Dipl.-Ing. Michael Eder, BSc

On the integration of the guiding-center equations in toroidal fields utilizing a local linearization approach

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to
Graz University of Technology

Supervisor
Ao.Univ.-Prof.i.R. Dipl.-Ing. Dr.phil. Martin Heyn

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

Institute of Theoretical and Computational Physics

Graz, June 2021

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 doctoral thesis.

21 June 2021 

Date, Signature

Abstract

In magnetic confinement fusion research the numerical simulation of collective plasma behavior is of high interest. For this purpose, direct modeling of charged particle orbits is utilized in important kinetic approaches. The reduced dynamics of such a charged particle in a slowly-varying electromagnetic field is described by the guiding-center equations of motion. In particular, the relatively fast circular motion around a point called the guiding-center is separated and primarily the comparatively slow drift motion of this point is treated.
In the present work, "a numerical integration method for guiding-center orbits of charged particles in toroidal fusion devices with three-dimensional field geometry is described. Here, high order interpolation of electromagnetic fields in space is replaced by a special linear interpolation leading to locally linear Hamiltonian equations of motion with piecewise constant coefficients. This approach reduces computational effort and noise sensitivity while the conservation of total energy, magnetic moment and phase-space volume is retained. The underlying formulation treats motion in piecewise linear fields exactly and thus preserves the non-canonical symplectic form. The algorithm itself is only quasi-geometric due to a series expansion in the orbit parameter. For practical purposes an expansion to the fourth order retains geometric properties down to computer accuracy in typical examples. When applied to collisionless guiding-center orbits in an axisymmetric tokamak and a realistic three-dimensional stellarator configuration, the method demonstrates stable long-term orbit dynamics conserving invariants. In Monte Carlo evaluation of transport coefficients, the computational efficiency of quasi-geometric integration is an order of magnitude higher than with a standard fourth order Runge-Kutta integrator."[1]
Moreover, a brief review of the Lagrangian and Hamiltonian description of classical mechanics is given and the derivation of guiding-center dynamics is presented. Further, the implementation of the guiding-center orbit code GORILLA, which is based on this integration method, is outlined and the integration method is tested for the computation of fusion alpha losses in a realistic stellarator configuration.

Acknowledgements

My sincere gratitude goes to my supervisor *Martin Heyn* 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 deepest gratitude to my co-supervisors *Winfried Kernbichler*, *Sergei Kasilov* and *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. Working in the plasma group of the Institute of Theoretical and Computational Physics under the lead of *Winfried Kernbichler* was characterized by an exceptional team spirit and a deep personal manner of support. My gained experience possesses a priceless value both professionally and personally.
Furthermore, I would like to thank my dear friend *Lukas Bauer* for his valuable contributions to this work. It was a pleasure to co-supervise his excellent Master's thesis and I am grateful for the heartfelt memory of the fruitful cooperation. As one can learn a lot by teaching, I owe a great deal of insight to the students with whom I had the pleasure to work together, especially *Michael Scheidt*, *Markus Meisterhofer* and *Georg Graßler*.
Moreover, I would like to thank *Michael Drevlak*, *Artem Savchenko* and *Harry Mynick* for respectively providing stellarator field configurations, a tool for non-linear coordinate transformations and a tutorial for neoclassical transport in 3D systems, as well as *Alex Runov*, *Philipp Ulbl, Rico Buchholz*, *Patrick Lainer* and *Markus Richter* for useful discussions. Special thanks go to *John R. Cary* for very valuable suggestions that lead to significant improvements of this thesis.
Finally, I would like to thank my parents, *Theresia* and *Johann*, my sister *Christina*, and my friends with all my heart for their continuous support and encouragement and most importantly my wife *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 programmes 2014-2018 and 2019-2020 under Grant Agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. Support from NAWI Graz, and from the OeAD under the WTZ Grant Agreement with Ukraine No. UA 04/2017 is gratefully acknowledged.

Contents

Publications and work associated with this thesis

Parts of this thesis have already been published or are in preparation for publication.

Publications as first author

- **Ref. [1]:** M. Eder, C.G. Albert, L.M.P. Bauer, S.V. Kasilov and W. Kernbichler
  "Quasi-geometric integration of guiding-center orbits in piecewise linear toroidal fields"
  Physics of Plasmas **27**, 122508 (2020)
  `https://doi.org/10.1063/5.0022117`

  Author contributions: M. Eder drafted and wrote the manuscript in collaboration with C.G. Albert and S.V. Kasilov. Based on preliminary work by S. V. Kasilov and W. Kernbichler, M. Eder formulated the derivation of locally linear Hamiltonian guiding-center equations and developed the numerical solution. The development and implementation of the guiding-center orbit code GORILLA was performed by M. Eder with support of L. M. P. Bauer. The literature review in the introduction was guided by C.G. Albert with support of all authors. All numerical computations and physical applications were executed by M. Eder, including the creation and design of figures. Appendix A was formulated by S. V. Kasilov with support of M. Eder, Appendix B was formulated by M. Eder with support of S. V. Kasilov.

- **Ref. [2]:** M. Eder, C.G. Albert, L.M.P. Bauer, S.V. Kasilov, W. Kernbichler, M. Meisterhofer and M.Scheidt
  "GORILLA: Guiding-center ORbit Integration with Local Linearization Approach"
  Public GitHub repository: `https://github.com/itpplasma/GORILLA`
  Zenodo: `https://doi.org/10.5281/zenodo.4593661`
  submitted to Journal of Open Source Software (2021)

Author contributions: M. Eder drafted and wrote the manuscript. The associated computer code GORILLA was developed and implemented by M. Eder and L. M. P. Bauer with support of M. Meisterhofer, C.G. Albert and S.V. Kasilov. The underlying numerical method was derived in a collaboration of M. Eder with C.G. Albert, S. V. Kasilov and W. Kernbichler. M. Eder and L. M. P. Bauer formulated the code's documentation. M.Scheidt prepared the Matlab examples. M. Eder and C.G. Albert arranged the GitHub repository and the installation instructions.

- **Ref. [3]:** M. Eder, C.G. Albert, S.V. Kasilov, W. Kernbichler and M. Scheidt "Integration of the guiding-center equations in toroidal fields utilizing a local linearization approach"
  47^{th} EPS Conference on Plasma Physics, Sitges, P3.1059 (2021):

  Author contributions: M. Eder drafted and wrote the manuscript in collaboration with S.V. Kasilov. Based on preliminary work by S. V. Kasilov and W. Kernbichler, M. Eder formulated the summary of locally linear Hamiltonian guiding-center equations and the method's application to a realistic stellarator configuration. The description of local magnetic coordinates was performed by S. V. Kasilov. The method's application to the confinement computation of fusion alphas in a realistic stellarator configuration was performed by M. Eder with support of M. Scheidt and C. G. Albert. All numerical computations were executed by M. Eder, including the creation and design of figures.

- **Ref. [4]:** M. Eder, C.G. Albert, L.M.P. Bauer, S.V. Kasilov and W. Kernbichler "Three-dimensional geometric integrator for charged particle orbits in toroidal fusion devices"
  46^{th} EPS Conference on Plasma Physics, Milano, ECA Vol. **43C**, P5.1100 (2019):
  `http://ocs.ciemat.es/EPS2019PAP/pdf/P5.1100.pdf`

  Author contributions: M. Eder drafted and wrote the manuscript in collaboration with C.G. Albert, L. M. P. Bauer and S.V. Kasilov. Based on preliminary work by S. V. Kasilov and W. Kernbichler, M. Eder formulated the derivation of locally linear Hamiltonian guiding-center equations and developed the numerical solution. The development and implementation of the prototype guiding-center orbit code was performed by M. Eder with support of L. M. P. Bauer. All numerical computations were executed by M. Eder, including the creation and design of figures.

Conferences and public outreach contributions

Oral contributions

- **M. Eder**
  "Integration of the guiding-center equations in toroidal fields utilizing a local linearization approach"
  Theory seminar, Princeton Plasma Physics Laboratory, Princeton University, virtual (2021)

- **M. Eder**
  "GORILLA: Quasi-geometric integration of guiding-center orbits in piecewise linear toroidal fields"
  DocDay February 2021, Physics doctoral seminar, University of Technology Graz, virtual (2021)

- **M. Eder**
  "GORILLA: Guiding-center ORbit Integration with Local Linearization Approach"
  FuseNet PhD Event, virtual (2020)

- **M. Eder**
  "GORILLA: Quasi-geometric integration of guiding-center orbits in piecewise linear toroidal fields"
  Numerical Methods for the Kinetic Equations of Plasma Physics (NumKin 2020), Max Planck Institute for Plasma Physics Garching, virtual (2020)

- **M. Eder**, L.M.P. Bauer
  "Geometric Integration of Guiding Center Orbits in Magnetically Confined Plasmas and Applications"
  6^{th} Fusion Day, Fusion@ÖAW, Salzburg (2019)

- **M. Eder**
  "Trajektorien von geladenen Teilchen in Kernfusionsreaktoren"
  Österreichisches Studienförderungswerk PRO SCIENTIA, Graz (2019)

- **M. Eder**
  "Geometric and Symplectic Integration of Guiding Center Orbits in Magnetically Confined Plasmas"
  5^{th} Fusion Day, Fusion@ÖAW, Vienna (2018)

Posters

- **M. Eder**, C.G. Albert, S.V. Kasilov, W. Kernbichler and M. Scheidt
  "Integration of the guiding-center equations in toroidal fields utilizing a local linearization approach"
  47^{th} EPS Conference on Plasma Physics, Sitges, virtual (2021)

- **M. Eder**, C.G. Albert, L.M.P. Bauer, S.V. Kasilov and W. Kernbichler
  "Three-dimensional geometric integrator for charged particle orbits in toroidal fusion devices"
  46^{th} EPS Conference on Plasma Physics, Milano (2019)

Co-supervised Bachelor's and Master's theses

- Michael Scheidt
  Preliminary: "Development of test cases for the guiding-center orbit code GORILLA and application to high-energetic particles."
  TU Graz, Master's Thesis, (2021)

- Lukas Bauer
  "Field-aligned mesh generation and enhancements of the guiding-center orbit code GORILLA"
  TU Graz, Master's Thesis, (2020)

- Georg Sebastian Graßler
  "Einführung in die nichtlineare Mechanik periodischer, hamiltonscher Systeme"
  TU Graz, Bachelor's Thesis, (2019)

Introduction

The content of this chapter can also be found in section I of Ref. [1] and Ref. [2] formulated by the author and including some modifications.

"Extremely hot plasmas with a temperature of the order of hundred million degrees Celsius are needed to produce energy from nuclear fusion. Under these conditions, hydrogen isotopes are fused, and energy is released. The energy release from 1 kg of fusion fuel corresponds approximately to that of 10000 tons of coal. A future use of this energy source is the subject of worldwide research projects. The confinement of such hot plasmas, however, poses major physical and technological problems for researchers. In particular, complex numerical methods are necessary to understand the physics of such plasmas in complicated toroidal magnetic fields.
An important kinetic approach for simulating the collective behavior of a plasma utilizes direct modeling of particle orbits. A well-known approximation for computing the motion of electrically charged particles in slowly-varying electromagnetic fields is to reduce the dynamical equations by separating the relatively fast circular motion around a point called the guiding-center, and primarily treat the relatively slow drift motion of this point. This drift motion is described by the guiding-center equations; see, e.g., Ref. [5, 6] and [7]."[2]
"In particular, global kinetic computations of quasi-steady plasma parameters in 3D toroidal fusion devices utilize the evaluation of the distribution function and/or its moments by direct modeling of particle orbits. This includes Monte Carlo transport simulations in given external fields[8–16] as well as self-consistent turbulence models with particle codes.[17–20] Kinetic modeling of 3D plasma equilibria[21] or edge plasmas[22] puts specific requirements on solving the guiding-center equations.[5–7,23] Namely, particle orbit integration should be computationally efficient, tolerant to statistical noise in the electromagnetic field and efficient in scoring statistical data from the orbits. Geometric integrators[24,25] address these targets by relaxing the requirement to the accuracy of guiding-center orbits while preserving physically consistent long time orbit dynamics. In this context the word geometric refers to the preservation of the geometric structure of phase-space being a symplectic manifold. The best-known class of geometric integrators is the class of symplectic integrators that relies on canonical coordinates in phase-space. Symplectic integrators are not directly applicable to guiding-center

dynamics that are formulated in non-canonical coordinates. One way to circumvent this problem is the use of a (usually implicit) transformation to canonical coordinates.[26–28] Up to now such approaches rely either on magnetic flux coordinates or require a more expensive transformation of phase-space coordinates in the general case. The other well-known alternative are variational integrators.[29,30] Such integrators do not assume canonical coordinates and include symplectic integrators as a special case. Stability problems of variational integrators arise for guiding-center orbit computations due to the degeneracy of their phase-space Lagrangian. This issue has only recently been resolved.[31–33] The resulting integrators are fully implicit and/or require an augmented set of dynamical variables, and their competitiveness for practical applications is still a topic open to investigation. Yet another alternative could arise from very recent results on slow-manifold methods[34] that introduce a different construction of the guiding-center equations well suited for geometric integration.

