by

$$\Omega_{tE} = c\frac{\partial\Phi}{\partial\Psi}, \tag{7.4}$$

where c is the speed of light, Φ the electrostatic potential and Ψ the poloidal magnetic flux.

In Figure 7.2 the toroidal precession frequency Ω_{tor} versus the pitch angle at the starting position λ_0 for various electrostatic potentials Φ is depicted. The missing data in the center of the plot is caused by the singularity of the bounce time at the transition from trapped particles to passing particles. Resulting from equation (7.4) and as one can assume from the plot, the toroidal precession frequencies of trapped particles (left side of the singularity) for various electrostatic potentials are equidistantly spaced.

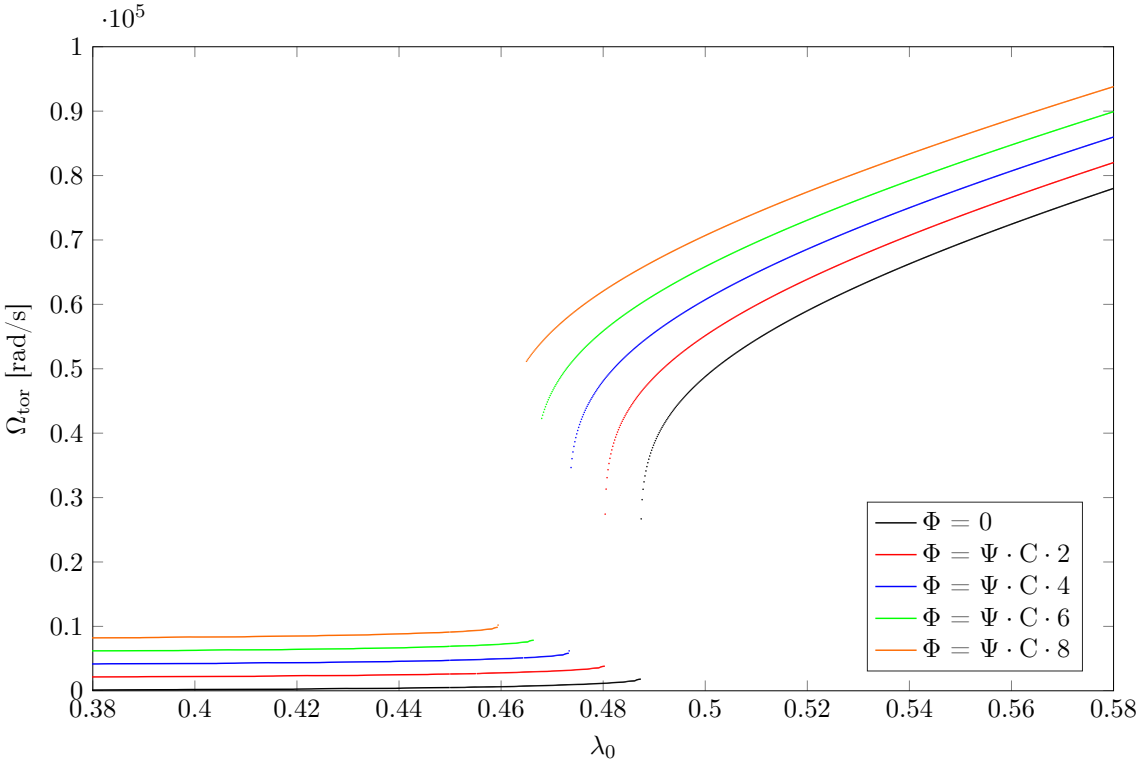


Figure 7.2: Toroidal precession frequency versus pitch angle for various electrostatic potentials:
Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
The electrostatic potential is linearly increased for equally sized steps. The toroidal precession frequency is depicted under the variation of the pitch angle at the starting position for the electrostatic potentials.

In order to examine if the calculation of the toroidal precession frequency by *3DGeoInt* is accurate, one tries to approve equation (7.4). By subtracting the toroidal precession frequencies of two different electrostatic potentials ($\Phi_2 = \Psi \cdot C \cdot 2$, $\Phi_4 = \Psi \cdot C \cdot 4$)

$$\begin{aligned} \Omega_{tor,4} - \Omega_{tor,2} &= \Omega_{tE,4} - \Omega_{tE,2} \\ &= c \cdot \left(\frac{\partial \Phi_4}{\partial \Psi} - \frac{\partial \Phi_2}{\partial \Psi} \right) \\ &= c \cdot \left(\frac{\partial}{\partial \Psi} \Psi \cdot C \cdot 4 - \frac{\partial}{\partial \Psi} \Psi \cdot C \cdot 2 \right) \\ &= \underbrace{c \cdot C}_{C'} \cdot 2 \end{aligned} \tag{7.5}$$

the magnetic toroidal precession frequency Ω_{tB} vanishes. One can normalize the difference of the electric toroidal precession frequencies by C', which is essentially the electric toroidal precession frequency for the electrostatic potential $\Phi = C \cdot \Psi$, and

obtains

$$\frac{\Omega_{tE,4} - \Omega_{tE,2}}{C'} = 2. \tag{7.6}$$

Figure 7.3 shows the normalized difference of two toroidal precession frequencies for the electrostatic potentials $\Phi_2 = \Psi \cdot C \cdot 2$, and $\Phi_4 = \Psi \cdot C \cdot 4$ under the variation of the pitch parameter. Once again, the mesh has the following number of hexahedrons: $n_R \times n_Z \times n_\varphi = 85 \times 100 \times 20$.
According to equation (7.6) the theoretical result is the constant 2. The plot clearly depicts relative deviations of up to 5 %, which is very likely caused by the choice of the number of cells of *3DGeoInt*.

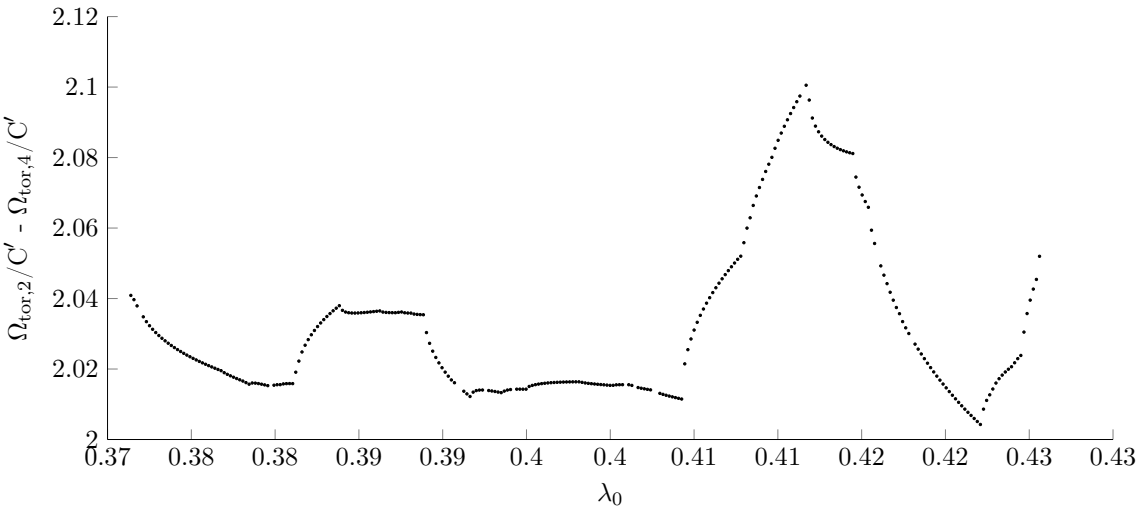


Figure 7.3: Normalized difference of two toroidal precession frequencies:
Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Total number of hexahedrons in the mesh: $n_R \times n_Z \times n_\varphi = 85 \times 100 \times 20$.
Toroidal precession frequencies for different electrostatic potentials:
$\Omega_{tor,2} = \Omega_{tor}(\Phi_2 = \Psi \cdot C \cdot 2)$, $\Omega_{tor,4} = \Omega_{tor}(\Phi_4 = \Psi \cdot C \cdot 4)$
The normalized difference of two toroidal precession frequencies is depicted under the variation of the pitch parameter at the starting position λ_0.

7.1.3 Variation of the mesh-size

In the previous sections the conservation of the canonical toroidal angular momentum and the electrical component of the toroidal precession frequency were examined in order to understand the accuracy of *3DGeoInt*. An important influence had been neglected so far: the mesh-size. Unlike high order adaptive ODE integrators, the algorithm of *3DGeoInt* does not check how accurate single Runge-Kutta steps are. That's why the size of a Runge-Kutta step (roughly equivalent to the size of the mesh) in *3DGeoInt* has an impact on its accuracy. Note, that the field quantities are approximated by linear functions in a cell. Thus, the piecewise linear representation of the

field quantities is dependent on the mesh-size. For further examination, the accuracy tests from the sections 7.1.1 and 7.1.2 are repeated under the variation of the mesh-size.

Canonical toroidal angular momentum:
First of all, one can see in figure 7.4 the canonical toroidal angular momentum $\langle p_\varphi \rangle$, averaged over 10^5 orbit mappings (orbit-crossing of poloidal plane at $\varphi = 0$), for several different mesh-sizes. Additionally, for each treated mesh-size the relative change of p_φ per tetrahedron-cell is given. It is clearly visible, that orbit-calculations with too few grid-points (①, ② and ③) yield a canonical toroidal angular momentum, that differs from a realistic one. Whereas, the canonical toroidal angular momentum of the calculations from ④ on seems to converge towards a realistic value.
For the calculations ④, ⑤ and ⑥ the number of mesh-points n_R and n_Z was held constant, while the number of points in toroidal φ-direction was increased. Since we deal with an axisymmetric system, the mesh-size in toroidal direction has almost no influence, as one can see in figure 7.4.
The relative change of p_φ per tetrahedron-cell decreases with a finer mesh. Doubling the number of mesh-points in each three directions (⑥, ⑦ and ⑧), results in a decrease of the respective relative change of almost one order of magnitude per mesh-point doubling.

Electrical component of the toroidal precession frequency:
Second, also the electrical component of the toroidal precession frequency is examined under the variation of the mesh-size. Figure 7.3 in section 7.1.2 shows the normalized difference of two toroidal precession frequencies for two electrostatic potentials under the variation of the pitch angle at the starting position. One can see a deviation from the expected value. The maximum relative deviation from this expected value of the difference of normalized electrical components under the variation of the mesh-size is shown in figure 7.5. Intuitively, one expects a decreasing relative deviation with the increase of mesh-points. Nevertheless, figure 7.5 clearly depicts, that one does not obtain such an expected result.
The reason might be the complicated physical behaviour of Ω_{tB} (see eq. 7.3). In order to examine the electrical component of the toroidal precession frequency, the difference of two toroidal precession frequencies with different electrostatic potentials is made. The underlying assumption is, that the magnetic component of the toroidal precession frequency does not change with a variation in the electrostatic potential. Due to the complicated physical behaviour of Ω_{tB}, this assumption might be wrong for $3DGeoInt$, which results in the deviation that can be seen in figure 7.5.