The method presented here is geometric in both a structure-preserving and a more literal sense, as it considers intersections of orbits with a spatial mesh. The underlying formulation and discretization of fields exactly preserves the (non-canonical) Hamiltonian structure of the equations. In contrast to usual geometric integrators, this is achieved by solving the exact motion in simplified fields, i.e. the Hamiltonian flow in the original fields is approximated by the exact Hamiltonian flow in piecewise linear fields. The final algorithm is only quasi-geometric, as it relies on a series expansion in the orbit parameter. Here the term quasi-geometric means that the error can be brought below any given (computer) accuracy by a sufficiently high order. Expansions to order 3 and 4 are shown to conserve invariants extremely well for at least 10^6 toroidal turns in numerical experiments for typical fusion devices. The approach has been introduced in Ref. [4] as a generalization of the 2D geometric integrator of Ref. [35] for general 3D toroidal fields. This approach has two features useful for application in global equilibrium and transport simulations: straightforward computation of spatial distributions of macroscopic parameters and robustness in the presence of noise in field data. Moreover, the new method preserves total energy, magnetic moment and phase-space volume. In its present version the method is designed for (quasi-)static electromagnetic fields.

The integration procedure is based on a special 3D discretization of space resulting in locally linear guiding-center equations while retaining the symplectic property of the original set. Formally, such an interpolation could be denoted by Whitney forms or finite element exterior calculus in a way similar to existing work on charged particle orbits[36–38] and results in a divergence-free magnetic field. Within this discretization, vectors and scalars characterizing the electromagnetic field are represented by

continuous piecewise linear functions which reduces the cost of spatial interpolation as compared to high-order interpolation with continuous derivatives (e.g. with help of 3D-splines) required for usual integration. In return, the integration procedure requires accurate tracing of intersections of the orbit with spatial cell boundaries where the coefficients of the linear guiding-center equation set are discontinuous. However, tracing of the boundaries is also required in Monte Carlo procedures for the computation of the spatial distribution of various velocity space moments of the distribution function which are computed as path integrals over the test particle dwell time within spatial cells.

In the original implementation of Ref. [4], the linear guiding-center equation set was solved numerically by using a usual Runge-Kutta (RK) integrator. In this case, tracing of orbit intersections with spatial cell boundaries requires a few Newton iterations for the computation of the dwell time within the cell. Since the integration error of this set scales with the third power of the Larmor radius, accurate results can still be obtained by a single RK integration step over the dwell time. For the same reason, sufficient accuracy can be achieved also by using the polynomial expansion of the solution over an orbit parameter which allows to compute the dwell time and integrals of velocity powers analytically.

In the case of magnetic fields with spatial symmetry (e.g., tokamaks with toroidal symmetry) the quasi-geometric integration accurately preserves the respective (toroidal) canonical momentum. Thus, the property of such systems to ideally confine the orbits is retained. In the absence of spatial symmetry, the parallel adiabatic invariant, which is an approximate conserved quantity in stellarators, is well preserved, since the quasi-geometric integration does not lead to a significant error accumulation.

It should be noted that the piecewise linear approximation of the guiding-center equations represents field lines as polygonal chains in coordinate space. Such a representation may introduce artificial chaos in case of 3D fields when using non-field-aligned coordinate systems. In our earlier publication[4], some field line diffusion has been observed in a perturbed tokamak field which seemed to be in agreement with the quasilinear estimate assuming a continuous safety factor profile of the unperturbed field (which is, actually, not the case), and, therefore, this diffusion has been attributed to the linearization. However, further detailed studies and resulting improvements of the algorithm revealed that the level of artificial diffusion observed earlier is not connected to the intrinsic deficiency of the method. In the improved algorithm this diffusion is actually much smaller so that it provides a negligible correction to the existing neoclassical transport even for the relatively strongly perturbed non-aligned 3D fields and a coarse mesh.

The above mentioned characteristics of the method allow, in particular, its effective application to the computation of neoclassical transport coefficients using the Monte Carlo method. For demonstration, the mono-energetic neoclasscial diffusion coefficient is evaluated here for a quasi-isodynamic reactor-scale stellarator field[39]. The results and performance of the new method are compared to usual (RK) orbit integration methods."[1]

Nevertheless, the inherent artificial chaos of the method, which is induced by the linearization of the electromagnetic field, strongly scales with the particle's Larmor radius. When applying the method to the computation of fusion alpha particle losses for the slowing-down time of 1 s, moderate grid sizes are not sufficient to yield accurate results. (The Larmor radius of a 3.5 MeV alpha particle is roughly 24 times larger than that of a 3 keV D ion.) Consequently, the quasi-geometric integration method is not suited to be used for tracing of high-energetic 3.5 MeV fusion alpha particles for the total slowing-down time. In return, the quasi-geometric integration method is successfully applied for the comparatively shorter trace time of 0.01 s utilizing a moderate grid size and showing a significant CPU speed-up.

The remainder of this thesis is organized in three parts: (I) theoretical background, (II) derivations and implementation and (III) applications, results and discussion; including the following chapters. Chapter 1 gives a brief recapitulation of the Lagrangian and Hamiltonian formulation of mechanics. In chapter 2 the guiding-center Lagrangian and the resulting equations of motion are derived. In chapter 3 the spatial discretization procedure is introduced and the numerical solution of the resulting piecewise linear guiding-center equations is described. Furthermore, details on the Hamiltonian structure of the underlying locally linear equations are discussed and a useful feature for scoring orbit statistics is presented. Chapter 4 gives a concise overview of the implementation of the guiding-center orbit code GORILLA. In chapter 5 collisionless guiding-center orbits obtained with the quasi-geometric integration method and respective invariants of motion for axisymmetric and non-axisymmetric geometries are analyzed in detail. Moreover, the introduced artificial chaos is studied. The application of the method to the evaluation of transport coefficients is presented in chapter 6, where the computational orbit-integration performance is benchmarked as well. In chapter 7 the method is tested for the computation of fusion alpha losses in a realistic stellarator configuration. Finally, conclusions from these studies and further outlook is given in chapter 8.

Part I

Theoretical background

Chapter 1

Lagrangian and Hamiltonian mechanics

The description of Lagrangian and Hamiltonian mechanics follows mainly the books by Landau and Lifshitz (Ref. [40]) and Goldstein (Ref. [41]) as well as the review paper by Cary and Brizard (Ref. [7]). This chapter is not considered to be an independent work of the author but outlines a brief review of the theoretical background which is a prerequisite for chapter 2 and the derivations in part II.

Lagrangian description of classical mechanics:
A classical mechanical system with N degrees of freedom can be fully described via its Lagrangian $L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t)$ which is a function of (arbitrary) generalized coordinates in configuration space, $\boldsymbol{q} = (q^1, q^2, \ldots, q^N)$, their respective time derivatives ("generalized velocities"), $\dot{\boldsymbol{q}} = (\dot{q}^1, \dot{q}^2, \ldots, \dot{q}^N)$, and time t. The motion of such a mechanical system is given by a trajectory in $2N$-dimensional phase-space. The stationary action principle requires that this trajectory's action,

$$S = \int_{t_0}^{t_1} L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t)\ \mathrm{d}t, \tag{1.1}$$

must be stationary with respect to variations of the trajectory $\delta\boldsymbol{q}(t)$ (virtual displacements) in configuration space. From this variational principle the Euler-Lagrange equations of motion follow,

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}^i} = \frac{\partial L}{\partial q^i}, \tag{1.2}$$

which are a set of N second order ordinary differential equations. In the case that the Lagrangian is independent of a coordinate q^i, i.e. q^i does not appear in the Lagrangian but only its time derivative $\dot{q}^i$ appears,

$$\frac{\partial L}{\partial q^i} = 0, \tag{1.3}$$

there exists due to this symmetry in the Lagrangian (in accordance with Noether's theorem) an associated invariant of motion,

$$\frac{\partial L}{\partial \dot{q}^i} = \text{const.} \tag{1.4}$$

Furthermore, it is worth to mention that the Lagrangian is not unique. In fact, one can add a total time derivative of a function $F(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ to the Lagrangian yielding a new Lagrangian,

$$\tilde{L} = L + \frac{\mathrm{d}F(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\mathrm{d}t}, \tag{1.5}$$

without affecting the equations of motion, since the added terms in the action integral,

$$\begin{aligned} \tilde{S} &= \int_{t_0}^{t_1} L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t) + \frac{\mathrm{d}F(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\mathrm{d}t}\, \mathrm{d}t \\ &= S(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t) + F(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), t_1) - F(\boldsymbol{q}(t_0), \dot{\boldsymbol{q}}(t_0), t_0), \end{aligned} \tag{1.6}$$

are independent of $\boldsymbol{q}$ and will thus vanish upon variation.
Moreover, another useful feature of the Lagrangian formalism is its coordinate independence, i.e. the action integral, Eq. (1.1), is allowed to be computed in any coordinate system. In order to obtain the Lagrangian for new generalized coordinates and velocities $(\boldsymbol{s}, \dot{\boldsymbol{s}})$ one simply applies the chain rule for partial derivatives and substitutes $\boldsymbol{q}(\boldsymbol{s}, t)$ and $\dot{\boldsymbol{q}} = \partial \boldsymbol{q}/\partial t + \Sigma \dot{s}^i \partial \boldsymbol{q}/\partial s^i$ in the former Lagrangian.

Canonical Hamiltonian description of classical mechanics:
The Hamiltonian formalism gives a different insight into problems of classical mechanics when compared with the Lagrangian formalism due to its description with a set of canonical coordinates $(\boldsymbol{q}, \boldsymbol{p})$. In contrast to the Lagrangian formalism, $\boldsymbol{q}$ and $\boldsymbol{p}$ are placed on equal footage and, thus, canonical transformations leading to new ideas of phase space structure are allowed.[7]
First, we define the components of the canonical momentum $\boldsymbol{p}$,

$$p_i = \frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\partial \dot{q}^i}, \tag{1.7}$$

and, second, we introduce the Hamiltonian $H(\boldsymbol{q}, \boldsymbol{p}, t)$ via the Legendre transformation,

$$H(\boldsymbol{q}, \boldsymbol{p}, t) = \boldsymbol{p} \cdot \dot{\boldsymbol{q}} - L(\boldsymbol{q}, \dot{\boldsymbol{q}}(\boldsymbol{q}, \boldsymbol{p}, t), t). \tag{1.8}$$

The equivalence to the Euler-Lagrange equations (1.2) are then Hamilton's canonical

equations,

$$\dot{q}^i = \frac{\partial H}{\partial p_i} \quad \text{and} \quad \dot{p}_i = -\frac{\partial H}{\partial q^i}, \tag{1.9}$$

which are a set of $2N$ first order ordinary differential equations.
A point in phase-space, whose time evolution obeys these equations, can be conveniently denoted by the $2N$-dimensional vector $\boldsymbol{z} = (q^1, \ldots, q^N, p^1, \ldots, p^N)$. Moreover, we introduce the fundamental symplectic $N \times N$ matrix,

$$\boldsymbol{\sigma} = \begin{pmatrix} 0 & \delta^{ij} \\ -\delta^{ij} & 0 \end{pmatrix}, \tag{1.10}$$

which we utilize to define the Poisson brackets for the coordinates and canonical momenta among themselves,

$$\{z^\alpha, z^\beta\} = \sigma^{\alpha\beta} \quad \text{or} \quad \{q^i, p_j\} = \delta^i_j. \tag{1.11}$$

Here, the greek indices α, β denote phase-space components and take values from 1 to $2N$, the latin indices i, j denote configuration or momentum space components and take the values from 1 to N.
The canonical Poisson bracket for any functions f and g in phase-space is defined as

$$\{f, g\} = \frac{\partial f}{\partial z^\alpha} \sigma^{\alpha\beta} \frac{\partial g}{\partial z^\beta} = \frac{\partial f}{\partial \boldsymbol{q}} \cdot \frac{\partial g}{\partial \boldsymbol{p}} - \frac{\partial f}{\partial \boldsymbol{p}} \cdot \frac{\partial g}{\partial \boldsymbol{q}}, \tag{1.12}$$

which is a bilinear antisymmetric differential operator. The total derivative of such a function $f(\boldsymbol{q}(t), \boldsymbol{p}(t), t)$, i.e. the convective derivative of f in phase-space, can be expressed with the help of the Poisson bracket,

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + \{f, H\}. \tag{1.13}$$

By utilizing this property we can now denote Hamilton's canonical equations of motion (1.9) in the form

$$\frac{\mathrm{d}z^\alpha}{\mathrm{d}t} = \sigma^{\alpha\beta} \frac{\partial H}{\partial z^\beta}. \tag{1.14}$$

Moreover, in a similar manner it can be shown that

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\partial H}{\partial t} = \frac{\partial L}{\partial t} \tag{1.15}$$

and, hence, it follows that H is conserved if L does not explicitly depend on time.
Coordinate transformations $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p}) \to \boldsymbol{Z} = (\boldsymbol{Q}, \boldsymbol{P})$ for which the Poisson bracket remains form invariant are termed canonical transformations. Hence, canonical

transformations preserve the form of Hamilton's equations, i.e. after transforming $\boldsymbol{z} \to \boldsymbol{Z}$ there exit a new Hamiltonian $K(\boldsymbol{Q}, \boldsymbol{P})$ for which the time evolution is given by Eqs. (1.9) in the new variables. E.g., replace H,$\boldsymbol{q}$ and $\boldsymbol{p}$ with K,$\boldsymbol{Q}$ and $\boldsymbol{P}$.
The Hamilton formalism and the corresponding Hamilton's equations can be used for problems of classical mechanics under the condition that Eq. (1.7) may be inverted, i.e. velocities can be expressed as functions of the canonical momenta, $\dot{\boldsymbol{q}}(\boldsymbol{q}, \boldsymbol{p}, t)$.[7]

Phase-space Lagrangian:
Without going into details, we will now also introduce the concept of the phase-space Lagrangian $\mathcal{L}$ which is a Lagrangian that yields the correct equations of motion in phase-space when all phase-space coordinates are varied.[7] The phase-space Lagrangian is defined via

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{p}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{p}}, t) = \boldsymbol{p} \cdot \dot{\boldsymbol{q}} - H(\boldsymbol{q}, \boldsymbol{p}, t). \tag{1.16}$$

and equals the Lagrangian L in value. When applying the phase-space Euler-Lagrange equations for $\boldsymbol{q}$ and $\boldsymbol{p}$ we obtain Hamilton's canonical equations,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^i} = \frac{\partial \mathcal{L}}{\partial q^i} \quad &\to \quad \dot{p}_i = -\frac{\partial H}{\partial q^i} \qquad \text{and} \\ \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{p}_i} = \frac{\partial \mathcal{L}}{\partial p_i} \quad &\to \quad \dot{q}^i = \frac{\partial H}{\partial p_i}. \end{aligned} \tag{1.17}$$

Non-canonical Hamiltonian description of classical mechanics:
We consider now a transformation from canonical variables $(\boldsymbol{q}, \boldsymbol{p})$ to any non-canonical variables z^α where α takes the values from 1 to $2N$. In order to express a new phase-space Lagrangian in these non-canonical variables, we require the appropriate phase-space parametrization, i.e. $q^i(\boldsymbol{z}, t)$ and $p_i(\boldsymbol{z}, t)$. The corresponding total derivative of the coordinates $\boldsymbol{q}$ is then

$$\dot{q}^i = \frac{\partial q^i}{\partial t} + \dot{z}^\alpha \frac{\partial q^i}{\partial z^\alpha} \tag{1.18}$$

and after inserting in the phase-space Lagrangian, Eq. (1.16), the non-canonical formulation follows,

$$\begin{aligned} \mathcal{L}(\boldsymbol{z}, t) &= \underbrace{p_i(\boldsymbol{z}, t) \cdot \frac{\partial q^i(\boldsymbol{z}, t)}{\partial z^\alpha}}_{\Lambda_\alpha} \dot{z}^\alpha - \left(\underbrace{H(q^i(\boldsymbol{z}, t), p_i(\boldsymbol{z}, t), t) - p_i(\boldsymbol{z}, t) \cdot \frac{\partial q^i(\boldsymbol{z}, t)}{\partial t}}_{\mathcal{H}} \right) \\ &= \Lambda_\alpha(\boldsymbol{z}, t)\dot{z}^\alpha - \mathcal{H}(\boldsymbol{z}, t). \end{aligned} \tag{1.19}$$

The vector $\Lambda_\alpha(\boldsymbol{z}, t)$ and the Hamiltonian $\mathcal{H}(\boldsymbol{z}, t)$ define Hamiltonian flows, i.e. the

integral lines of the Hamiltonian vector fields in both canonical and non-canonical coordinates. The first term $\Lambda_\alpha \dot{z}^\alpha$ is the symplectic part, in which the time derivatives $\dot{z}^\alpha$ appear only at first order and the second term $-\mathcal{H}$ is the Hamiltonian part.[7]
We define now the skew symmetric Lagrange tensor,

$$\omega_{\alpha\beta} = \frac{\partial \Lambda_\beta(\boldsymbol{z},t)}{\partial z^\alpha} - \frac{\partial \Lambda_\alpha(\boldsymbol{z},t)}{\partial z^\beta}, \qquad \omega_{\alpha\beta} = -\omega_{\beta\alpha} \tag{1.20}$$

which is inspired from the Euler-Lagrange equations for $\boldsymbol{z}$,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}(\boldsymbol{z},t)}{\partial \dot{z}^\alpha} &= \frac{\partial \mathcal{L}(\boldsymbol{z},t)}{\partial z^\alpha} \\ \left(\underbrace{\frac{\partial \Lambda_\beta(\boldsymbol{z},t)}{\partial z^\alpha} - \frac{\partial \Lambda_\alpha(\boldsymbol{z},t)}{\partial z^\beta}}_{\omega_{\alpha\beta}} \right) \dot{z}^\alpha &= \frac{\partial \mathcal{H}(\boldsymbol{z},t)}{\partial z^\beta} + \frac{\partial \Lambda_\beta(\boldsymbol{z},t)}{\partial t}. \end{aligned} \tag{1.21}$$

Eventually, we invert the Lagrange tensor $\omega_{\alpha\beta}$ and obtain the Poisson matrix $\Pi^{\alpha\beta}$ ($\Pi^{\alpha\beta}\omega_{\beta\gamma} = \delta^\alpha_\gamma$) which allows us to express the non-canonical equations of motion as

$$\frac{\mathrm{d}z^\alpha}{\mathrm{d}t} = \Pi^{\alpha\beta}\left(\frac{\partial \mathcal{H}(\boldsymbol{z},t)}{\partial z^\beta} + \frac{\partial \Lambda_\beta(\boldsymbol{z},t)}{\partial t} \right). \tag{1.22}$$

Let us now transform the canonical equations of motion, Eq. (1.14), by directly transforming the canonical variables $\boldsymbol{y} = (\boldsymbol{q},\boldsymbol{p})$ to the non-canonical variables $\boldsymbol{z}$ and by expressing the canonical Hamiltonian in terms of the non-canonical variables, $H(\boldsymbol{q}(\boldsymbol{z},t),\boldsymbol{p}(\boldsymbol{z},t),t)$.

$$\begin{aligned} \frac{\mathrm{d}y^\alpha}{\mathrm{d}t} &= \sigma^{\alpha\beta}\frac{\partial H(\boldsymbol{y},t)}{\partial y^\beta} \\ &= \sigma^{\alpha\beta}\frac{\partial H(\boldsymbol{y}(\boldsymbol{z},t),t)}{\partial z^\gamma}\frac{\partial z^\gamma}{\partial y^\beta}. \end{aligned} \tag{1.23}$$

Furthermore, the direct transformation of $\dot{\boldsymbol{y}}$ in accordance with Eq. (1.18) yields

$$\frac{\mathrm{d}y^\alpha}{\mathrm{d}t} = \frac{\partial y^\alpha}{\partial t} + \frac{\partial y^\alpha}{\partial z^\beta}\dot{z}^\beta. \tag{1.24}$$

By combining these two expressions and multiplication from the left with $\partial z^\beta/\partial y^\alpha$

we obtain

$$\begin{aligned}
\dot{z}^\beta &= \frac{\partial z^\beta}{\partial y^\alpha}\left(\frac{\mathrm{d}y^\alpha}{\mathrm{d}t} - \frac{\partial y^\alpha}{\partial t}\right) \\
&= \frac{\partial z^\beta}{\partial y^\alpha}\left(\sigma^{\alpha\delta}\frac{\partial z^\gamma}{\partial y^\delta}\frac{\partial H(\boldsymbol{y}(\boldsymbol{z},t),t)}{\partial z^\gamma} - \frac{\partial y^\alpha}{\partial t}\right) \\
&= \underbrace{\frac{\partial z^\beta}{\partial y^\alpha}\sigma^{\alpha\delta}\frac{\partial z^\gamma}{\partial y^\delta}}_{\Pi^{\beta\gamma}}\left(\frac{\partial H(\boldsymbol{y}(\boldsymbol{z},t),t)}{\partial z^\gamma} - \frac{\partial y^\mu}{\partial z^\gamma}\sigma_{\mu\nu}\frac{\partial y^\nu}{\partial t}\right). \qquad (1.25)
\end{aligned}$$

In accordance with Eq. (1.12) we identify here the Poisson matrix $\Pi^{\alpha\beta}$ to be the Poisson bracket of z^α and z^β. Thus, the Poisson matrix $\Pi^{\alpha\beta}$ equals the symplectic matrix $\sigma^{\alpha\beta}$ in the case of canonical coordinates.

Chapter 2

Guiding-center dynamics

The derivation of guiding-center dynamics follows mainly the papers by Littlejohn (Ref. [5]) and Boozer (Ref. [6]) as well as the review paper by Cary and Brizard (Ref. [7]). This chapter is not considered to be an independent work of the author but outlines a brief review of the theoretical background for the derivations in part II.

2.1 Motivation for guiding-center theory

The dynamics of a charged particle with mass m and charge e in a straight and uniform magnetic field, $\mathbf{B} = B\hat{\mathbf{h}}$, are well known as a movement along a helix. In the absence of an electric field its kinetic energy remains conserved and its speed $v = |\mathbf{v}|$. Furthermore, this particle velocity can be separated into two constant components, specifically the velocity $v_\parallel = \mathbf{v} \cdot \hat{\mathbf{h}}$ parallel to the magnetic field and the perpendicular velocity $v_\perp = |\mathbf{v}_\perp| = (v^2 - v_\parallel^2)^{1/2}$. While the motion parallel to the magnetic field is uniform along a magnetic field line, the perpendicular motion ("gyromotion") is confined to a circle, whose center of gyration moves along the same magnetic field line. The cyclotron frequency $\omega_c = eB/mc$ with the speed of light c is obtained by the equality of centripetal force and magnetic Lorentz force. The radius vector of gyration ("Larmor radius") $\boldsymbol{\rho}(v_\perp, \phi) = \hat{\mathbf{h}} \times \mathbf{v}_\perp/\omega_c$ is dependent on the gyrophase ϕ.[7] Inspired by the simple motion of the gyration-center along the magnetic field line in a straight and uniform magnetic field, we long for a theory for reduced dynamical equations describing the charged particle motion in a slowly-varying electromagnetic field, i.e. the scale length L of the magnetic field is large in comparison with the Larmor radius and the distance $v_\parallel/\omega_c$ travelled by the particle during one period of gyration. Under these assumptions the electromagnetic field is approximately constant. Thus, the above described motion in a constant field should be approximately correct in a slowly-varying field. This includes that invariants of motion in the constant field, e.g. the magnetic moment $\mu = mv_\perp^2/2B$, should still be approximately invariants.[7] In 1940 H. Alfvén showed that in fact the magnetic moment is the adiabatic invariant

associated with the fast gyromotion in a slowly-varying magnetic field[42]. Furthermore, Alfvén derived that cross-field drift speeds due to the gradient of the magnetic field and the curvature of the magnetic field line are smaller than $v_\perp$ by the factor ρ/L. The goal of this chapter is to start from a first principle variational approach to derive Hamiltonian equations of motion which describe the reduced dynamics of a charged particle in a slowly-varying electromagnetic field and which are independent of the gyrophase. Specifically, we separate the relatively fast circular motion around a point called the guiding-center, and primarily treat the relatively slow drift motion of this point. Moreover, we specifically seek a Hamiltonian formulation due to two major beneficial properties: "Hamiltonian formulations possess Noether's theorem (hence invariants follow from symmetries), and they preserve the Poincaré invariants (so that spurious attractors are prevented from appearing in simulations of guiding-center dynamics)."[7]