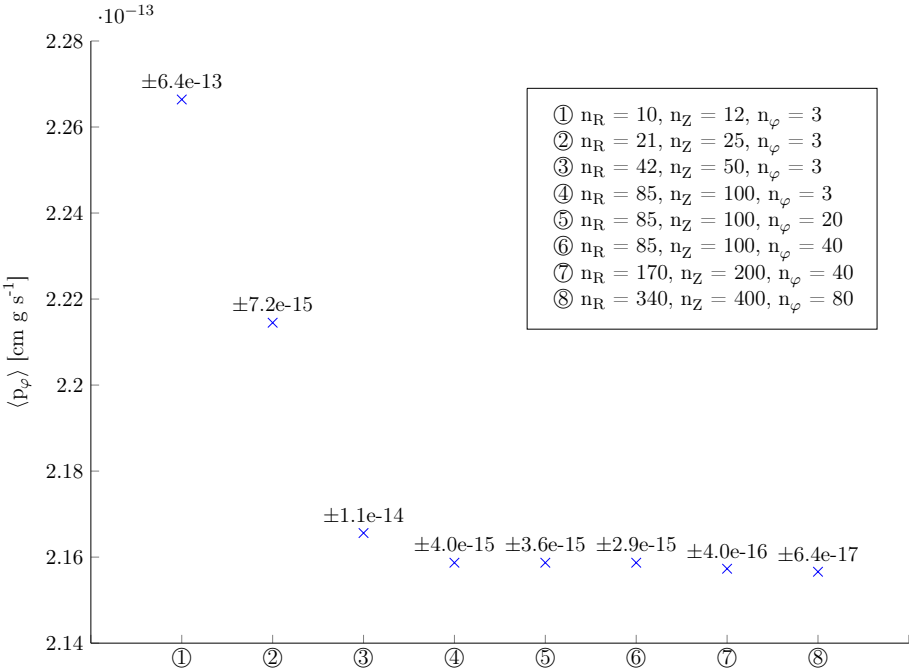


Figure 7.4: Accuracy of the averaged canonical toroidal angular momentum under the variation of the mesh-size:
Starting position: $x^i = (R = 195 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch parameter: $\lambda_0 = 0.56$
The canonical toroidal angular momentum was averaged over 10^5 orbit mappings (orbit-crossing of poloidal plane at $\varphi = 0$). Additionally, the relative change of $\langle p_\varphi \rangle$ per tetrahedron-cell is given.

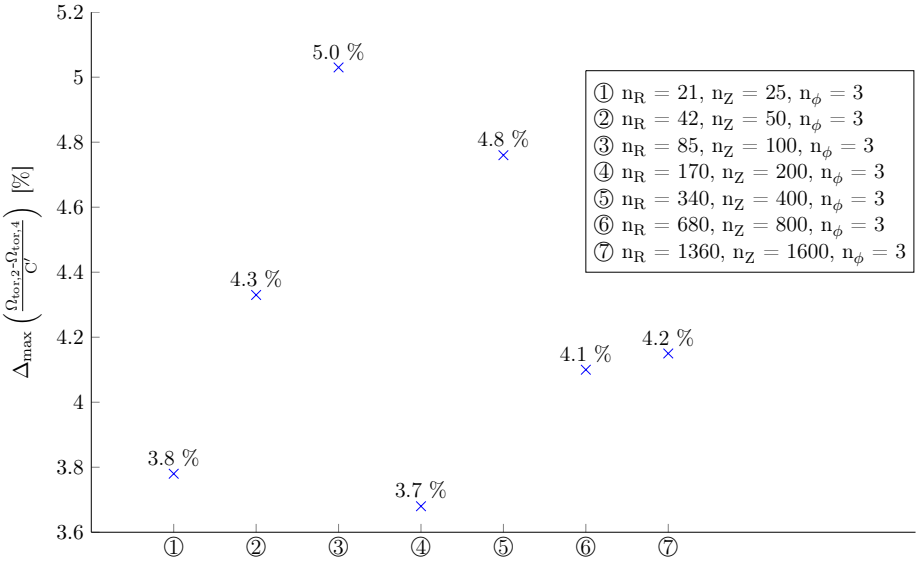


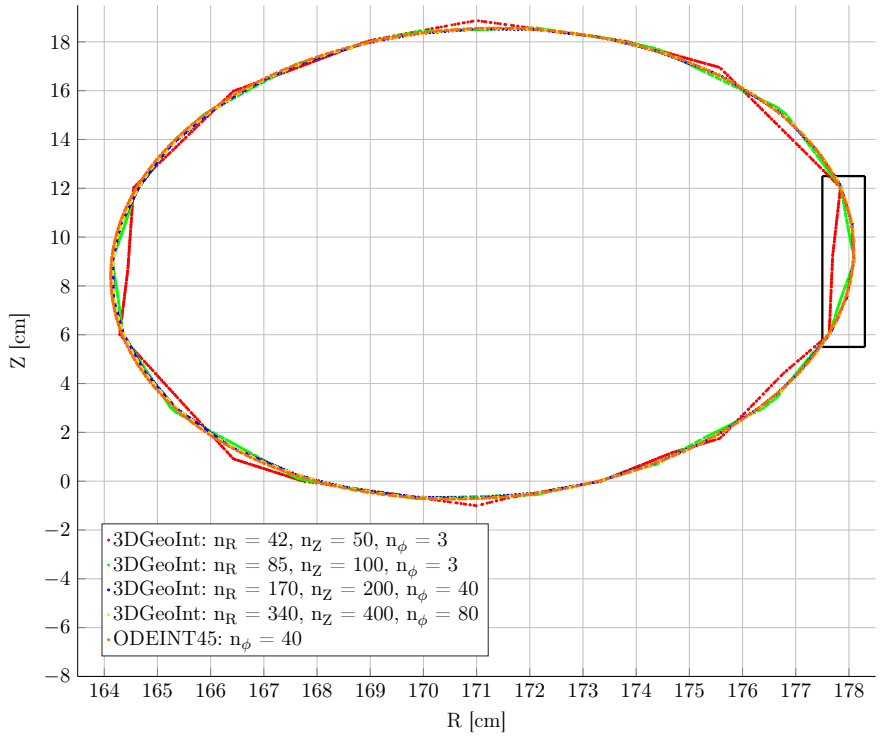
Figure 7.5: Maximum relative deviation of the difference of normalized electrical components of two toroidal precession frequencies with different electrostatic potentials:
Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch parameter for every mesh-size varied: $\lambda_0 = 0.37 \ldots 0.45$

7.2 Comparison of *3DGeoInt* with a high order adaptive ODE integrator *ODEINT45*

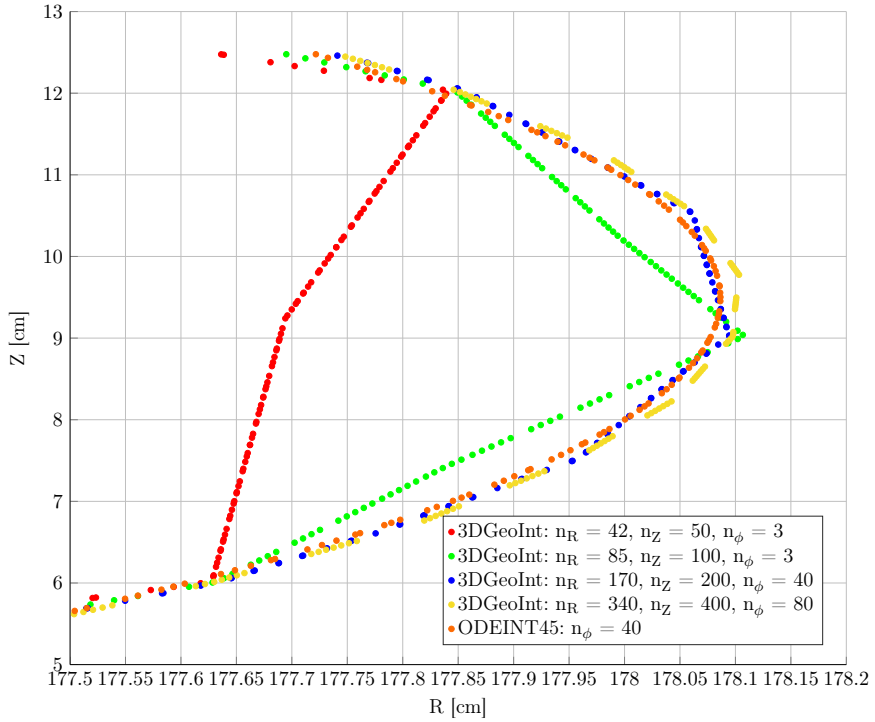
In the following sections, particle orbits that were calculated by *3DGeoInt* are compared to particle orbits, originating from a high order adaptive ODE integrator [25], called *ODEINT45*. The major difference in between these two integrators is, that in the case of *3DGeoInt* the field quantities are approximated by linear functions in cells, which results in a piecewise linear representation of the field quantities with the benefit of less sensitivity to noise in the data. Whereas, the field quantities in the case of *ODEINT45* are interpolated by high order splines. For the sake of convenience, particle orbits are compared for axisymmetric systems.

7.2.1 Orbits

First of all, we want to analyse qualitatively how the particle orbits of the two integrators differ. For that purpose several particle orbits are calculated by *3DGeoInt* with different mesh-sizes, meaning a variation of the cell-size in which the field quantities are approximated by linear functions. Figure 7.6 depicts the Poincaré cut of these *3DGeoInt*-particle orbits, and additionally also one orbit calculated by *ODEINT45*. An overview of the orbits of passing particles is given in figure 7.6a with an indication of enlargement, that is depicted in figure 7.6b.
As one can see very clearly, the size of the mesh has a strong influence on the particle's orbit. The orbit accuracy is dramatically reduced, if there are too few cell-elements. The finer the mesh-grid is, the more accurate are the particle orbits that are calculated by *3DGeoInt*. As one can read in section 7.2.2, the computational effort increases with the number of cells. Thus, one has to make a trade-off in between orbit-accuracy and computation speed.
The particle orbit that is calculated by *ODEINT45* can be treated as the "real" reference-orbit.