2.2 Derivation of guiding-center Lagrangian

Phase-space Lagrangian:
We start our derivation with the Lagrangian in Cartesian coordinates corresponding to a charged particle with mass m and charge e moving in a (quasi-)static electromagnetic field,

$$L(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \frac{m}{2}|\dot{\boldsymbol{x}}|^2 + \frac{e}{c}\dot{\boldsymbol{x}} \cdot \mathbf{A}(\boldsymbol{x}) - e\Phi(\boldsymbol{x}), \tag{2.1}$$

with the speed of light c and in terms of the scalar electrostatic potential Φ and the vector potential $\mathbf{A}$. The electromagnetic field is given via $\mathbf{E} = -\nabla\Phi$ and $\mathbf{B} = \nabla \times \mathbf{A}$. By computing the canonical momentum, $p_i = \partial L/\partial \dot{x}^i$, and performing the Legendre transformation, $H(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{p} \cdot \dot{\boldsymbol{x}} - L$, we obtain the corresponding Hamiltonian,

$$H(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{2m}\left|\boldsymbol{p} - \frac{e}{c}\mathbf{A}(\boldsymbol{x})\right|^2 + e\Phi(\boldsymbol{x}), \tag{2.2}$$

with the canonical momentum

$$\boldsymbol{p} = m\dot{\boldsymbol{x}} + \frac{e}{c}\mathbf{A}. \tag{2.3}$$

The phase-space Lagrangian $\mathcal{L}$, which equals the configuration-space Lagrangian L in value, can now be denoted as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{p}, \dot{\boldsymbol{x}}, \dot{\boldsymbol{p}}) = \boldsymbol{p} \cdot \dot{\boldsymbol{x}} - \frac{1}{2m}\left|\boldsymbol{p} - \frac{e}{c}\mathbf{A}(\boldsymbol{x})\right|^2 - e\Phi(\boldsymbol{x}). \tag{2.4}$$

In the Lagrangian formalism, arbitrary coordinate transformations are allowed. Thus,

we are free to define the particle velocity as

$$\boldsymbol{v} = \frac{1}{m}\left(\boldsymbol{p} - \frac{e}{c}\mathbf{A}(\boldsymbol{x})\right) \tag{2.5}$$

and transform from the canonical coordinates $(\boldsymbol{x}, \boldsymbol{p})$ to the non-canonical coordinates $(\boldsymbol{x}, \boldsymbol{v})$ yielding

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}, \dot{\boldsymbol{x}}, \dot{\boldsymbol{v}}) = \left(m\boldsymbol{v} + \frac{e}{c}\mathbf{A}(\boldsymbol{x})\right) \cdot \dot{\boldsymbol{x}} - \left(e\Phi(\boldsymbol{x}) + \frac{m}{2}|\boldsymbol{v}|^2\right). \tag{2.6}$$

When applying the phase-space Euler-Lagrange equations for the velocity variables $\boldsymbol{v}$, we obtain the identification

$$\dot{\boldsymbol{x}} = \boldsymbol{v}. \tag{2.7}$$

An explicit evaluation of the phase-space Euler-Lagrange equations for the coordinate variables $\boldsymbol{x}$ yields

$$m\dot{\boldsymbol{v}} = e\mathbf{E}(\boldsymbol{x}) + \frac{e}{c}\dot{\boldsymbol{x}} \times \mathbf{B}(\boldsymbol{x}), \tag{2.8}$$

which is the Lorentz force after the identification Eq. (2.7) is made.[7] (The solution for a uniform magnetic field yields the particle motion along a helix which is described in section 2.1.)

Transformation to guiding-center coordinates:
Inspired by the solution in a uniform magnetic field, we transform now from the non-canonical particle coordinates $(\boldsymbol{x}, \boldsymbol{v})$ in phase-space to new guiding-center coordinates $(\boldsymbol{x}_{\rm gc}, \phi, v_\parallel, v_\perp)$ related to particle position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$ via

$$\boldsymbol{x}(\boldsymbol{x}_{\rm gc}, \phi, v_\perp) = \boldsymbol{x}_{\rm gc} + \boldsymbol{\rho}(\boldsymbol{x}_{\rm gc}, \phi, v_\perp), \qquad \text{where} \tag{2.9}$$

$$\boldsymbol{\rho}(\boldsymbol{x}_{\rm gc}, \phi, v_\perp) \equiv \frac{mcv_\perp}{eB(\boldsymbol{x}_{\rm gc})}\hat{\boldsymbol{\rho}}(\boldsymbol{x}_{\rm gc}, \phi) \equiv \rho_c(v_\perp, \boldsymbol{x}_{\rm gc})\hat{\boldsymbol{\rho}}(\boldsymbol{x}_{\rm gc}, \phi), \qquad \text{and} \tag{2.10}$$

$$\boldsymbol{v}(\boldsymbol{x}_{\rm gc}, \phi, v_\parallel, v_\perp) = v_\parallel \hat{\boldsymbol{h}}(\boldsymbol{x}_{\rm gc}) + v_\perp \hat{\boldsymbol{n}}(\boldsymbol{x}_{\rm gc}, \phi). \tag{2.11}$$

Here, the unit vector in the direction of the magnetic field, $\hat{\boldsymbol{h}}(\boldsymbol{x}_{\rm gc}) = \boldsymbol{B}(\boldsymbol{x}_{\rm gc})/B(\boldsymbol{x}_{\rm gc})$, forms a tripod with some pair of fixed but possibly space-dependent unit vectors $\hat{\boldsymbol{e}}_1(\boldsymbol{x}_{\rm gc})$ and $\hat{\boldsymbol{e}}_2(\boldsymbol{x}_{\rm gc})$ via $\hat{\boldsymbol{e}}_1(\boldsymbol{x}_{\rm gc}) \times \hat{\boldsymbol{e}}_2(\boldsymbol{x}_{\rm gc}) = \hat{\boldsymbol{h}}(\boldsymbol{x}_{\rm gc})$. The dynamic gyrating unit vectors $\hat{\boldsymbol{\rho}}$ and $\hat{\boldsymbol{n}}$ are defined as

$$\hat{\boldsymbol{\rho}}(\boldsymbol{x}_{\rm gc}, \phi) = \cos(\phi)\,\hat{\boldsymbol{e}}_1(\boldsymbol{x}_{\rm gc}) - \sin(\phi)\,\hat{\boldsymbol{e}}_2(\boldsymbol{x}_{\rm gc}) = -\frac{\partial\hat{\boldsymbol{n}}(\boldsymbol{x}_{\rm gc}, \phi)}{\partial\phi}, \tag{2.12}$$

$$\hat{\boldsymbol{n}}(\boldsymbol{x}_{\mathrm{gc}},\phi)=\hat{\boldsymbol{\rho}}(\boldsymbol{x}_{\mathrm{gc}},\phi)\times\hat{\boldsymbol{h}}(\boldsymbol{x}_{\mathrm{gc}})=-\sin(\phi)\,\hat{\boldsymbol{e}}_1(\boldsymbol{x}_{\mathrm{gc}})-\cos(\phi)\,\hat{\boldsymbol{e}}_2(\boldsymbol{x}_{\mathrm{gc}})=\frac{\partial\hat{\boldsymbol{\rho}}(\boldsymbol{x}_{\mathrm{gc}},\phi)}{\partial\phi}. \tag{2.13}$$

Furthermore, $(\hat{\boldsymbol{e}}_1,\hat{\boldsymbol{e}}_2,\hat{\boldsymbol{h}})$ and $(\hat{\boldsymbol{n}},\hat{\boldsymbol{\rho}},\hat{\boldsymbol{h}})$ form an orthonormal basis, where $\hat{\boldsymbol{e}}_1,\hat{\boldsymbol{e}}_2$ depend only on $\boldsymbol{x}_{\mathrm{gc}}$, but $\hat{\boldsymbol{n}},\hat{\boldsymbol{\rho}}$ also on ϕ.[43]

In the case of a uniform magnetic field, the variables ϕ, $v_\parallel$ and $v_\perp$ are respectively equal to the gyrophase and parallel and perpendicular velocity of the particle. In a slowly-varying field, these variables still approximate the particle values averaged over one gyro-period. In the same manner, $\boldsymbol{x}_{\mathrm{gc}}$ describes the center of the Larmor circle only in the limiting case of a uniform magnetic field, but remains close to it in a slowly-varying field. We term $\boldsymbol{x}_{\mathrm{gc}}$ the guiding-center position.[43]

We express the phase-space Lagrangian, Eq. (2.6), now in terms of the guiding-center coordinates $(\boldsymbol{x}_{\mathrm{gc}},\phi,v_\parallel,v_\perp)$

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_{\mathrm{gc}},\dot{\boldsymbol{x}}_{\mathrm{gc}},\phi,\dot{\phi},v_\parallel,v_\perp) &= \left(mv_\parallel\hat{\boldsymbol{h}}(\boldsymbol{x}_{\mathrm{gc}})+mv_\perp\hat{\boldsymbol{n}}(\boldsymbol{x}_{\mathrm{gc}},\phi)+\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}+\boldsymbol{\rho})\right)\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}+\dot{\boldsymbol{\rho}})\\ &- \left(\frac{m(v_\parallel^2+v_\perp^2)}{2}+e\Phi(\boldsymbol{x}_{\mathrm{gc}}+\boldsymbol{\rho})\right)\end{aligned} \tag{2.14}$$

with the Larmor radius $\boldsymbol{\rho}=\boldsymbol{\rho}(\boldsymbol{x}_{\mathrm{gc}},\phi,v_\perp)=\boldsymbol{x}-\boldsymbol{x}_{\mathrm{gc}}$ from Eq. (2.9). It is worth mentioning that the transformation of Eqs. (2.9-2.11) and the phase-space Lagrangian in the guiding-center coordinates, Eq. (2.14), are still exact.

Transformation to guiding-center phase-space Lagrangian:

As stated in the motivation of this chapter, we seek a transformation of the Lagrangian, Eq. (2.14), to a new guiding-center phase-space Lagrangian $\mathcal{L}_{\mathrm{gc}}$ in which the degree of freedom corresponding to the gyromotion, i.e. the gyrophase ϕ, is absent from the equations of motion. In accordance with Noether's theorem that means, if the gyrophase ϕ is an ignorable variable in the Lagrangian, it does not appear in the equations of motion and, furthermore, its conjugated variable in the Lagrangian is an invariant of motion.[7]

We will now make the actual guiding-center approximation that relies on a "small" Larmor radius and a slowly-varying field. Specifically, the scale length L of the electromagnetic field must be large in comparison with the Larmor radius and the distance $v_\parallel/\omega_c$ travelled by the particle during one period of gyration. This transformation is accomplished by utilizing a Taylor series expansion of the field quantities that is truncated at the first order and in addition we introduce an ordering parameter ε that will help us to collect terms of similar order of magnitude. Eventually, we want to omit small terms up to the order ε in the Lagrangian with the result of describing

purely the guiding-center motion (including appropriate drifts) and neglecting the fast gyromotion.

There exist different approaches how to introduce the ordering parameter ε. On the one hand special units can be introduced such that the transformation is performed with $\varepsilon = e/m$ and units are restored at the end of the transformation; see Ref. [7]. On the other hand terms of the Lagrangian can simply be marked by ε in the intermediate calculations and in the final result ε is set to one. We pursue here the latter approach. In order to satisfy the small Larmor radius and slowly-varying field condition, we choose

$$\rho_c = |\boldsymbol{\rho}| = \frac{mcv_\perp}{eB(\boldsymbol{x}_{\mathrm{gc}})} \sim \varepsilon \tag{2.15}$$

and we define the cyclotron frequency at the guiding-center position

$$\omega_c = \frac{eB(\boldsymbol{x}_{\mathrm{gc}})}{mc} \sim \varepsilon^{-1}, \tag{2.16}$$

as a large parameter compared to typical rates. This choice corresponds to the physical conditions in which the guiding-center approximation is applicable. Furthermore, the perpendicular velocity $v_\perp$, which is a a quantity of interest, has naturally the order $\varepsilon^0 = 1$ and is thus prevented from being swallowed by our perturbation expansion.[43] In Ref. [7] an ordering is chosen, in which the electric drift is in the same order as the perpendicular velocity. In contrast, we assume here the electric field to be small ($\Phi \sim \varepsilon^0$); see Ref. [5]. The magnetic field should dominate the remaining terms and is thus in the order of ε^{-1}.

The vector potential and the scalar electrostatic potential at the particle position are expanded via a Taylor series,

$$\varepsilon^{-1}\boldsymbol{A}(\boldsymbol{x}) = \varepsilon^{-1}\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}) + \varepsilon\boldsymbol{\rho}\cdot\varepsilon^{-1}\nabla\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}) + \mathcal{O}(\varepsilon^2), \tag{2.17}$$

$$\Phi(\boldsymbol{x}) = \Phi(\boldsymbol{x}_{\mathrm{gc}}) + \varepsilon\boldsymbol{\rho}\cdot\nabla\Phi(\boldsymbol{x}_{\mathrm{gc}}) + \mathcal{O}(\varepsilon^2), \tag{2.18}$$

where the ordering parameter ε marks small terms.

The ordering specification of Eqs. (2.15) and (2.16) is however not sufficient for completing our derivation, since also the relative sizes of the time derivatives of variables need to be known. This becomes apparent if we look at the total time derivative of the Larmor radius vector

$$\begin{aligned}\dot{\boldsymbol{\rho}} &= \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho}(\boldsymbol{x}_{\mathrm{gc}}, \phi, v_\perp) = \dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{\rho} + \dot{\phi}\frac{\partial\boldsymbol{\rho}}{\partial\phi} + \dot{v}_\perp\frac{\partial\boldsymbol{\rho}}{\partial v_\perp} \\ &= \varepsilon\dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{\rho} + \varepsilon^{-1}\dot{\phi}\varepsilon\rho_c\hat{\boldsymbol{n}} + \varepsilon\frac{\dot{v}_\perp}{v_\perp}\boldsymbol{\rho} = \dot{\phi}\rho_c\hat{\boldsymbol{n}} + \varepsilon(\dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{\rho} + \frac{\dot{v}_\perp}{v_\perp}\boldsymbol{\rho}),\end{aligned} \tag{2.19}$$

where in the first term the fast gyration $\dot{\phi}$ and the small ρ_c cancel to order ε^0.
The Taylor series expansion of the fields and the ordering of the individual terms with respect to their orders of magnitude allows us now to write the Lagrangian as

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_{\mathrm{gc}}, \dot{\boldsymbol{x}}_{\mathrm{gc}}, \phi, \dot{\phi}, v_{\parallel}, v_{\perp}) &= \left(mv_{\parallel}\hat{\boldsymbol{h}} + mv_{\perp}\hat{\boldsymbol{n}} + \varepsilon^{-1}\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}) + \frac{e}{c}\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})\right)\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}} \\ &+ mv_{\perp}\dot{\phi}\rho_c + \left(\varepsilon^{-1}\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}) + \frac{e}{c}\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})\right)\cdot\dot{\boldsymbol{\rho}} \\ &- \left(\frac{m(v_{\parallel}^2 + v_{\perp}^2)}{2} + e\Phi(\boldsymbol{x}_{\mathrm{gc}})\right) + \mathcal{O}(\varepsilon), \end{aligned} \tag{2.20}$$

by keeping only terms until $\mathcal{O}(1)$. Here, the term with $\hat{\boldsymbol{h}}\cdot\hat{\boldsymbol{n}}$ vanished and we performed a scalar product $\hat{\boldsymbol{n}}\cdot\hat{\boldsymbol{n}} = 1$ in order to obtain the first term in the second line. All occurring components of the electromagnetic field $(\boldsymbol{A}, \nabla\boldsymbol{A}, \Phi)$ are now evaluated at the guiding-center position.[43]
However, in Eq. (2.20) the vectors $\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}$ and $\hat{\boldsymbol{n}}$ are still functions of the gyrophase ϕ, despite omitting all terms up to the order ε. Thus, the appropriate terms with dependencies on ϕ should be pushed up to higher orders so that the Lagrangian of the order ε^0 does not contain gyrophase-dependent terms.
For this purpose we perform gauge transformations of some terms of the Lagrangian. Specifically, we will use consecutively twice the fact that one can add or subtract an arbitrary total time derivative to the Lagrangian without changing its dynamics.
First of all, we want to cancel the term $\varepsilon^{-1}ec^{-1}\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})\cdot\dot{\boldsymbol{\rho}}$. By taking

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\rho}\cdot\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})) &= \dot{\boldsymbol{\rho}}\cdot\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}) + \boldsymbol{\rho}\cdot\dot{\boldsymbol{A}}(\boldsymbol{x}_{\mathrm{gc}}) \\ &= \dot{\boldsymbol{\rho}}\cdot\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}) + \boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})), \end{aligned} \tag{2.21}$$

we can combine the second term with the term $(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A})\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}} = \boldsymbol{\rho}\cdot((\nabla\boldsymbol{A})\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}})$ by utilizing Graßmann's identity for the cross product of a curl,

$$(\nabla\boldsymbol{A})\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}} - \dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{A} = \dot{\boldsymbol{x}}_{\mathrm{gc}}\times(\nabla\times\boldsymbol{A}) = \dot{\boldsymbol{x}}_{\mathrm{gc}}\times\boldsymbol{B}. \tag{2.22}$$

The combination then yields

$$(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A})\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}} - \boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{A}) = \boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\times\boldsymbol{B}), \tag{2.23}$$

which allows us to express the first term together with Eq. (2.21) and we finally obtain

$$(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A})\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}} = \boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\times\boldsymbol{B} + \dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\nabla\boldsymbol{A}) = \boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\times\boldsymbol{B}) + \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\rho}\cdot\boldsymbol{A}) - \dot{\boldsymbol{\rho}}\cdot\boldsymbol{A}. \tag{2.24}$$

By inserting Eq. (2.24) into the Lagrangian, Eq. (2.20), the term including $\boldsymbol{A}\cdot\dot{\boldsymbol{\rho}}$ is successfully cancelled and we obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_{\mathrm{gc}},\dot{\boldsymbol{x}}_{\mathrm{gc}},\phi,\dot{\phi},v_{\parallel},v_{\perp}) &= \left(mv_{\parallel}\hat{\boldsymbol{h}}+mv_{\perp}\hat{\boldsymbol{n}}+\varepsilon^{-1}\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})\right)\cdot\dot{\boldsymbol{x}}_{\mathrm{gc}}\\ &+ \frac{e}{c}\boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\times\boldsymbol{B})+mv_{\perp}\dot{\phi}\rho_c+\frac{e}{c}(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}}))\cdot\dot{\boldsymbol{\rho}}\\ &- \left(\frac{m(v_{\parallel}^2+v_{\perp}^2)}{2}+e\Phi(\boldsymbol{x}_{\mathrm{gc}})\right)\\ &+ \frac{e}{c}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\rho}\cdot\boldsymbol{A}(\boldsymbol{R}))+\mathcal{O}(\varepsilon),\end{aligned}\tag{2.25}$$

where the term including the total time derivative can be omitted from the Lagrangian without affecting the equations of motion. Furthermore, two terms of this Lagrangian are cancelling, which becomes apparent by rotating the triple-product and extracting the basis vectors,

$$\frac{e}{c}\boldsymbol{\rho}\cdot(\dot{\boldsymbol{x}}_{\mathrm{gc}}\times\boldsymbol{B})=\frac{e}{c}\rho_c B\dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot(\hat{\boldsymbol{h}}\times\hat{\boldsymbol{\rho}})=-mv_{\perp}\dot{\boldsymbol{x}}_{\mathrm{gc}}\cdot\hat{\boldsymbol{n}}.\tag{2.26}$$

Thus, the remaining term in the Lagrangian that needs to be eliminated in order to obtain an gyrophase-independent Lagrangian is the term $ec^{-1}(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A})\cdot\dot{\boldsymbol{\rho}}$. For this purpose we use the total time derivative

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\varepsilon\boldsymbol{\rho}\cdot\varepsilon^{-1}\nabla\boldsymbol{A}\cdot\varepsilon\boldsymbol{\rho}\right)&=\frac{1}{2}\left(\dot{\boldsymbol{\rho}}\cdot\nabla\boldsymbol{A}\cdot\boldsymbol{\rho}+\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}\cdot\dot{\boldsymbol{\rho}}\right)\\ &\quad+\frac{\varepsilon}{2}\boldsymbol{\rho}\cdot\left(\frac{\mathrm{d}\nabla\boldsymbol{A}}{\mathrm{d}t}\right)\cdot\boldsymbol{\rho}.\end{aligned}\tag{2.27}$$

Furthermore, we utilize the expression for the term

$$\begin{aligned}(\boldsymbol{\rho}\times\dot{\boldsymbol{\rho}})\cdot\boldsymbol{B}&=(\boldsymbol{\rho}\times\dot{\boldsymbol{\rho}})\cdot(\nabla\times\boldsymbol{A})\\ &=(\varepsilon_{ijk}\rho_j\dot{\rho}_k)(\varepsilon_{imn}\partial_m A_n)\\ &=(\delta_{jm}\delta_{kn}-\delta_{jn}\delta_{km})(\rho_j\dot{\rho}_k\partial_m A_n)\\ &=\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}\cdot\dot{\boldsymbol{\rho}}-\dot{\boldsymbol{\rho}}\cdot\nabla\boldsymbol{A}\cdot\boldsymbol{\rho}\end{aligned}\tag{2.28}$$

in order to express

$$\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}\cdot\dot{\boldsymbol{\rho}}=\frac{1}{2}\underbrace{\left(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}\cdot\dot{\boldsymbol{\rho}}-\dot{\boldsymbol{\rho}}\cdot\nabla\boldsymbol{A}\cdot\boldsymbol{\rho}\right)}_{(\boldsymbol{\rho}\times\dot{\boldsymbol{\rho}})\cdot\boldsymbol{B}}+\frac{1}{2}\underbrace{\left(\dot{\boldsymbol{\rho}}\cdot\nabla\boldsymbol{A}\cdot\boldsymbol{\rho}+\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}\cdot\dot{\boldsymbol{\rho}}\right)}_{\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\rho}\cdot\nabla\boldsymbol{A}\cdot\varepsilon\boldsymbol{\rho})-\mathcal{O}(\varepsilon)}.\tag{2.29}$$

Finally, we extract the basis vectors of the first term,

$$(\boldsymbol{\rho} \times \dot{\boldsymbol{\rho}}) \cdot \boldsymbol{B} = \left(\rho_c \hat{\boldsymbol{\rho}} \times \dot{\phi} \rho_c \hat{\boldsymbol{n}}\right) \cdot \boldsymbol{B} = -\dot{\phi} \rho_c^2 B = -\frac{m c v_\perp}{e} \rho_c \dot{\phi} \tag{2.30}$$

and we can replace $(\boldsymbol{\rho} \cdot \nabla \boldsymbol{A}) \cdot \dot{\boldsymbol{\rho}}$ in the Lagrangian, Eq. (2.25), in order to obtain

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}_{\mathrm{gc}}, \dot{\boldsymbol{x}}_{\mathrm{gc}}, \phi, \dot{\phi}, v_\parallel, v_\perp) &= \left(m v_\parallel \hat{\boldsymbol{h}} + \varepsilon^{-1} \frac{e}{c} \boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})\right) \cdot \dot{\boldsymbol{x}}_{\mathrm{gc}} \\ &+ m v_\perp \dot{\phi} \rho_c - \frac{m v_\perp}{2} \dot{\phi} \rho_c \\ &- \left(\frac{m(v_\parallel^2 + v_\perp^2)}{2} + e\Phi(\boldsymbol{x}_{\mathrm{gc}})\right) \\ &+ \frac{e}{2c} \frac{\mathrm{d}}{\mathrm{d}t} (\boldsymbol{\rho} \cdot \nabla \boldsymbol{A} \cdot \varepsilon \boldsymbol{\rho}) - \mathcal{O}(\varepsilon), \end{aligned} \tag{2.31}$$

where the term including the total time derivative can again be omitted. By further omitting all terms (including) up to $\mathcal{O}(\varepsilon)$ and eventually setting $\varepsilon = 1$, we obtain the sought guiding-center phase-space Lagrangian,

$$\begin{aligned} \mathcal{L}_{\mathrm{gc}}(\boldsymbol{x}_{\mathrm{gc}}, \dot{\boldsymbol{x}}_{\mathrm{gc}}, \dot{\phi}, v_\parallel, v_\perp) &= \left(m v_\parallel \hat{\boldsymbol{h}} + \frac{e}{c} \boldsymbol{A}(\boldsymbol{x}_{\mathrm{gc}})\right) \cdot \dot{\boldsymbol{x}}_{\mathrm{gc}} + \frac{m v_\perp}{2} \rho_c \dot{\phi} \\ &- \left(\frac{m(v_\parallel^2 + v_\perp^2)}{2} + e\Phi(\boldsymbol{x}_{\mathrm{gc}})\right), \end{aligned} \tag{2.32}$$

in which all fields are evaluated at the guiding-center position.

Associated (perpendicular) invariant of motion:
Due to the fact that the derived guiding-center phase-space Lagrangian, Eq. (2.32), is gyrophase-independent, we can find in accordance with Noether's theorem an invariant of motion which is the conjugate to $\dot{\phi}$. In particular, we apply the Euler-Lagrange equations for the gyrophase,

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{L}_{\mathrm{gc}}}{\partial \dot{\phi}}\right) = \frac{\partial \mathcal{L}_{\mathrm{gc}}}{\partial \phi} = 0, \tag{2.33}$$

to obtain the conservation and in order to identify the perpendicular adiabatic invariant ("gyroaction")

$$J_\perp \equiv \frac{\partial \mathcal{L}_{\mathrm{gc}}}{\partial \dot{\phi}} = \frac{m v_\perp}{2} \rho_c = \frac{m^2 c v_\perp^2}{2 e B(\boldsymbol{x}_{\mathrm{gc}})}. \tag{2.34}$$

2.3 Derivation of guiding-center equations of motion

In order to reduce notational clutter, in the following the sub-index "gc" is dropped and the guiding-center Lagrangian $\mathcal{L}_{\mathrm{gc}}$ is simply denoted by $\mathcal{L}$, as well as the guiding-center particle position $\boldsymbol{x}_{\mathrm{gc}}$ is denoted by $\boldsymbol{x}$. The guiding-center equations of motion are separately derived for both the parallel velocity $v_\parallel$ and the Hamiltonian H as independent phase-space coordinates. We will further show that the obtained equations of motion are equivalent for both choices. In chapter 3 we will make use of this fact when we will derive locally linear guiding-center equations of motion.