(a) Overview with enlargement indication



(b) Enlargement

Figure 7.6: Poincaré cut of particle orbits, differing in the integration method and mesh size:
Starting position: $x^i = (R = 173 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch angle: $\lambda_0 = 0.9$
The orbit accuracy is dramatically influenced by the mesh-size.

7.2.2 Computation speed

One of the requirements for *3DGeoInt* is computational efficiency. To examine the CPU-speed of *3DGeoInt*, particle orbits with same starting conditions but different computation (orbit-integration) times for various mesh-sizes were calculated. Before the actual orbit-integration can take place, the field quantities at the mesh-points must be provided and the tetrahedron-specific algorithm-constants of the linearized set of equations of motion have to calculated. That is why, there is the same offset for a certain mesh-size in each and every orbit calculation for different integration times. In order to obtain the computation speed, a linear fit was made for the tracked samples of CPU-time versus orbit integration time.

As one can see in figure 7.7, the computation speed was tracked for both *3DGeoInt* and *ODEINT45* in order to compare the geometric integrator with a conventional high order adaptive ODE integrator. The maximum step size of the adaptive *ODEINT45* can be manually stated: n_φ gives the minimum number of steps per toroidal orbit circulation (orbit integration in toroidal φ-direction) that are imposed to *ODEINT45* for the orbit integration.
Since we are interested in systems with weak non-axisymmetric perturbations, a not too coarse mesh in toroidal direction is a requirement. Thus, one is in particular focused on the computation speed for higher numbers of n_φ for the orbit integration. Figure 7.7 clearly depicts, that the computation speed of *3DGeoInt* and *ODEINT45* are about the same for $n_\varphi = 40$. Furthermore, one has to mention, that *3DGeoInt* already provides the particle's coordinates and velocities at the boundaries of spacial cells, what is needed for the evaluation of the distribution function. Thus, it is more efficient than a direct solution of the equations of motion with *ODEINT45*, where these quantities at the cell-boundaries have to be computed additionally.

Obviously, the computation effort decreases, with decreasing mesh-size, whereby *ODEINT45* slightly beats *3DGeoInt* in speed, but note that we are mainly not interested in that regime.

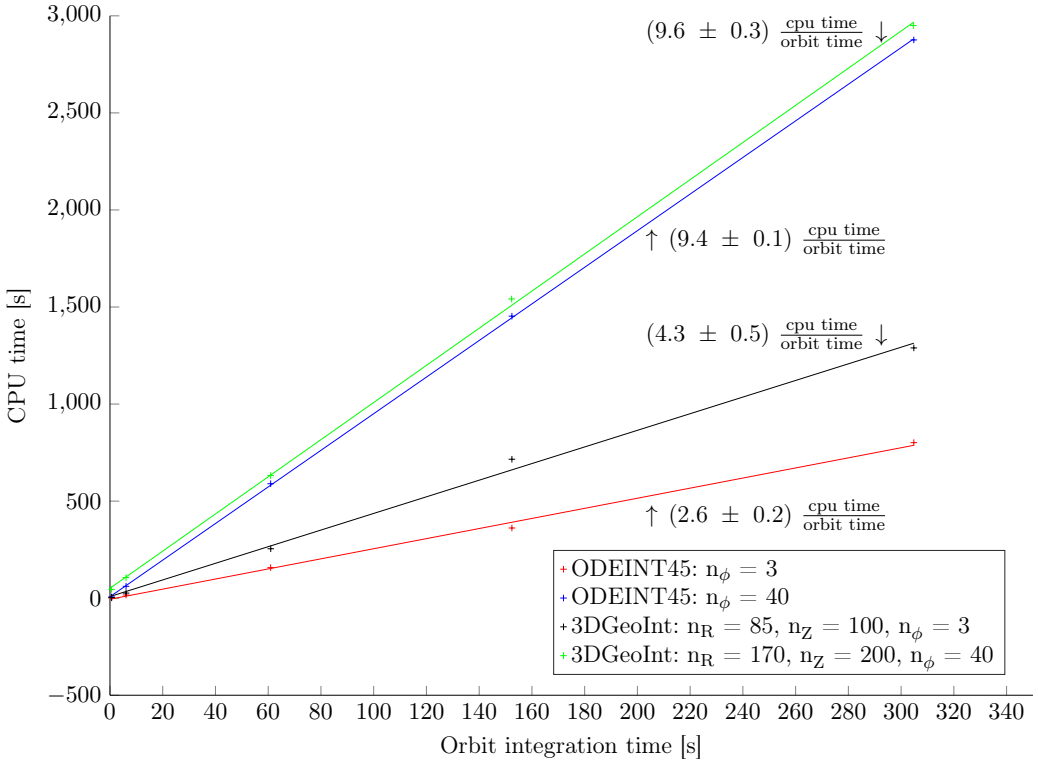


Figure 7.7: Computation speed: comparison of *3DGeoInt* with *ODEINT45*:
Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch parameter: $\lambda_0 = 0.56$
The CPU-time versus the orbit time is tracked for various mesh-sizes and linearly fitted.

7.3 Effect of noise in the field quantities

One of the big disadvantages of high order adaptive ODE integrators are unwanted oscillations of the field quantities. These are caused by a high order spline-interpolation of an inaccurately represented electromagnetic field (noise in the data). Such oscillations in the field quantities can destroy the physical properties of the particle orbit. Typically, a high order adaptive ODE integrator is unable to calculate proper particle trajectories, if there is noise in the underlying data of the electromagnetic field.
In the following section, it is qualitatively demonstrated that *3DGeoInt* is able to compute particle orbits even for inaccurately represented electromagnetic fields, due to the approximation of field quantities with linear functions.
For all calculations a mesh size of $n_R \times n_Z \times n_\varphi = 85 \times 100 \times 3$ was chosen.

7.3.1 Axisymmetric noise in the vector potential

First of all, the vector potential is perturbed by an axisymmetric noise. For that purpose a random number generator ($\xi = 0 \ldots 1$) is used to create noise, which was subsequently added in a relative magnitude to the vector potential, such that

$$A_k^{\text{noise}} = A_k \cdot (1 + \epsilon \cdot \xi), \tag{7.7}$$

where A_k^{noise} and A_k are the perturbed and respective unperturbed vector potential and ϵ gives the relative magnitude of noise.

Figure 7.8 and 7.9 show the Poincaré cut of trapped and passing particles for $\epsilon = 1$ %, 5 %, 10 % and 20 %.
As one can easily see, the particle orbits remain closed in the poloidal plane, due to the fact that the perturbation is axisymmetric, meaning that the canonical toroidal angular momentum is conserved. Even though the magnitude of the added noise is quite high, the particle orbits keep a similar shape in comparison with the unperturbed orbit. Thus, the physical behaviour of particle trajectories computed by *3DGeoInt* is not completely destroyed (chaos), if the electromagnetic field is inaccurately but axisymmetric represented.

7.3.2 Non-axisymmetric noise in the vector potential

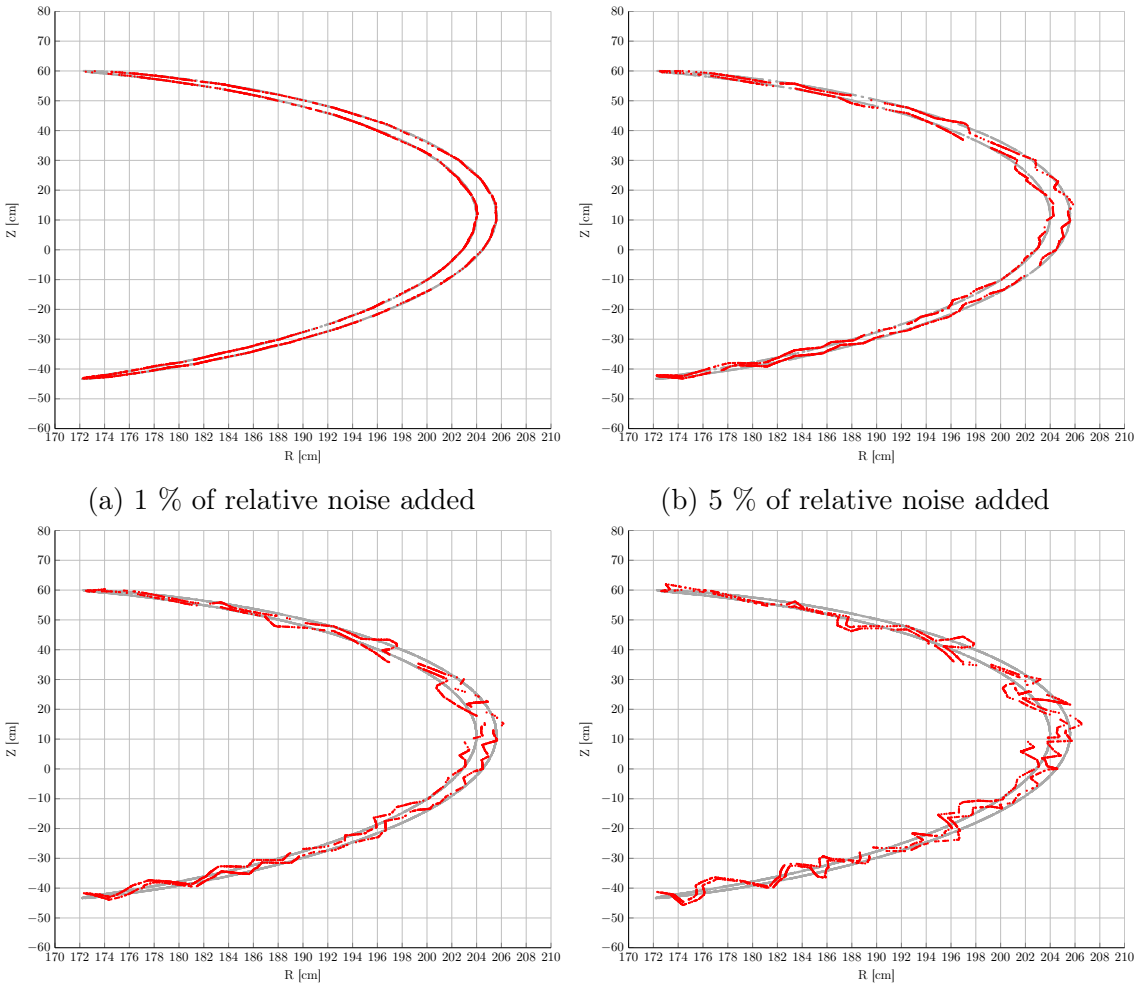
Second, the vector potential is perturbed by a non-axisymmetric periodic noise, such that

$$A_k^{\text{noise}} = A_k \cdot (1 + \epsilon \cdot \xi \cdot \cos\varphi), \tag{7.8}$$

where φ is the toroidal angle coordinate.