2.3.1 $v_\parallel$ as independent phase-space coordinate

Starting with the guiding-center Lagrangian with $v_\parallel$ as independent phase-space coordinate

$$\mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}}, v_\parallel, J_\perp, \dot{\phi}) = \left(\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}) + m v_\parallel \hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \dot{\boldsymbol{x}} - J_\perp \dot{\phi} - H(\boldsymbol{x}, v_\parallel, J_\perp) \quad \text{with}$$
$$H(\boldsymbol{x}, v_\parallel, J_\perp) = \frac{m v_\parallel^2}{2} + J_\perp \omega_c(\boldsymbol{x}) + e\Phi(\boldsymbol{x}), \tag{2.35}$$

we compute its partial derivatives with respect to the generalized phase-space coordinates

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = \left(\frac{e}{c}\nabla \boldsymbol{A}(\boldsymbol{x}) + m v_\parallel \nabla \hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \dot{\boldsymbol{x}} - (J_\perp \nabla \omega_c(\boldsymbol{x}) + e\nabla\Phi(\boldsymbol{x})) \tag{2.36}$$
$$\frac{\partial \mathcal{L}}{\partial \phi} = 0 \tag{2.37}$$
$$\frac{\partial \mathcal{L}}{\partial v_\parallel} = m\hat{\boldsymbol{h}}(\boldsymbol{x}) \cdot \dot{\boldsymbol{x}} - m v_\parallel \tag{2.38}$$
$$\frac{\partial \mathcal{L}}{\partial J_\perp} = -\dot{\phi} - \omega_c, \tag{2.39}$$

and its partial derivatives with respect to the generalized phase-space velocities

$$\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{x}}} = \frac{e}{c}\boldsymbol{A}(\boldsymbol{x}) + m v_\parallel \hat{\boldsymbol{h}}(\boldsymbol{x}) \tag{2.40}$$
$$\frac{\partial \mathcal{L}}{\partial \dot{\phi}} = -J_\perp. \tag{2.41}$$

Recalling the Euler-Lagrange equations (1.2),

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^i} = \frac{\partial \mathcal{L}}{\partial q^i}, \tag{2.42}$$

we see that on their left-hand sides (LHS) the total derivative with respect to time is needed, which can be expressed as a convective derivative along the phase-space trajectory

$$\frac{\mathrm{d}}{\mathrm{d}t} = \dot{\boldsymbol{x}} \cdot \nabla + \dot{v}_\parallel \frac{\partial}{\partial v_\parallel} + \dot{J}_\perp \frac{\partial}{\partial J_\perp} + \dot{\phi}\frac{\partial}{\partial \phi} + \frac{\partial}{\partial t}. \tag{2.43}$$

As already shown in section 2.2, by looking at the guiding-center Lagrangian, Eq. (2.35), we notice that ϕ does not occur explicitly and, therefore, a corresponding integral of motion must exist which can be found by $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{\phi} = \partial\mathcal{L}/\partial\phi$ yielding

$$\frac{\mathrm{d}}{\mathrm{d}t} J_\perp = 0. \tag{2.44}$$

The conservation of $J_\perp$ is equivalent to the conservation of the magnetic moment μ due to its proportionality with a constant factor by $J_\perp = \mu\frac{mc}{e}$.
Second, we perform the Euler-Lagrange equations in $v_\parallel$ and $J_\perp$, namely $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{v}_\parallel = \partial\mathcal{L}/\partial v_\parallel$ and $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{J}_\perp = \partial\mathcal{L}/\partial J_\perp$. Both phase-space coordinates do not have generalized velocity terms in the guiding-center Lagrangian, Eq. (2.35), and therefore, their left-hand sides are identically zero. Hence, from Eq. (2.38) and Eq. (2.39) we obtain, respectively,

$$v_\parallel = \hat{\boldsymbol{h}}(\boldsymbol{x}) \cdot \dot{\boldsymbol{x}} \tag{2.45}$$

and

$$\dot{\phi} = -\omega_c(\boldsymbol{x}). \tag{2.46}$$

Finally, the Euler-Lagrange equations in $\boldsymbol{x}$ using the convective derivative of Eq.(2.43), in which some terms vanish due to above observations, are

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{x}}} - \frac{\partial\mathcal{L}}{\partial\boldsymbol{x}} &= \dot{\boldsymbol{x}} \cdot \Big(\frac{e}{c}(\nabla\boldsymbol{A}(\boldsymbol{x})) + mv_\parallel(\nabla\hat{\boldsymbol{h}}(\boldsymbol{x}))\Big) + m\dot{v}_\parallel\hat{\boldsymbol{h}}(\boldsymbol{x}) \\ &\quad - \Big(\frac{e}{c}(\nabla\boldsymbol{A}(\boldsymbol{x})) + mv_\parallel(\nabla\hat{\boldsymbol{h}}(\boldsymbol{x}))\Big) \cdot \dot{\boldsymbol{x}} + J_\perp\nabla\omega_c(\boldsymbol{x}) + e\nabla\Phi(\boldsymbol{x}) = 0.\end{aligned} \tag{2.47}$$

Here, it can be mentioned that the scalar product in the convective derivative is not commutative, e.g. $(\dot{\boldsymbol{x}} \cdot \nabla)\boldsymbol{A} = \dot{\boldsymbol{x}} \cdot (\nabla\boldsymbol{A}) \neq (\nabla\boldsymbol{A}) \cdot \dot{\boldsymbol{x}}$.
Before we proceed, let us define

$$\boldsymbol{A}^\star(\boldsymbol{x}, v_\parallel) = \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e}v_\parallel\hat{\boldsymbol{h}}(\boldsymbol{x}), \tag{2.48}$$

$$\boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel) = \nabla \times \boldsymbol{A}^\star(\boldsymbol{x}, v_\parallel), \tag{2.49}$$

$$\boldsymbol{E}^\star(\boldsymbol{x}) = -\frac{1}{e}\nabla H = -\frac{1}{mc}J_\perp\nabla B(\boldsymbol{x}) - \nabla\Phi(\boldsymbol{x}). \tag{2.50}$$

Further, let us use Graßmann's identity for the cross product of a curl, $\boldsymbol{a} \times (\nabla \times \boldsymbol{h}) = (\nabla \boldsymbol{h}) \cdot \boldsymbol{a} - \boldsymbol{a} \cdot \nabla \boldsymbol{h}$, in order to express

$$\dot{\boldsymbol{x}} \times \left(\nabla \times \boldsymbol{A}^\star(\boldsymbol{x}, v_\parallel)\right) = \left(\nabla \boldsymbol{A}^\star(\boldsymbol{x}, v_\parallel)\right) \cdot \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}} \cdot \left(\nabla \boldsymbol{A}^\star(\boldsymbol{x}, v_\parallel)\right). \tag{2.51}$$

The Euler-Lagrange equations in $\boldsymbol{x}$, Eq. (2.47) can now be written with the definitions (2.48) - (2.50) and the identity (2.51) as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{x}}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = 0 &= -\frac{e}{c}\dot{\boldsymbol{x}} \times \left(\nabla \times \boldsymbol{A}^\star(\boldsymbol{x}, v_\parallel)\right) + m\dot{v}_\parallel \hat{\boldsymbol{h}}(\boldsymbol{x}) - e\boldsymbol{E}^\star(\boldsymbol{x}) \\ &= -\frac{e}{c}\dot{\boldsymbol{x}} \times \boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel) + m\dot{v}_\parallel \hat{\boldsymbol{h}}(\boldsymbol{x}) - e\boldsymbol{E}^\star(\boldsymbol{x}). \end{aligned} \tag{2.52}$$

As a next step we would like to explicitly express $\dot{\boldsymbol{x}}$ which we achieve by taking a cross product of $\hat{\boldsymbol{h}}(\boldsymbol{x})$ with Eq. (2.52)

$$-\frac{e}{c}\hat{\boldsymbol{h}}(\boldsymbol{x}) \times \left(\dot{\boldsymbol{x}} \times \boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel)\right) - e\hat{\boldsymbol{h}}(\boldsymbol{x}) \times \boldsymbol{E}^\star(\boldsymbol{x}) = 0, \tag{2.53}$$

where the second term vanished. By applying again Graßmann's identity, $\hat{\boldsymbol{h}} \times (\dot{\boldsymbol{x}} \times \boldsymbol{B}^\star) = \dot{\boldsymbol{x}}\left(\hat{\boldsymbol{h}} \cdot \boldsymbol{B}^\star\right) - \boldsymbol{B}^\star\left(\dot{\boldsymbol{x}} \cdot \hat{\boldsymbol{h}}\right)$, and using $\hat{\boldsymbol{h}}(\boldsymbol{x}) \cdot \dot{\boldsymbol{x}} = v_\parallel$ from (2.45), Eq. (2.53) yields

$$\begin{aligned} -\frac{e}{c}\left(\dot{\boldsymbol{x}}\left(\hat{\boldsymbol{h}}(\boldsymbol{x}) \cdot \boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel)\right) - \boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel)v_\parallel\right) &= e\hat{\boldsymbol{h}}(\boldsymbol{x}) \times \boldsymbol{E}^\star(\boldsymbol{x}) \\ \dot{\boldsymbol{x}} &= \frac{\boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel)v_\parallel - c\hat{\boldsymbol{h}}(\boldsymbol{x}) \times \boldsymbol{E}^\star(\boldsymbol{x})}{\hat{\boldsymbol{h}}(\boldsymbol{x}) \cdot \boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel)}. \end{aligned} \tag{2.54}$$

In a similar manner an explicit expression for $\dot{v}_\parallel$ is obtained by the scalar product of $\boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel)$ with Eq. (2.52)

$$\begin{aligned} m\dot{v}_\parallel \boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x}) - e\boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel) \cdot \boldsymbol{E}^\star(\boldsymbol{x}) &= 0 \\ \dot{v}_\parallel &= \frac{e}{m}\frac{\boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel) \cdot \boldsymbol{E}^\star(\boldsymbol{x})}{\boldsymbol{B}^\star(\boldsymbol{x}, v_\parallel) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})}. \end{aligned} \tag{2.55}$$

The obtained guiding-center equations of motion, (2.55) and (2.54), can be used to derive the guiding-center energy equation for time-independent fields,

$$\frac{\mathrm{d}}{\mathrm{d}t}H(\boldsymbol{x}, v_\parallel, J_\perp) = \dot{\boldsymbol{x}} \cdot \nabla H(\boldsymbol{x}, v_\parallel, J_\perp) + \dot{v}_\parallel \frac{\partial H(\boldsymbol{x}, v_\parallel, J_\perp)}{\partial v_\parallel} = 0, \tag{2.56}$$

by utilizing the convective total time derivative of Eq. (2.43).

2.3.2 H as independent phase-space coordinate

In this subsection it will be shown that the actual guiding-center orbits do not depend on the choice of their representing variables in phase-space, specifically the parallel velocity $v_\parallel$ does not need to be treated as independent phase-space coordinate necessarily, but can also be treated as a function of position, if the Hamiltonian H is an independent coordinate. Choosing $v_\parallel = v_\parallel(\mathbf{x}, H, J_\perp)$ or $H = H\left(\mathbf{x}, v_\parallel, J_\perp\right)$ is equivalent as long as

$$H = \frac{m v_\parallel^2}{2} + J_\perp \omega_c(\boldsymbol{x}) + e\Phi(\boldsymbol{x}) \tag{2.57}$$

is kept exact. Hence, the guiding-center Lagrangian with H as an independent phase-space coordinate is

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}}, J_\perp, \dot{\phi}, H) &= \left(\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}) + m v_\parallel(H, J_\perp, \boldsymbol{x})\hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \dot{\boldsymbol{x}} - J_\perp \dot{\phi} - H \quad \text{with} \\ v_\parallel(H, J_\perp, \boldsymbol{x}) &= \sigma\sqrt{\frac{2}{m}\left(H - J_\perp \omega_c(\boldsymbol{x}) - e\Phi(\boldsymbol{x})\right)},\end{aligned} \tag{2.58}$$

where σ is ± 1, namely the direction of the particle velocity parallel $(+1)$ or anti-parallel (-1) to the magnetic field.
The derivation of the equations of motion is very similar to that in section 2.3.1 and thus, only the main differences are highlighted here.
Let us start with the convective time derivative

$$\frac{\mathrm{d}}{\mathrm{d}t} = \dot{\boldsymbol{x}} \cdot \nabla + \dot{H}\frac{\partial}{\partial H} + \dot{J}_\perp \frac{\partial}{\partial J_\perp} + \dot{\phi}\frac{\partial}{\partial \phi} + \frac{\partial}{\partial t}. \tag{2.59}$$

By utilizing the time t as an extended phase-space variable, the last term of Eq. (2.58) can be written as $H = H\dot{t}$ because the conjugated velocity of H in extended phase-space is $\dot{t} = 1$. Thus, the Euler-Lagrange equation in t, $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{t} = \partial\mathcal{L}/\partial t$, yields

$$\frac{\mathrm{d}}{\mathrm{d}t} H = 0, \tag{2.60}$$

and the respective term in Eq. (2.59) including $\dot{H}$ vanishes. Furthermore, we define similar to Eqs. (2.48) and (2.49)

$$\widetilde{\boldsymbol{A}}^\star(\boldsymbol{x}, H, J_\perp) = \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_\parallel(H, J_\perp, \boldsymbol{x})\hat{\boldsymbol{h}}(\boldsymbol{x}), \tag{2.61}$$

$$\widetilde{\boldsymbol{B}}^\star(\boldsymbol{x}, H, J_\perp) = \nabla \times \widetilde{\boldsymbol{A}}^\star(\boldsymbol{x}, H, J_\perp), \tag{2.62}$$

where it must be mentioned that $\widetilde{\boldsymbol{A}}^\star = \boldsymbol{A}^\star$, if Eq. (2.57) is kept exact. However, $\widetilde{\boldsymbol{B}}^\star \neq \boldsymbol{B}^\star$ since $v_\parallel$ is a function of $\boldsymbol{x}$ in (2.61) and thus, an additional term occurs

in (2.62).