Figure 7.10.a shows the Poincaré cut of a passing particle orbit, whereby the vector potential was perturbed by a non-axisymmetric periodic noise with a relative magnitude of $\epsilon = 0.1$ %. It is clearly visible, that the orbit does not remain closed in the poloidal plane. Even more, dynamical chaotic behaviour becomes apparent.
Furthermore, one is interested in time-evolution of the canonical toroidal angular momentum. Therefore, in figure 7.10.b p_φ is depicted for roughly $2 \cdot 10^8$ tetrahedron passes which corresponds to 10^6 toroidal mappings. Qualitatively spoken, the variance of p_φ doesn't broaden extremely within the "high" number of mappings, that's why one can assume diffusive behaviour. In order to calculate the particle distribution function within Monte-Carlo simulations, the mean free path length of a particle is

much smaller than 10^6 toroidal mappings, thus the relative change of p_φ seems to be within sufficient accuracy limits. Obviously, a detailed quantitative examination of the time evolution of p_φ is needed in order to draw a conclusion regarding diffusive behaviour.



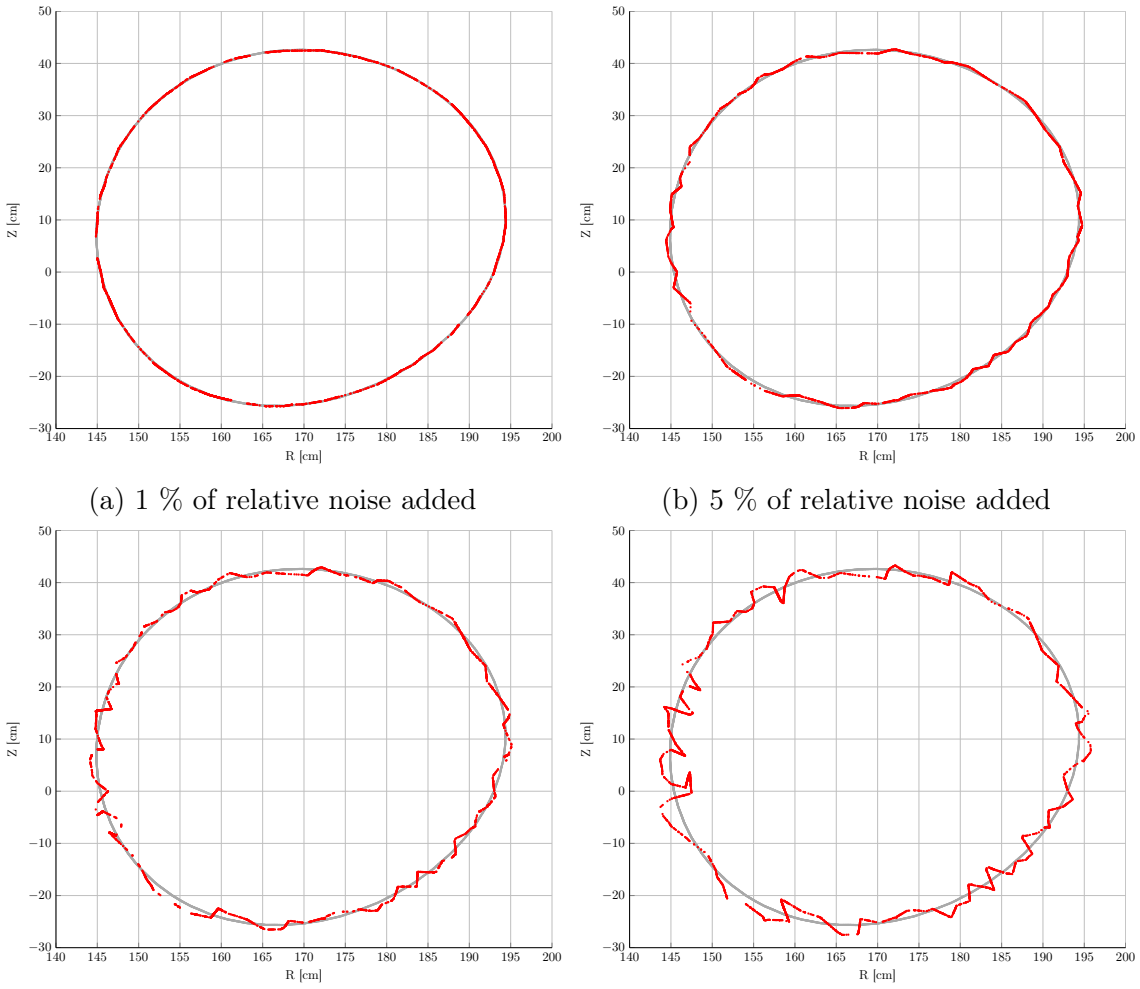
(a) 1 % of relative noise added

(b) 5 % of relative noise added

(c) 10 % of relative noise added

(d) 20 % of relative noise added

Figure 7.8: Poincaré cut of trapped particle orbits, differing in magnitude of relative axisymmetric noise added to the vector potential:
Starting position: $x^i = (R = 204 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch angle: $\lambda_0 = 0.37$
The noise was created with a random number generator $(0 \ldots 1)$ and added to the vector potential, whereby the magnitude of the relative noise was varied. Additionally to the particle orbits with noisy vector potential (red), also the unperturbed orbits (grey) are depicted.



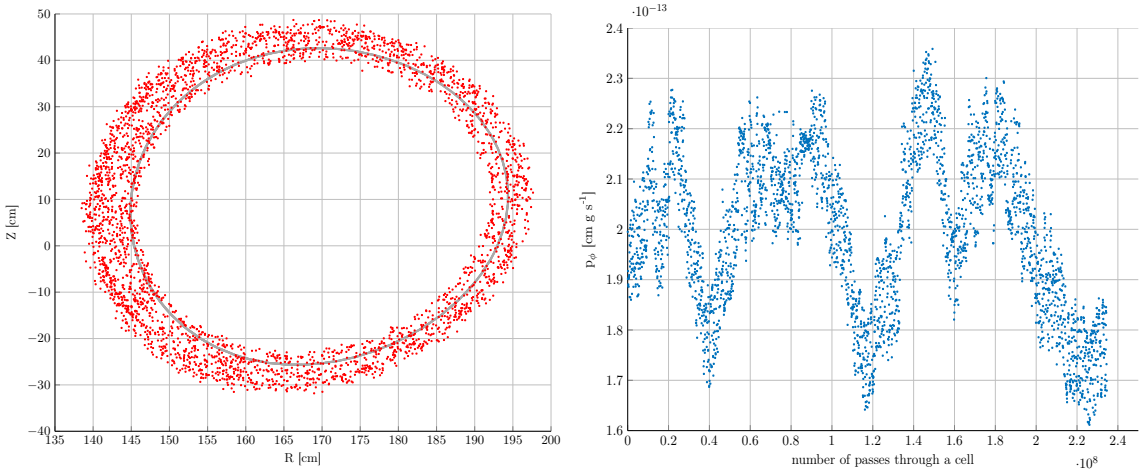
(a) 1 % of relative noise added

(b) 5 % of relative noise added

(c) 10 % of relative noise added

(d) 20 % of relative noise added

Figure 7.9: Poincaré cut of passing particle orbits, differing in magnitude of relative axisymmetric noise added to the vector potential:
Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch angle: $\lambda_0 = 0.90$
The noise was created with a random number generator $(0 \ldots 1)$ and added periodically in φ-direction to the vector potential, whereby the magnitude of the relative noise was set to 0.1 %. In figure (a), additionally to the particle orbit with noisy vector potential (red), also the unperturbed orbit (grey) is depicted. Figure (b) shows the respective canonical toroidal angular momentum of the passing particle orbit.



(a) 0.1 % of relative noise added

(b) Canonical toroidal angular momentum

Figure 7.10: Poincaré cut of passing particle orbit and canonical toroidal angular momentum under non-axisymmetric noise in the vector potential:
Starting position: $x^i = (R = 193 \text{ cm}, \varphi = 0, Z = 0 \text{ cm})$
Pitch angle: $\lambda_0 = 0.90$
Number of toroidal orbit-turns (mappings): 10^6.
The noise was created with a random number generator $(0 \ldots 1)$ and added to the vector potential, whereby the magnitude of the relative noise was varied. Additionally to the particle orbits with noisy vector potential (red), also the unperturbed orbits (grey) are depicted.

7.4 Statistical evaluation of the iteration procedure

In section 5.5 the equations of motion are solved by using the Runge-Kutta method, applied in an iteration procedure. This procedure uses Newton's method, a quadratic approach and further analytical approximations, in order to meet the convergence criterion to find an appropriate exit point. One might be interested, how these different methods are used quantitatively for random orbits. For that purpose several orbits under random starting positions and random pitch parameters were computed. In total particles had to pass more than $5 \cdot 10^8$ tetrahedron-cells.
In table 7.1 the different methods, that are used in the iteration procedure, are listed with their relative contribution among the used methods.

Table 7.1: Relative contribution of methods used in the iteration procedure until the orbit is converged at the exit point:

Total number of hexahedrons in the mesh: $n_R \times n_Z \times n_\varphi = 85 \times 100 \times 20$

Methods used in the iteration procedure	**relative contribution / %**
0 Newton iterations (1 analytical approximation)	0.0514
1 Newton iteration	64.3124
2 Newton iterations	35.2788
3 Newton iterations	0.2941
4 Newton iterations	0.0055
Quadratic approach + Newton iterations	0.0003
New analytical approximation + Newton iterations	0.0204
Troubleshooting: Wrong convergence	0.2941
Troubleshooting failed: Bisection method	0.0055

The iteration procedure in a cell starts with an analytical approximation for the time to pass through the cell. If the convergence criterion is not met, the methods in table 7.1 are applied to obtain additional integration time steps.
As one can easily see in table 7.1, the particle orbit converged at the appropriate exit point in about 64 % of the cases after only one Newton iteration. Further 35 % are covered with a second Newton iteration. Troubleshooting for convergence points, that are not the appropriate exit point, has to be applied in 0.3 % of the cases and the computationally expensive bisection method is only used in 0.006 %.
These figures indicate, that the analytical approximation that is made in the beginning of the iteration procedure, matches the actual time of a particle to pass the tetrahedron-cell quite good. Thus, the reason for the computational efficiency of *3DGeoInt* can be found in the order of accuracy of the analytical approximation. A further analysis of the theoretically possible extrema of the initial integration time step τ^α can be found in appendix A.