The cyclic variable ϕ in Eq. (2.58) yields the conservation of $J_\perp$. The Euler-Lagrange equations in H and $J_\perp$, namely $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{H} = \partial\mathcal{L}/\partial H$ and $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{J}_\perp = \partial\mathcal{L}/\partial J_\perp$, result in $v_\parallel(H, J_\perp, \boldsymbol{x}) = \hat{\boldsymbol{h}}(\boldsymbol{x}) \cdot \dot{\boldsymbol{x}}$ and $\dot{\phi} = -\omega_c(\boldsymbol{x})$. Here, attention must be paid to the parallel velocity's dependence on the phase-space coordinates when computing the respective implicit derivatives.

Finally, performing the Euler-Lagrange equations in $\boldsymbol{x}$, namely $\mathrm{d}/\mathrm{d}t\ \partial\mathcal{L}/\partial\dot{\boldsymbol{x}} = \partial\mathcal{L}/\partial\boldsymbol{x}$, and subsequently using Graßmann's identity as above yields

$$\dot{\boldsymbol{x}} \times \left(\nabla \times \widetilde{\boldsymbol{A}}^\star(\boldsymbol{x}, v_\parallel)\right) = 0. \tag{2.63}$$

Making the cross product of $\hat{\boldsymbol{h}}(\boldsymbol{x})$ with this equation allows for eventually rearranging to

$$\dot{\boldsymbol{x}} = \frac{\widetilde{\boldsymbol{B}}^\star(\boldsymbol{x}, H, J_\perp) v_\parallel(H, J_\perp, \boldsymbol{x})}{\widetilde{\boldsymbol{B}}^\star(\boldsymbol{x}, H, J_\perp) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})}, \tag{2.64}$$

which is the most compact form of the guiding-center equations of motion. With the choice of H as an independent coordinate, the parallel velocity $v_\parallel$ is a function of coordinates given in second of (2.58), nevertheless, a differential equation for an explicit expression of $\dot{v}_\parallel$ can be obtained by

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} v_\parallel(H, J_\perp, \boldsymbol{x}) &= \frac{\mathrm{d}}{\mathrm{d}t} \sigma \sqrt{\frac{2}{m}\left(H - J_\perp \omega_c(\boldsymbol{x}) - e\Phi(\boldsymbol{x})\right)} \\ \dot{v}_\parallel &= \frac{1}{m} \frac{1}{v_\parallel(H, J_\perp, \boldsymbol{x})} \dot{\boldsymbol{x}} \cdot \left(-J_\perp \nabla\omega_c(\boldsymbol{x}) - e\nabla\Phi(\boldsymbol{x})\right). \end{aligned} \tag{2.65}$$

Equivalence of guiding-center equations of motion:

We will show now that Eqs. (2.64) and (2.65) are respectively equivalent to Eqs. (2.54) and (2.55).

First, let us have a look at the equation for the guiding-center position, Eq. (2.64)

expressed with $\widetilde{\boldsymbol{A}}^{\star}(\boldsymbol{x}, v_{\parallel})$,

$$
\begin{aligned}
\dot{\boldsymbol{x}} =& \frac{\left(\nabla \times \widetilde{\boldsymbol{A}}^{\star}(\boldsymbol{x}, v_{\parallel})\right) v_{\parallel}(H, J_{\perp}, \boldsymbol{x})}{\left(\nabla \times \widetilde{\boldsymbol{A}}^{\star}(\boldsymbol{x}, v_{\parallel})\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})} \\
=& \frac{\left(\nabla \times \left(\boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \hat{\boldsymbol{h}}(\boldsymbol{x})\right)\right) v_{\parallel}(H, J_{\perp}, \boldsymbol{x})}{\left(\nabla \times \left(\boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \hat{\boldsymbol{h}}(\boldsymbol{x})\right)\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})} \\
=& \frac{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} \left(v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x}) + \nabla v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right)\right) v_{\parallel}(H, J_{\perp}, \boldsymbol{x})}{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} \left(v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x}) + \nabla v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right)\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})} \\
=& \frac{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right) v_{\parallel}(H, J_{\perp}, \boldsymbol{x})}{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})} + \\
& \frac{\left(\frac{c}{e} \frac{1}{v_{\parallel}(H, J_{\perp}, \boldsymbol{x})} \left(-J_{\perp} \nabla \omega_c(\boldsymbol{x}) - e \nabla \Phi(\boldsymbol{x})\right) \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right) v_{\parallel}(H, J_{\perp}, \boldsymbol{x})}{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel}(H, J_{\perp}, \boldsymbol{x}) \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})}.
\end{aligned} \tag{2.66}
$$

Here, $\nabla \times (f\boldsymbol{F}) = f\,(\nabla \times \boldsymbol{F}) + (\nabla f) \times \boldsymbol{F}$ was used in the numerator and $(\boldsymbol{a} \times \boldsymbol{h}) \cdot \boldsymbol{h} = 0$ in the denominator.
Now, we choose $v_{\parallel}$ as an independent coordinate implying that $H = H\left(\mathbf{x}, v_{\parallel}, J_{\perp}\right)$ and thus,

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \frac{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel} \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right) v_{\parallel} + \frac{c}{e} \left(-J_{\perp} \nabla \omega_c(\boldsymbol{x}) - e \nabla \Phi(\boldsymbol{x})\right) \times \hat{\boldsymbol{h}}(\boldsymbol{x})}{\left(\nabla \times \boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel} \nabla \times \hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})} \\
&= \frac{\nabla \times \left(\boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel} \hat{\boldsymbol{h}}(\boldsymbol{x})\right) v_{\parallel} - \frac{c}{e} \hat{\boldsymbol{h}}(\boldsymbol{x}) \times \left(-J_{\perp} \nabla \omega_c(\boldsymbol{x}) - e \nabla \Phi(\boldsymbol{x})\right)}{\left(\nabla \times \left(\boldsymbol{A}(\boldsymbol{x}) + \frac{mc}{e} v_{\parallel} \hat{\boldsymbol{h}}(\boldsymbol{x})\right)\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})}.
\end{aligned} \tag{2.67}
$$

The definitions (2.48) - (2.50) can be identified in this equation, yielding

$$
\dot{\boldsymbol{x}} = \frac{\boldsymbol{B}^{\star}(\boldsymbol{x}, v_{\parallel}) v_{\parallel} - c \hat{\boldsymbol{h}}(\boldsymbol{x}) \times \boldsymbol{E}^{\star}(\boldsymbol{x})}{\boldsymbol{B}^{\star}(\boldsymbol{x}, v_{\parallel}) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x})}, \tag{2.68}
$$

which is identical to Eq.(2.54). Here, we note that

$$
\begin{aligned}
(\nabla \times \boldsymbol{A}^{\star}(\boldsymbol{x})) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x}) &= \boldsymbol{B}^{\star}(\boldsymbol{x}, v_{\parallel}) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x}) \\
&= \widetilde{\boldsymbol{B}}^{\star}(\boldsymbol{x}, H, J_{\perp}) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x}) \\
&= \left(\nabla \times \widetilde{\boldsymbol{A}}^{\star}(\boldsymbol{x})\right) \cdot \hat{\boldsymbol{h}}(\boldsymbol{x}).
\end{aligned} \tag{2.69}
$$

Second, let us also choose $v_\parallel$ as an independent coordinate in Eq. (2.65) and plug in the result of $\dot{\boldsymbol{x}}$ of Eq. (2.68),

$$\begin{aligned}\dot{v}_\parallel &= \frac{1}{m}\frac{1}{v_\parallel}\frac{\boldsymbol{B}^\star(\boldsymbol{x},v_\parallel)v_\parallel - c\hat{\boldsymbol{h}}(\boldsymbol{x})\times\boldsymbol{E}^\star(\boldsymbol{x})}{\boldsymbol{B}^\star(\boldsymbol{x},v_\parallel)\cdot\hat{\boldsymbol{h}}(\boldsymbol{x})}\cdot(-J_\perp\nabla\omega_c(\boldsymbol{x})-e\nabla\Phi(\boldsymbol{x})) \\ &= \frac{e}{m}\frac{\boldsymbol{B}^\star(\boldsymbol{x},v_\parallel)\cdot\boldsymbol{E}^\star(\boldsymbol{x}))}{\boldsymbol{B}^\star(\boldsymbol{x},v_\parallel)\cdot\hat{\boldsymbol{h}}(\boldsymbol{x})} - \frac{e}{m}\frac{1}{v_\parallel}\frac{c\hat{\boldsymbol{h}}(\boldsymbol{x})\times\boldsymbol{E}^\star(\boldsymbol{x})}{\boldsymbol{B}^\star(\boldsymbol{x},v_\parallel)\cdot\hat{\boldsymbol{h}}(\boldsymbol{x})}\cdot\boldsymbol{E}^\star(\boldsymbol{x})), \end{aligned} \tag{2.70}$$

which is identical to Eq.(2.55) since the second term vanishes here.

2.4 Illustration of guiding-center orbits

The derivation of the guiding-center equations in slowly-varying time-independent fields revealed the conservation of energy and the perpendicular adiabatic invariance,

$$H = \frac{mv_\parallel^2}{2} + J_\perp\frac{eB(\boldsymbol{x})}{mc} + e\Phi(\boldsymbol{x}) = \text{const.} \tag{2.71}$$

If we assume for the sake of simplicity the absence of an electric field, a gradient of the magnetic field modulus $B(\boldsymbol{x})$ must cause a force parallel to the magnetic field. This force can reflect particles in the direction of the magnetic field and is thus termed "magnetic mirror force". Further, a particle dichotomy emerges: (a) particles with sufficient parallel kinetic energy to overcome the magnetic mirror force are not reflected and thus termed "passing particles", (b) those particles which are reflected are termed "trapped particles".

In Fig. 2.1 a passing particle (a) and a trapped particle (b) guiding-center orbit of a Deuterium ion with a kinetic energy of 3 keV in the axisymmetric magnetic field configuration of ASDEX Upgrade is depicted. The guiding-center equations were integrated with an adaptive Runge-Kutta 4/5 integrator and the electromagnetic field was represented by splines. The blue transparent area in Fig. 2.1 shows the poloidal $\varphi = 0$ plane. We term the orbit intersections with this plane as "Poincaré plot" and we will further (in this thesis) utilize this graphical representation as an important tool to investigate the properties of guiding-center orbits.

Fig. 2.2 shows the Poincaré plot of (a) two passing and (b) two trapped 3 keV D ions in axisymmetric ASDEX Upgrade configuration in order to illustrate the drift motion due to the $\boldsymbol{B}\times\nabla B$ drift. In dependence of the signs of the parallel velocities $\sigma(v_\parallel)$ at the start, the particles drift either inside or outside the magnetic flux surface.