Chapter 8

Conclusion and outlook

In this thesis a three-dimensional integrator for guiding center orbits of charged particles in toroidal fusion devices was described.
Apart from its theoretical derivation and numerical implementation, a qualitative evaluation of particle orbits was performed. It was shown that for axisymmetric devices the physical behaviour of particle orbits calculated by *3DGeoInt* coincides with the description in the literature. In particular, it was demonstrated that the orbits in axisymmetric devices remain closed in the poloidal projection plane. Furthermore, two kind of particles appeared, depending on the pitch parameter at the starting position or the electrostatic potential: trapped and passing ones.
With the introduction of a weak non-axisymmetric perturbation, dynamical chaotic behaviour of the particle trajectories became apparent.

In addition to the qualitative demonstration of the physical properties of particle orbits, the accuracy of physical quantities was also studied. For that purpose the conservation of the canonical toroidal angular momentum, which is an invariant of motion in axisymmetric systems, was examined. It could be shown that for a fine mesh-size the average relative change of the canonical toroidal angular momentum in a cell is below the order of 10^{-16}, which is a sufficient accuracy for the purpose of evaluating the particle distribution function where the respective conservation is important. Furthermore, it could clearly be demonstrated that the accuracy of the canonical toroidal angular momentum is a function of the mesh-size.
Additionally, the electrical component of the toroidal precession frequency was compared to an analytical expression. By varying the mesh-size, the accuracy of the electrical component of the toroidal precession frequency could not be enhanced. Thus, further examination is needed, in particular of the complicated physical behaviour of the magnetic component of the toroidal precession frequency, in order to understand how the accuracy of the electrical component is related to the mesh-size.

One of the requirements for *3DGeoInt* was computational efficiency. Therefore,

the computation speed of *3DGeoInt* was compared to a high order adaptive ODE integrator, which resulted in roughly the same value. However, it must me mentioned that *3DGeoInt* already provides the particle's coordinates and velocities at the boundaries of spacial cells, which is needed for the evaluation of the distribution function. Thus, it is more efficient than a direct solution of the equations of motion with a high order adaptive ODE integrator, where these quantities at the cell-boundaries have to be computed additionally.
For the purpose of understanding the underlying reason for the computational efficiency of *3DGeoInt*, a quantitative evaluation of the numerical iteration procedure was accomplished. This examination exposed that the particle orbit converged at the appropriate exit point in about 64 % of all cases after only one Newton iteration. Further 35 % were covered with a second Newton iteration. Thus, the reason for the computational efficiency of *3DGeoInt* is the accuracy of the initial analytical approximation.

Finally, it was qualitatively demonstrated that unlike high order adaptive ODE Integrators, *3DGeoInt* is able to compute particle orbits even for inaccurately represented electromagnetic fields due to the approximation of field quantities with linear functions. An axisymmetric noise term in the relative magnitude of up to 20 % was added to the vector potential. The particle orbits remained closed in the poloidal projection plane and kept a similar shape in comparison with the unperturbed orbit. With the introduction of periodic non-axisymmetric noise, the particle orbits behaved chaotically. In order to understand the time evolution of the canonical toroidal angular momentum, a detailed quantitative examination is needed to draw a conclusion regarding diffusive behaviour.

Before *3DGeoInt* can be inserted in the Monte-Carlo simulation, which solves the overall problem of calculating the particle distribution function, further studies and improvements have to be conducted. First of all, a more sophisticated field-aligned mesh must be applied to *3DGeoInt*, because that is needed in the overall Monte-Carlo simulation.
Second, the chaotic behaviour of particle orbits must be examined quantitatively when weak non-axisymmetric perturbations are introduced. In particular, the diffusive behaviour should be studied under the variation of the mesh-size.

Appendix A

Analysis of linear set of ODEs

A.1 Introduction

The approach that is used in section (5) to calculate the orbits in a tetrahedron as efficient as possible, is to make an analytical approximation for the integration variable τ. In order to obtain a good value for τ, the set of ODEs is simplified by setting the implicitly τ-dependent variables $x^l(\tau)$ and $v_\parallel(\tau)$ to constants of motion. For this reason the following assumptions were made:

1. The parallel velocity $v_\parallel$ doesn't change while the particle is passing the tetrahedron.

   $v_{\parallel,const.} = v_{\parallel,e.p.}$

2. The Matrix $a^i_l = 0$ for $1 \leq i, l \leq 3$. This is equivalent to the assumption of setting $x^l_{const.}$ to the coordinate origin.

   $x^l_{const.} = 0$

The set of ODEs could then be analytically solved.

It is in our interest to improve the *3D-Integrator*, thus one should analyse the quality of those assumptions and, if possible, improve the proposal for the integration variable τ.

A.2 Extrema of the integration variable τ

If one follows the path of the particle during its way through the tetrahedron, the implicitly τ-dependent variables $x^l(\tau)$ and $v_\parallel(\tau)$ will adopt different values. Our first task is to understand, how the solution for τ can vary for obtaining an intersection in

between the orbit and the tetrahedron. Therefore, we are interested in the extrema of the integration variable τ. In section (5.2.1) a quadratic equation for τ was obtained, after $x^l(\tau)$ and $v_\parallel(\tau)$ were set to constants of motion and the set of ODEs was integrated:

$$\frac{1}{2}a^\alpha(\tau^\alpha)^2 + b^\alpha\tau^\alpha + c^\alpha = 0, \tag{A.1}$$

with the constants

$$\begin{aligned} a^\alpha &= n_i^\alpha \cdot a_4^i(a_4^4 \cdot v_{\parallel,const.} + b^4) \\ b^\alpha &= n_i^\alpha \cdot (a_4^i v_{\parallel,e.p.} + b^i + a_l^i x^l_{const.}) \\ c^\alpha &= n_i^\alpha \cdot x^i_{e.p.} - d^\alpha \end{aligned} \tag{A.2}$$

To obtain the extrema of τ, the coefficients a^α, b^α and c^α must be examined, and thus the extremal values of $x^l(\tau)$ and $v_\parallel(\tau)$, that can be adopted in the tetrahedron, have to be evaluated.

A.2.1 Extrema of $v_\parallel$ and x^l

Extrema of $v_\parallel$:
First of all, one has to understand, what is the minimum and the maximum that $v_\parallel$ can adopt in the cell. By introducing a notation with the quantity U (2.6), the complications that can occur with the sign of $v_\parallel$ vanish.
Thus, we will try to find the minimum and maximum of $v_\parallel^2$ and we will later analyze how to determine the extrema of $v_\parallel$. Following the notation (2.6), $v_\parallel^2$ takes the form

$$v_\parallel^2 = 2U = \frac{2}{m}(w - J_\perp\omega_c - e_\alpha\Phi), \tag{A.3}$$

where $w = \frac{mv^2_{\text{e.p.}}}{2}$, $J_\perp = \frac{mv^2_{\perp,\text{e.p.}}}{2\omega_{c,\text{e.p.}}}$ and $\omega_c = \frac{eB}{mc}$.
After A_k, $\frac{B_k}{\omega_c}$, Φ, and ω_c are independently linearly interpolated, $v_\parallel^2$ has the form

$$v_\parallel^2 = 2 \cdot (U_0 + x^i\frac{\partial U}{\partial x^i}), \tag{A.4}$$

where U_0 is the quantity U at the origin of the coordinates, namely the vertex with index-number 1. As shown in section (4.2), both U_0 and $\frac{\partial U}{\partial x^i}$ are constant in a cell.
Since equation (A.4) is linear in x^i, the extrema of $v_\parallel^2$ must lie in the vertices of the tetrahedron.
Now we are interested in the extrema of $v_\parallel$. If $v_\parallel^2$ is negative, $v_\parallel$ is imaginary and this is why there is no real solution of the orbit in this subvolume of the tetrahdron. When the particle moves to a region, where $v_\parallel^2 = 0$ and therefore also $v_\parallel = 0$, the orbit

will make a turn and $v_\parallel$ will change in sign. Depending of the sign of $v_{\parallel,\text{e.p.}}$ at the orbit entry point of the tetrahedron and the sign of the extrema of $v_\parallel^2$, the following cases have to be distinguished:

1. $v_\parallel^2(x^i_{vertexj}) > 0 \qquad \text{for } j = 1 \ldots 4 \qquad (v_\parallel^2 \text{ is positive in all 4 vertices.})$

   The square root of $v_\parallel^2$ cannot become zero or negative and therefore the sign of $v_\parallel$ is conserved. The extrema of $v_\parallel$ depend on the sign of $v_\parallel$ at the entry point.

   (a) $v_{\parallel,e.p.} > 0$

   $$min(v_\parallel) = +\sqrt{\min_j(v_\parallel^2(x^i_{vertexj}))}$$
   $$max(v_\parallel) = +\sqrt{\max_j(v_\parallel^2(x^i_{vertexj}))}$$

   (b) $v_{\parallel,e.p.} < 0$

   $$min(v_\parallel) = -\sqrt{\max_j(v_\parallel^2(x^i_{vertexj}))}$$
   $$max(v_\parallel) = -\sqrt{\min_j(v_\parallel^2(x^i_{vertexj}))}$$

2. $v_\parallel^2(x^i_{vertexj}) < 0 \qquad \text{for } j = 1 \ldots 4 \qquad (v_\parallel^2 \text{ is negative in all 4 vertices.})$

   There is no real solution of $v_\parallel$ in the whole tetrahedron.

3. $v_\parallel^2(x^i_{vertexj}) < 0$ in at least 1 vertex and $v_\parallel^2(x^i_{vertexj}) > 0$ in at least 1 vertex.

   The square root of $v_\parallel^2$ can become zero or negative and therefore the sign of $v_\parallel$ is not conserved.

   $$min(v_\parallel) = -\sqrt{\max_j(v_\parallel^2(x^i_{vertexj}))}$$
   $$max(v_\parallel) = +\sqrt{\max_j(v_\parallel^2(x^i_{vertexj}))}$$

Thus, if for the approximation of τ, $v_\parallel = v_{\parallel,const.}$ is set to a constant, $v_{\parallel,const.}$. must lie inside the range $[min(v_\parallel), max(v_\parallel]$.

Extrema of x^l:
It is also necessary for the integration of the set of ODEs to set another implicitly τ-dependent variable constant: $x^l = x^l_{const.}$. In this case it is easy to analyse, which values can be taken by $x^l_{const.}$. Obviously only coordinates x^l, that are inside the tetrahedron are allowed as constants. Thus, the extrema of x^l are given by

$$\begin{aligned} \min\left(x^l\right) &= \min_j\left(x^i_{vertexj}\right), \\ \max\left(x^l\right) &= \max_j\left(x^i_{vertexj}\right), \end{aligned} \tag{A.5}$$

and $x^l_{const.}$ must lie inside the range $[\min\left(x^l\right), \max\left(x^l\right)]$.

A.2.2 Extrema of the quadratic equations coefficients a^α and b^α

To find the extrema of τ^α one has to examine the extrema of the quadratic equations coefficients a^α and b^α.

Extrema of the quadratic equations coefficients a^α:
By assuming, that the constant $v_{\parallel,const.}$ is variable again, we get

$$\begin{aligned} a^\alpha(v_{\parallel,const.}) \to a^\alpha(v_\parallel) &= n^\alpha_i a^i_4 (a^4_4 \cdot v_\parallel + b^4) \\ &= \underbrace{n^\alpha_i a^i_4 b^4}_{a^\alpha_0} + \underbrace{v_\parallel \cdot n^\alpha_i a^i_4 a^4_4}_{\delta a^\alpha} \end{aligned} \tag{A.6}$$

which is a linear function in $v_\parallel$.
Thus $a^\alpha(v_\parallel)$ is extremal, if the function is evaluated at the extrema of $v_\parallel$:

$$\begin{aligned} v_{\parallel,j} &= \begin{pmatrix} \min(v_\parallel) \\ \max(v_\parallel) \end{pmatrix} \\ \min(a^\alpha) &= n^\alpha_i a^i_4 b^4 + \min_j(v_{\parallel,j} \cdot n^\alpha_i a^i_4 a^4_4) = a^\alpha_0 + \min(\delta a^\alpha) \\ \max(a^\alpha) &= n^\alpha_i a^i_4 b^4 + \max_j(v_{\parallel,j} \cdot n^\alpha_i a^i_4 a^4_4) = a^\alpha_0 + \max(\delta a^\alpha) \end{aligned} \tag{A.7}$$

Extrema of the quadratic equations coefficients b^α:
By making the constant $x^l_{const.}$ variable again, we get the function

$$\begin{aligned} b^\alpha(x^l_{const.}) \to b^\alpha(x^l) &= n^\alpha_i a^i_l\, x^l + n^\alpha_i (a^i_4 v_{\parallel,e.p.} + b^i) \\ &= \underbrace{n^\alpha_i b^i + n^\alpha_i a^i_4 v_{\parallel,e.p.}}_{b^\alpha_0} + \underbrace{n^\alpha_i a^i_l x^l}_{\delta b^\alpha}, \end{aligned} \tag{A.8}$$

which is linear in x^l.
On account of the linearity of this equation, the extrema must lie in the vertices of the tetrahedron:

$$\begin{aligned} \min(b^\alpha) &= n^\alpha_i (a^i_4 v_{\parallel,e.p.} + b^i) + \min_j (n^\alpha_i a^i_l\ x^l_{vertex\,j}) = b^\alpha_0 + \min(\delta b^\alpha) \\ \max(b^\alpha) &= n^\alpha_i (a^i_4 v_{\parallel,e.p.} + b^i) + \max_j (n^\alpha_i a^i_l\ x^l_{vertex\,j}) = b^\alpha_0 + \max(\delta b^\alpha) \end{aligned} \tag{A.9}$$

A.2.3 Dependence of the solution for τ^α on the coefficients a^α and b^α of the quadratic equation

As it was shown in section (5.2.3), the smallest positive solution for τ^α can be found among the following solutions of the quadratic equation:

$$\begin{aligned} \tau^\alpha_{c=0} &= -\frac{2b^\alpha}{a^\alpha} \quad \text{for} \quad c = 0, \quad \text{or} \\ \tau^\alpha_{c>0} &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \quad \text{for} \quad a^\alpha < 0, \quad \text{or} \\ \tau^\alpha_{c>0} &= \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha} \quad \text{for} \quad a^\alpha > 0 \quad \text{and} \quad b^\alpha < 0. \end{aligned} \tag{A.10}$$

Now we are interested, how the solutions τ^α are changing, when the coefficients a^α and b^α are varied. Is τ^α getting smaller or larger, when a^α and b^α are varied?

The partial derivatives of $\tau^\alpha_{c>0}$ with respect to a^α and b^α are

$$\begin{aligned} \frac{\partial \tau^\alpha_{c>0}}{\partial a^\alpha} &= \frac{b^\alpha \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha} - a^\alpha c^\alpha + (b^\alpha)^2}{(a^\alpha)^2 \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}} \quad \text{and} \\ \frac{\partial \tau^\alpha_{c>0}}{\partial b^\alpha} &= -\frac{\sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha} + b^\alpha}{a^\alpha \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}. \end{aligned} \tag{A.11}$$

Under the condition

$$\begin{aligned} &a^\alpha > 0 \quad , \qquad b^\alpha < 0 \quad \text{and} \quad 0 < c^\alpha < \frac{(b^\alpha)^2}{2a^\alpha} \quad \text{or} \\ &a^\alpha < 0 \quad , \quad c^\alpha > 0 \end{aligned} \tag{A.12}$$

we want to examine, if there are negative solutions of the partial derivatives

$$\begin{aligned} \frac{\partial \tau^\alpha_{c>0}}{\partial a^\alpha} &< 0 \qquad \text{and} \\ \frac{\partial \tau^\alpha_{c>0}}{\partial b^\alpha} &< 0. \end{aligned} \tag{A.13}$$

One can show that for both inequalities (A.13) under both conditions no solutions exist.

Finally also the partial derivatives for the case $c^\alpha = 0$ ($\tau^\alpha_{c=0} = -\frac{2b^\alpha}{a^\alpha}$) have to be calculated:

$$\begin{aligned} \frac{\partial \tau^\alpha_{c=0}}{\partial a^\alpha} &= \frac{2b^\alpha}{(a^\alpha)^2} \\ \frac{\partial \tau^\alpha_{c=0}}{\partial b^\alpha} &= -\frac{2}{a^\alpha} \end{aligned} \tag{A.14}$$

Under the condition

$$a^\alpha > 0, b^\alpha < 0 \tag{A.15}$$

no solutions for positive derivatives exist, whereas under the condition

$$a^\alpha < 0, b^\alpha > 0 \tag{A.16}$$

no solutions for negative derivatives exist.

A.2.4 Extrema of the quadratic equations solution τ^α

By summarizing the results of the previous sections, table A.1 is generated. The interval of the solutions for τ^α and the sign of their partial derivatives determine the extremal values of the smallest positive real solution for τ^α.

Table A.1: Extrema of τ^α for various solution intervals

$c = 0$	$a^\alpha < 0, b^\alpha > 0$	$\frac{\partial \tau^\alpha_{c=0}}{\partial a^\alpha} > 0$ $\frac{\partial \tau^\alpha_{c=0}}{\partial b^\alpha} > 0$	$\min(\tau^\alpha_{c=0}) = \tau^\alpha_{c=0}(a^\alpha_{\min}, b^\alpha_{\min})$ $\max(\tau^\alpha_{c=0}) = \tau^\alpha_{c=0}(a^\alpha_{\max}, b^\alpha_{\max})$
$\tau^\alpha_{c=0} = -\frac{2b^\alpha}{a^\alpha}$	$a^\alpha > 0, b^\alpha < 0$	$\frac{\partial \tau^\alpha_{c=0}}{\partial a^\alpha} < 0$ $\frac{\partial \tau^\alpha_{c=0}}{\partial b^\alpha} < 0$	$\min(\tau^\alpha_{c=0}) = \tau^\alpha_{c=0}(a^\alpha_{\max}, b^\alpha_{\max})$ $\max(\tau^\alpha_{c=0}) = \tau^\alpha_{c=0}(a^\alpha_{\min}, b^\alpha_{\min})$
$c > 0$	$a^\alpha > 0, b^\alpha < 0, c^\alpha < \frac{(b^\alpha)^2}{2a^\alpha}$	$\frac{\partial \tau^\alpha_{c>0}}{\partial a^\alpha} > 0$	see ①
$\tau^\alpha_{c>0} = \frac{-b^\alpha - \sqrt{(b^\alpha)^2 - 2a^\alpha c^\alpha}}{a^\alpha}$	$a^\alpha < 0$	$\frac{\partial \tau^\alpha_{c>0}}{\partial b^\alpha} > 0$	$\min(\tau^\alpha_{c>0}) = \tau^\alpha_{c>0}(a^\alpha_{\min}, b^\alpha_{\min})$ $\max(\tau^\alpha_{c>0}) = \tau^\alpha_{c>0}(a^\alpha_{\max}, b^\alpha_{\max})$

To determine the extrema of ① in table (A.1), one must consider, that only imaginary solutions for the quadratic equation exists, when the discriminant becomes negative.

Extrema of $\tau^\alpha_{c>0}$ in ①:
There is a solution of the quadratic equation for $\tau^\alpha_{c>0}$, if the coefficient c^α lies in the interval $\left(0, \frac{(b^\alpha)^2}{2a^\alpha}\right)$, otherwise the discriminant becomes negative (no real solutions). Since c^α is a constant in every tetrahedron and a^α and b^α are linear functions of $v_\parallel$ and x^l, these interval boundaries can only be exceeded, when $\frac{(b^\alpha)^2}{2a^\alpha}$ becomes smaller than c^α.
To understand, how a^α and b^α have to be varied in order to decrease $\frac{(b^\alpha)^2}{2a^\alpha}$, one has to examine the sign of the partial derivatives.

$$\begin{aligned} \frac{\partial}{\partial a^\alpha}\frac{(b^\alpha)^2}{2a^\alpha} &= -\frac{(b^\alpha)^2}{2(a^\alpha)^2} < 0 \\ \frac{\partial}{\partial b^\alpha}\frac{(b^\alpha)^2}{2a^\alpha} &= \frac{b^\alpha}{a^\alpha} < 0 \end{aligned} \tag{A.17}$$

Since both partial derivatives are negative, $\frac{(b^\alpha)^2}{2a^\alpha}$ will decrease, if a^α and b^α increase. Thus $c^\alpha = \frac{(b^\alpha)^2}{2a^\alpha}$ is an upper bound for the quadratic equation coefficients a^α and b^α. In principal we are interested in the extrema of $\tau^\alpha_{c>0}$. When the discriminant is zero, the solution for $\tau^\alpha_{c>0} = -\frac{b^\alpha}{a^\alpha}$. If we insert the upper bound ($c^\alpha = \frac{(b^\alpha)^2}{2a^\alpha} \rightarrow a^\alpha = \frac{(b^\alpha)^2}{2c^\alpha}$) into the solution, we get

$$\tau^\alpha_{c>0} = -\frac{b^\alpha}{a^\alpha} = -\frac{b^\alpha}{\frac{(b^\alpha)^2}{2c^\alpha}} = -\frac{2c^\alpha}{b^\alpha}. \tag{A.18}$$

As one can see in table (A.1), the partial derivatives of $\tau^\alpha_{c>0}$ in the considered solution interval are positive. That's why the upper bound for a^α and b^α is the upper bound

for $\tau^{\alpha}_{c>0}$, due to the discriminant that becomes zero.

$$\tau^{\alpha}_{\text{max-discr}} = \frac{2c^{\alpha}}{b^{\alpha}_{\text{max}}}. \tag{A.19}$$

Moreover it is also possible to rearrange $c^{\alpha} = \frac{(b^{\alpha})^2}{2a^{\alpha}} \rightarrow b^{\alpha} = \pm\sqrt{2a^{\alpha}c^{\alpha}}$ and insert the negative b^{α} (see table A.1) into the solution, then we get

$$\tau^{\alpha}_{c>0} = -\frac{b^{\alpha}}{a^{\alpha}} = -\frac{\sqrt{2a^{\alpha}c^{\alpha}}}{a^{\alpha}}. \tag{A.20}$$

Since $\frac{\partial}{\partial a^{\alpha}}\left(-\frac{\sqrt{2a^{\alpha}c^{\alpha}}}{a^{\alpha}}\right) < 0$, the maximum of $\tau^{\alpha}_{c>0}$ can be found, when a^{α} becomes minimal:

$$\tau^{\alpha}_{\text{max-discr}} = \frac{\sqrt{2a^{\alpha}_{\text{min}}c^{\alpha}}}{a^{\alpha}_{\text{min}}} \tag{A.21}$$

One must keep in mind, that $a^{\alpha}_{\text{min}} > 0$. When a^{α} and b^{α} decrease, $\frac{(b^{\alpha})^2}{2a^{\alpha}}$ increases and the interval for c^{α} becomes larger. There is no lower bound for a^{α} and b^{α} due to the discriminant, that can become negative.
Furthermore the extrema of a^{α} and b^{α} must still be taken into account, when one is looking for the extrema of $\tau^{\alpha}_{c>0}$.

$$\begin{aligned} \tau^{\alpha}_{\text{minval}} &= \tau^{\alpha}_{c>0}(a^{\alpha}_{\text{min}}, b^{\alpha}_{\text{min}}) \\ \tau^{\alpha}_{\text{maxval}} &= \tau^{\alpha}_{c>0}(a^{\alpha}_{\text{max}}, b^{\alpha}_{\text{max}}) \end{aligned} \tag{A.22}$$

In conclusion the extrema of ① in table (A.1) are

$$\begin{aligned} \min(\tau^{\alpha}_{c>0}) &= \tau^{\alpha}_{c>0}(a^{\alpha}_{\text{min}}, b^{\alpha}_{\text{min}}) \\ \max(\tau^{\alpha}_{c>0}) &= \max\left(\tau^{\alpha}_{c>0}(a^{\alpha}_{\text{max}}, b^{\alpha}_{\text{max}}), -\frac{2c^{\alpha}}{b^{\alpha}_{\text{max}}}\right). \end{aligned} \tag{A.23}$$

An equivalent formulation of the maximum of $\tau^{\alpha}_{c>0}$ is

$$\max(\tau^{\alpha}_{c>0}) = \max\left(\tau^{\alpha}_{c>0}(a^{\alpha}_{\text{max}}, b^{\alpha}_{\text{max}}), \frac{\sqrt{2a^{\alpha}_{\text{min}}c^{\alpha}}}{a^{\alpha}_{\text{min}}}\right). \tag{A.24}$$

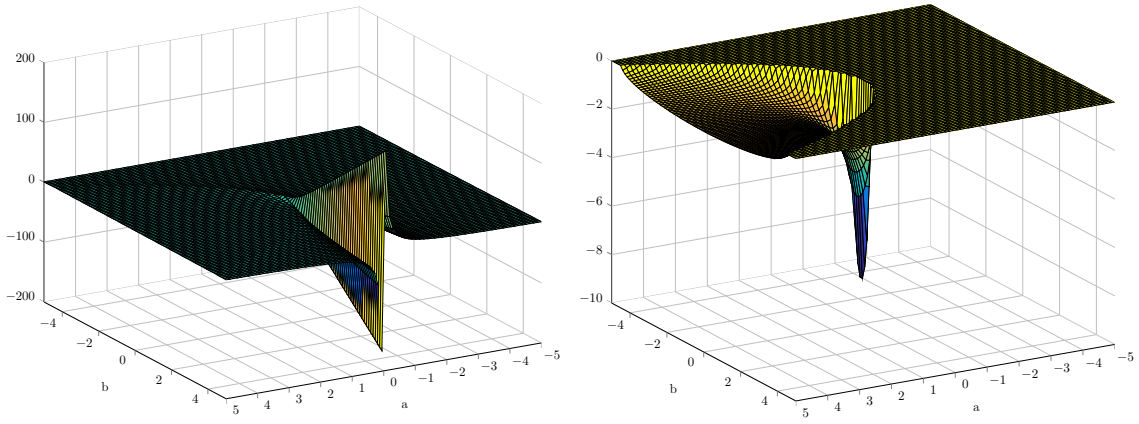
A.2.5 Extrema of τ^α under the consideration of the sign of a^α and b^α

As it was shown in section (A.2.2) a^α and b^α are linear functions of $v_\parallel$ and x^l:

$$\begin{aligned} a^\alpha(v_\parallel) &= n_i^\alpha \cdot a_4^i(a_4^4 \cdot v_\parallel + b^4) = a_0^\alpha + \delta a^\alpha, \\ b^\alpha(x^l) &= n_i^\alpha a_l^i \; x^l + n_i^\alpha(a_4^i v_{\parallel,e.p.} + b^i) = b_0^\alpha + \delta b^\alpha \end{aligned} \tag{A.25}$$

For this reason the sign of a^α and b^α can be both, positive and negative. Thus, also the sign of $a^\alpha_{\min}$, $a^\alpha_{\max}$, $b^\alpha_{\min}$ and $b^\alpha_{\max}$ can be positive or negative.
In order to evaluate the extrema of τ^α under a given $a^\alpha_{\min}$, $a^\alpha_{\max}$, $b^\alpha_{\min}$ and $b^\alpha_{\max}$ with various signs, the extrema for all possible combinations must be taken into consideration.
In principal there are 3 combinations of signs for $a^\alpha_{\min}$ and $a^\alpha_{\max}$. (The combination $\sigma(a^\alpha_{\min}) = +1$ and $\sigma(a^\alpha_{\max}) = -1$ is not possible.) These 3 combinations can be combined with 3 combinations of signs for $b^\alpha_{\min}$ and $b^\alpha_{\max}$, which gives in total 9 combinations.

Extrema of the solutions for $\tau^\alpha_{c>0}$ under various signs of a^α and b^α:
In figure (A.1) one can see the real and imaginary part of the solution of the quadratic equation for $c > 0$. Table (A.1) in a previous section shows the extrema of τ^α for various solution intervals. A given $a^\alpha_{\min}$ can for instance belong to a different solution interval than $a^\alpha_{\max}$. Not all of these solution intervals are continuously connected to another solution interval. In some cases one can observe singularities, when $a^\alpha \to 0$.



(a) Real part of the solution

(b) Imaginary part of the solution

Figure A.1: The solution $\tau^{\alpha}_{c>0} = \frac{-b^{\alpha} - \sqrt{(b^{\alpha})^2 - 2a^{\alpha}c^{\alpha}}}{a^{\alpha}}$ of the quadratic equation for $a^{\alpha} = (-10, 10)$, $b^{\alpha} = (-10, 10)$ and $c^{\alpha} = +2$ is depicted. As one can easily see in figure (a) the real part of the solution has a singularity at $a^{\alpha} \to 0$ for positive values of b^{α}. The solution, where the discriminant is zero $\left(c^{\alpha} = \frac{(b^{\alpha})^2}{2a^{\alpha}}\right)$, is an upper bound for a^{α} and b^{α}. In the region, where both a^{α} and b^{α} are positive, there are only negative solutions for τ^{α}_2 and therefore no physical solutions exist.

Table (A.2) lists the extrema of $\tau^{\alpha}_{c>0}(a^{\alpha}, b^{\alpha})$ under consideration of the combination of different signs of the quadratic equation coefficients a^{α} and b^{α}. The plot in figure (A.1) helps to understand, why for a certain combination of a^{α}_{min}, a^{α}_{max}, b^{α}_{min} and b^{α}_{max} the maximum value of $\tau^{\alpha}_{c>0}$ becomes infinite.

Table A.2: Extrema of the solutions for $\tau^{\alpha}_{c>0}$ under various signs of a^{α} and b^{α}

$\sigma(a^{\alpha}_{\min})$	$\sigma(a^{\alpha}_{\max})$	$\sigma(b^{\alpha}_{\min})$	$\sigma(b^{\alpha}_{\max})$	extrema of $\tau^{\alpha}_{c>0}$
$+1$	$+1$	$+1$	$+1$	no solution no solution
$+1$	$+1$	-1	$+1$	$\min(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = \max\left(\tau^{\alpha}_{c>0}(a^{\alpha}_{max}, b^{\alpha}_{max}), \frac{\sqrt{2a^{\alpha}_{\min} c^{\alpha}}}{a^{\alpha}_{\min}}\right)$
$+1$	$+1$	-1	-1	$\min(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = \max\left(\tau^{\alpha}_{c>0}(a^{\alpha}_{max}, b^{\alpha}_{max}), -\frac{2c^{\alpha}}{b^{\alpha}_{max}}\right)$
-1	$+1$	$+1$	$+1$	$\min(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = +\infty$
-1	$+1$	-1	$+1$	$\min(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = +\infty$
-1	$+1$	-1	-1	$\min(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = \max\left(\tau^{\alpha}_{c>0}(a^{\alpha}_{max}, b^{\alpha}_{max}), -\frac{2c^{\alpha}}{b^{\alpha}_{max}}\right)$
-1	-1	$+1$	$+1$	$\min(\tau^{\alpha}_{2}) = \tau^{\alpha}_{2}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$
-1	-1	-1	$+1$	$\min(\tau^{\alpha}_{2}) = \tau^{\alpha}_{2}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$
-1	-1	-1	-1	$\min(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c>0}) = \tau^{\alpha}_{c>0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$

Extrema of the solutions for $\tau^{\alpha}_{c=0}$ under various signs of a^{α} and b^{α}:
In figure (A.2) one can see the solution of the quadratic equation for $c = 0$. Table (A.3) lists the extrema of $\tau^{\alpha}_{c=0}(a^{\alpha},\, b^{\alpha})$.

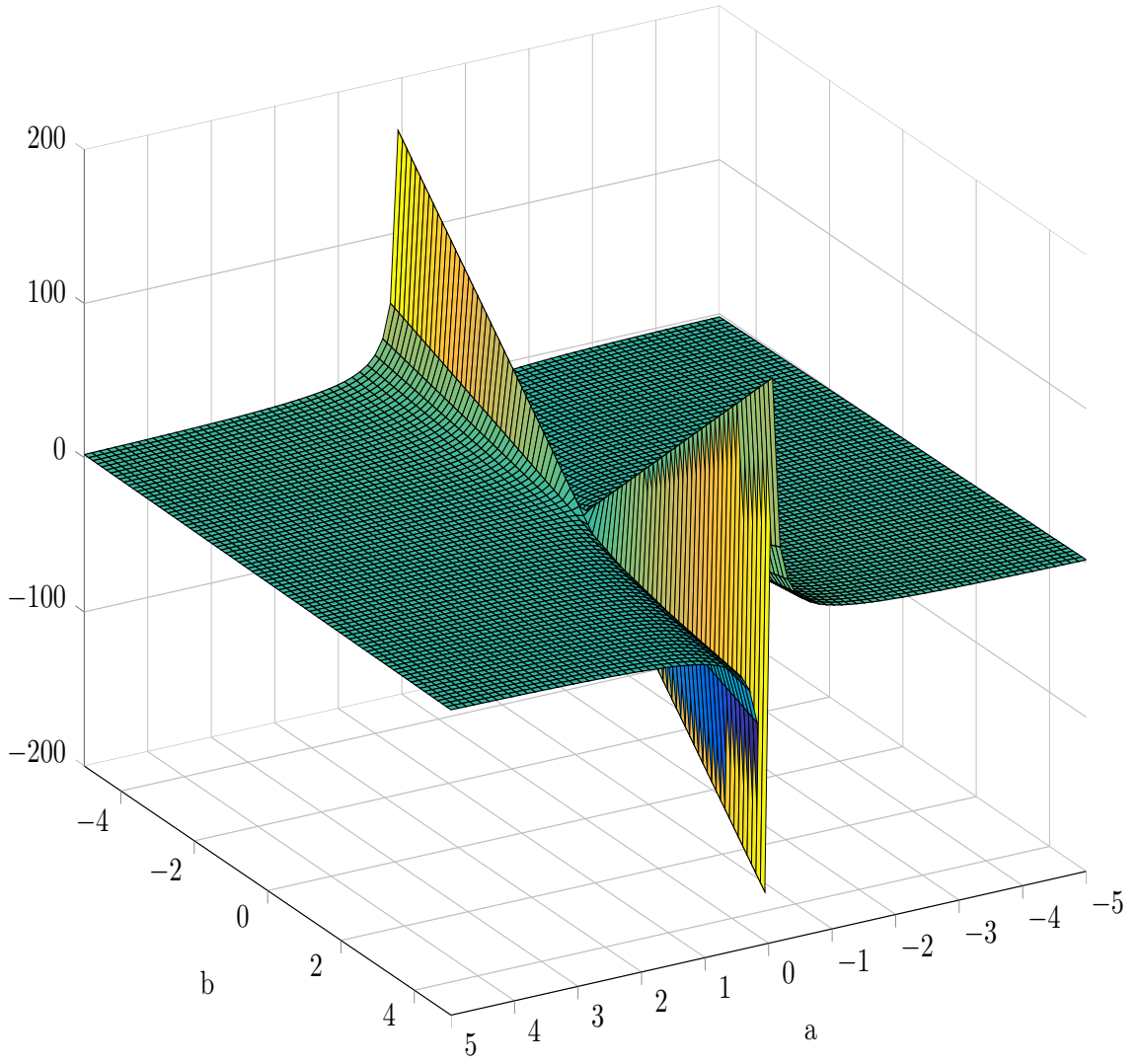


Figure A.2: The solution $\tau^{\alpha}_{c=0} = \frac{-2b^{\alpha}}{a^{\alpha}}$ of the quadratic equation for $a^{\alpha} = (-10, 10)$, $b^{\alpha} = (-10, 10)$ is depicted. As one can easily see the solution has a singularity at $a^{\alpha} \to 0$ for both positive and negative values of b^{α}. In the regions, where either both a^{α} and b^{α} are positive or negative, there are only negative solutions for $\tau^{\alpha}_{c=0}$ and therefore no physical solutions exist.

Table A.3: Extrema of the solutions for $\tau^{\alpha}_{c=0}$ under various signs of a^{α} and b^{α}

$\sigma(a^{\alpha}_{\min})$	$\sigma(a^{\alpha}_{\max})$	$\sigma(b^{\alpha}_{\min})$	$\sigma(b^{\alpha}_{\max})$	extrema of $\tau^{\alpha}_{c=0}$
$+1$	$+1$	$+1$	$+1$	no solution no solution
$+1$	$+1$	-1	$+1$	$\min(\tau^{\alpha}_{c=0}) = 0$ $\max(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$
$+1$	$+1$	-1	-1	$\min(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$ $\max(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$
-1	$+1$	$+1$	$+1$	$\min(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c=0}) = +\infty$
-1	$+1$	-1	$+1$	$\min(\tau^{\alpha}_{c=0}) = 0$ $\max(\tau^{\alpha}_{c=0}) = +\infty$
-1	$+1$	-1	-1	$\min(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$ $\max(\tau^{\alpha}_{c=0}) = +\infty$
-1	-1	$+1$	$+1$	$\min(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\min}, b^{\alpha}_{\min})$ $\max(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$
-1	-1	-1	$+1$	$\min(\tau^{\alpha}_{c=0}) = 0$ $\max(\tau^{\alpha}_{c=0}) = \tau^{\alpha}_{c=0}(a^{\alpha}_{\max}, b^{\alpha}_{\max})$
-1	-1	-1	-1	no solution no solution

A.3 Application to *3DGeoInt*

The obtained values for τ^{α} can be introduced to all analytical approximations in the iteration procedure, presented in section 5.5. One has to compare, the usually calculated values for τ^{α} in the iteration procedure, with the more sophisticated values, that were presented in this chapter. If discrepancies occur, meaning that the sophisticated approach provides additional values for τ^{α}, these values are proposed in the iteration procedure as the "best guess" for τ^{α}, obviously on condition that τ^{α} is the smallest positive value.

Bibliography

[1] U. Stroth. *Plasmaphysik*. Vieweg+Teubner Verlag, 2011.

[2] S.V. Kasilov, A.M. Runov and W. Kernbichler. Geometric integrator for charged particle orbits in axisymmetric fusion devices. *Computer Physics Communications*, 207 (2016) 282-286.

[3] G. Jost, T.M. Tran, W.A. Cooper, L. Villard and K. Appert, *Phys. Plasmas*, 8 (2001) 3321.

[4] V. Kornilov, R. Kleiber, R. Hatzky, L. Villard and G. Jost, *Phys. Plasmas*, 11 (2004) 3196.

[5] J.A. Heikkinen, T.P. Kiviniemi, T. Kurki-Suonio, A.G. Peeters and S.K. Sipilä, *J. Comput. Phys.*, 173 (2001) 527.

[6] A.M. Runov, S.V. Kasilov and P. Helander, *J. Comput. Phys.*, 300 (2015) 605.

[7] C.G. Albert *et al.* Kinetic modeling of 3D equilibria in a tokamak. *J. Phys.: Conf. Ser.*, 775 012001, 2016.

[8] A.H. Boozer and G. Kuo-Petravic, *Phys. Fluids*, 24 (1981) 851.

[9] A.I. Morozov, L.S. Solov'ev, in: M.A. Leontovitch (Ed.), *Reviews of Plasma Physics*, vol. 2, Consultants Bureau, New York, 1966.

[10] R.J. Littlejohn, *J. Plasma Phys.*, 29 (1983) 111.

[11] R.I. McLachlan, G. Reinout and W.Quispel. Geometric Integrators for ODEs. *Journal of Physics: A General Physics*, 39(19):5251-5285, 2006.

[12] N. Takemoto. Symplectic Integrator [Internet]. *Wikipedia, The Free Encyclopedia*. 2018 Mar 2, 20:49 UTC [cited 2018 Jun 29]. Available from: `https://en.wikipedia.org/wiki/Symplectic_integrator`.

[13] J.M. Sanz-Serna and M.P. Calvo. *Applied Mathematics and Mathematical Computation*. Chapman & Hall, 1994.

[14] P. Birken. Klassisches Runge-Kutta-Verfahren [Internet]. *Wikipedia, The Free Encyclopedia*. 2017 Nov 26, 18:26 [cited 2018 Jun 29]. Available from: `https://de.wikipedia.org/wiki/Klassisches_Runge-Kutta-Verfahren`.

[15] P. Helander and D.J. Sigmar. *Collisional Transport in Magnetized Plasmas*. Cambridge University Press, 2002.

[16] M.F. Heyn. *Physics of Drift Orbits*. Plasmaphysik. Technische Universität Graz, 2016.

[17] M.F. Heyn, I.B. Ivanov, S.V. Kasilov, W. Kernbichler, P. Leitner, V.V. Nemov, W. Suttrop. Quasilinear modelling of RMP interaction with a tokamak plasma: Application to ASDEX Upgrade ELM mitigation experiments. *Nuclear Fusion*, 54:064005, 2014.

[18] Max-Planck-Insitut für Plasmaphysik. ASDEX Upgrade. Boltzmannstraße 2, Garching.

[19] C. Albert. *Hamiltonian Approach to Resonant Transport Regimes due to Non-Axisymmetric Magnetic Perturbations in Tokamak Plasmas*. PhD thesis, Technische Universität Graz, 2017.

[20] W. Suttrop *et al.* Studies of edge localized mode mitigation with new active in-vessel saddle coils in ASDEX Upgrade. *Plasma Phys. Control Fusion*, 53:124014, 2011.

[21] D. Boffi, F. Brezzi and M. Fortin. *Mixed Finite Element Methods and Applications*. Springer-Verlag, 2013.

[22] I. Joseph *et al.* Calculation of stochastic thermal transport due to resonant magnetic perturbations in DIII-D. *Nuclear Fusion*, 48:045009, 2008.

[23] I. Joseph *et al.* Stochastic transport modeling of resonant magnetic perturbations in DIII-D. *Journal of Nuclear Materials*, 363–365 (2007) 591–595.

[24] M.F. Heyn, S.V. Kasilov *et al.* Kinetic estimate of the shielding of resonant magnetic field perturbations by the plasma in DIII-D. *Nuclear Fusion*, 48:024005, 2008.

[25] H.W. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *Numerical Recepies in Fortran 77: The Art of Scientific Computing*. Press Syndicate of the University of Cambridge, 1992.