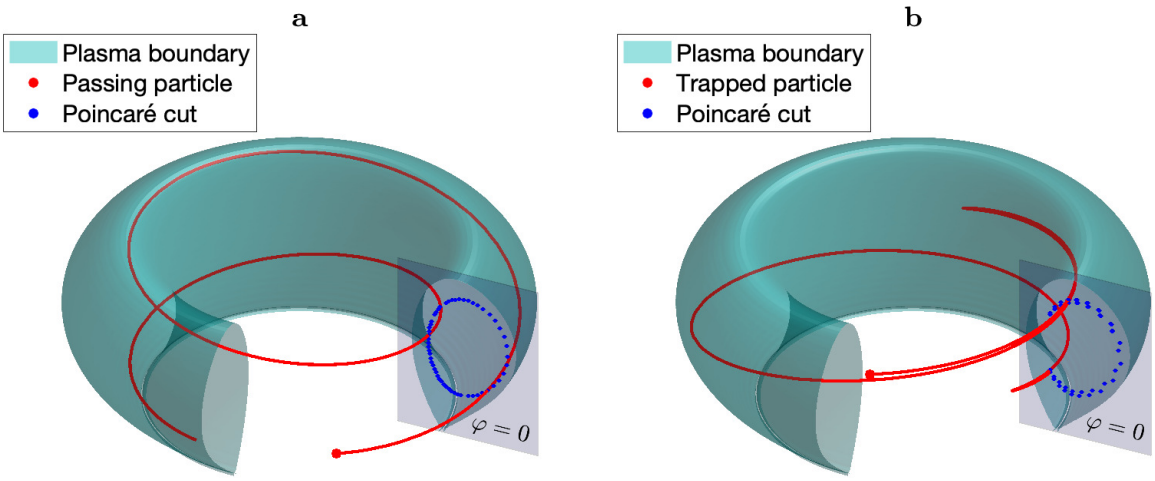

Figure 2.1: Illustration of (a) passing particle and (b) trapped particle guiding-center orbits of a Deuterium ion with a kinetic energy of 3 keV in the axisymmetric magnetic field configuration of ASDEX Upgrade. The blue transparent area shows the poloidal $\varphi = 0$ plane with blue dots indicating the intersections of the orbit with this plane (Poincaré plot). Red solid lines represent the guiding-center orbits.

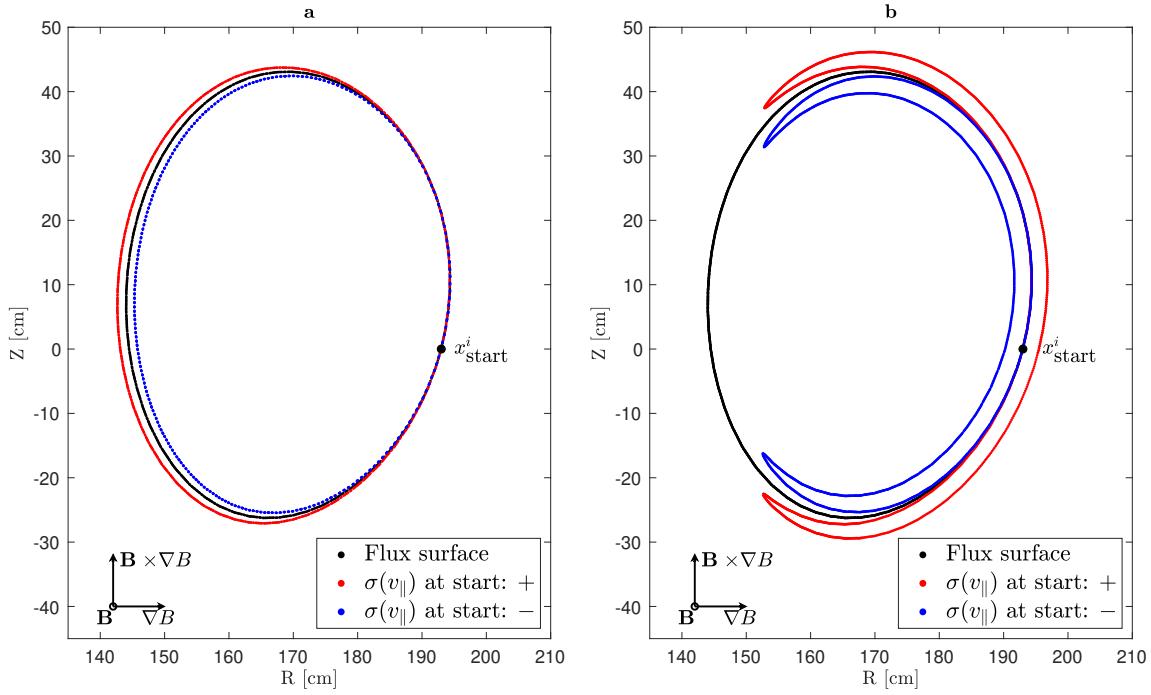

Figure 2.2: Poincaré plot ($\varphi = 0$) for illustration of guiding-center drift motion of (a) two passing and (b) two trapped 3 keV D ions in axisymmetric ASDEX Upgrade configuration. All particle orbits start from the same starting position x^i_{start}. The magnetic flux surface which contains the orbit starting position is marked in black. The signs of the parallel velocities $\sigma(v_\parallel)$ at the start are in accordance with the legends. *The figure is taken from Ref. [44] and modified.*

2.5 Summary of guiding-center dynamics

We have started the discussion of guiding-center dynamics with the Lagrangian in Cartesian coordinates corresponding to a charged particle with mass m and charge e moving in a (quasi-)static electromagnetic field,

$$L(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \frac{m}{2}|\dot{\boldsymbol{x}}|^2 + \frac{e}{c}\dot{\boldsymbol{x}} \cdot \mathbf{A}(\boldsymbol{x}) - e\Phi(\boldsymbol{x}). \tag{2.72}$$

By applying the small Larmor radius and slowly-varying field approximation we obtained the guiding-center phase-space Lagrangian,

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}}, v_\parallel, J_\perp, \dot{\phi}) &= \left(\frac{e}{c}\boldsymbol{A}(\boldsymbol{x}) + mv_\parallel \hat{\boldsymbol{h}}(\boldsymbol{x})\right) \cdot \dot{\boldsymbol{x}} - J_\perp \dot{\phi} - H(\boldsymbol{x}, v_\parallel, J_\perp) \quad \text{with} \\ H(\boldsymbol{x}, v_\parallel, J_\perp) &= \frac{mv_\parallel^2}{2} + J_\perp \omega_c(\boldsymbol{x}) + e\Phi(\boldsymbol{x}), \end{aligned} \tag{2.73}$$

which is gyrophase-independent and in which all fields are evaluated at the guiding-center position.

Furthermore, we have shown in this chapter that the guiding-center equations of motion can be obtained from the guiding-center Lagrangian by utilizing the Euler-Lagrange equations. The choice of independent coordinates, whether the parallel velocity $v_\parallel$ or the Hamiltonian H, does not influence the actual equations of motion. In their most compact form, the guiding-center equations of motion are denoted by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \frac{\nabla \times \boldsymbol{A}^\star v_\parallel}{\boldsymbol{B}_\parallel^\star}, \qquad \boldsymbol{A}^\star = \boldsymbol{A} + \frac{mc}{e} v_\parallel \hat{\boldsymbol{h}}, \qquad \boldsymbol{B}_\parallel^\star = (\nabla \times \boldsymbol{A}^\star) \cdot \hat{\boldsymbol{h}} \\ v_\parallel &= \sigma \sqrt{\frac{2}{m}(H - J_\perp \omega_c - e\Phi)}, \qquad \dot{J}_\perp = 0, \qquad \dot{\phi} = -\omega_c. \end{aligned} \tag{2.74}$$

Part II

Derivations and implementation

Chapter 3

Quasi-geometric integration of the guiding-center equations in piecewise linear toroidal fields

The content of this chapter including the figures can also be found in section II and appendix A and B of Ref. [1] formulated by the author and including some minor modifications.

In the following chapter we derive locally linear Hamiltonian guiding-center equations. We do not approximately solve Hamiltonian guiding-center equations in exact fields, but we solve the guiding-center equations exactly to computer accuracy in piecewise linearized fields.

3.1 Derivation of locally linear Hamiltonian guiding-center equations

"As a starting point, the equations of guiding-center motion, Eq. (11) of Ref. [5] equivalent to Eq. (A7) of Ref. [6], are expressed in general curvilinear coordinates x^i,

$$\dot{x}^i = \frac{v_\parallel \varepsilon^{ijk}}{\sqrt{g}\, B^*_\parallel} \frac{\partial A^*_k}{\partial x^j}, \qquad A^*_k = A_k + \frac{v_\parallel}{\omega_c} B_k. \tag{3.1}$$

Here, A_k, B_k, ω_c, Φ and $\sqrt{g}$ are the covariant components of the vector potential and the magnetic field, cyclotron frequency, electrostatic potential and the metric determinant, respectively, and $\sqrt{g}\, B^*_\parallel = \varepsilon^{ijk}(B_i/B)\partial A^*_k/\partial x^j$. Charge e_α and mass m_α of the considered species α enter $\omega_c = e_\alpha B/(m_\alpha c)$ together with the magnetic field modulus $B = \sqrt{B_k B^k}$ and the speed of light c. The equations of motion are considered with the invariants $w = m_\alpha v^2/2 + e_\alpha \Phi$ and $J_\perp = m_\alpha v_\perp^2/(2\omega_c)$ being total energy and perpendicular adiabatic invariant (gyroaction), respectively, and used as

independent phase-space variables. The latter variable is related to the magnetic moment μ by a constant factor, $J_\perp = \mu m_\alpha c/e_\alpha$. The parallel velocity $v_\parallel$ in (3.1) is not an independent variable but a known function of coordinates,

$$v_\parallel^2 = 2U, \qquad U = U(\mathbf{x}) = \frac{1}{m}\left(w - J_\perp \omega_c(\mathbf{x}) - e_\alpha \Phi(\mathbf{x})\right). \tag{3.2}$$

Due to the fact that the actual guiding-center orbits do not depend on the choice of their representing variables in phase-space, the parallel velocity $v_\parallel$ can also be treated as an independent variable. (Choosing $v_\parallel = v_\parallel(\mathbf{x}, w, J_\perp)$ or $w = w\left(\mathbf{x}, v_\parallel, J_\perp\right)$ is equivalent as long as Eq. (3.2) is kept exact.) By replacing the first expression of (3.2) with the differential equation

$$\dot{v}_\parallel = \frac{1}{v_\parallel} \dot{x}^i \frac{\partial U}{\partial x^i}, \tag{3.3}$$

the set (3.1) turns into

$$\begin{aligned} B_\parallel^* \sqrt{g}\, \dot{x}^i = \frac{\mathrm{d}x^i}{\mathrm{d}\tau} &= \varepsilon^{ijk}\left(v_\parallel \frac{\partial A_k}{\partial x^j} + 2U \frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \frac{B_k}{\omega_c}\frac{\partial U}{\partial x^j}\right), \\ B_\parallel^* \sqrt{g}\, \dot{v}_\parallel = \frac{\mathrm{d}v_\parallel}{\mathrm{d}\tau} &= \varepsilon^{ijk}\frac{\partial U}{\partial x^i}\left(\frac{\partial A_k}{\partial x^j} + v_\parallel \frac{\partial}{\partial x^j}\frac{B_k}{\omega_c}\right). \end{aligned} \tag{3.4}$$

Here, $v_\parallel$ should be treated as an independent variable only at its explicit occurrences, while the quantity $U(\mathbf{x})$ should only be treated as a function of coordinates as defined by the second expression of (3.2). Note that the invariants of motion remain in (3.4) as parameters entering the function of coordinates U. As long as the contra-variant components of the phase-space velocity defined by the right hand sides of set (3.4) are used for obtaining the exact solution of this set, both representations (3.2) lead to the same result. However, if the derivatives of these components are computed for the Jacobian as, e.g., in Appendix 3.3.1 the quantity U should be treated as a function of parallel velocity defined by the first of (3.2). In (3.4) the time variable is replaced by an orbit parameter τ related to time by $\mathrm{d}t = B_\parallel^* \sqrt{g}\mathrm{d}\tau$, and the time evolution is obtained implicitly from the integral $t(\tau)$.

The special form (3.4) allows to reduce computational effort and noise sensitivity by independently approximating the field quantities A_k, B_k/ω_c, ω_c and Φ by continuous piecewise linear functions. Thus, curvilinear coordinate space is split into tetrahedral cells with exact field values on the cell's vertices. Fig. 3.1 depicts such a real space illustration of a curvilinear field-aligned grid for the plasma core of a tokamak."[1] Within the tetrahedral cells the linear representation of the electromagnetic fields is a result of an interpolation but not of a regression which would be the case for

polyhedra with more than four vertices. "These tetrahedral cells must be specially oriented (explained below in section 4.2) in order to preserve an invariant of motion in the case of axisymmetry.

As a result of this piecewise field linearization, in each cell, the equations of motion (3.4) turn into a set of four linear ODEs with constant coefficients

$$\frac{\mathrm{d}z^i}{\mathrm{d}\tau} = a^i_l z^l + b^i, \tag{3.5}$$

in phase-space variables $z^i = x^i$ for $i = 1, 2, 3$ and $z^4 = v_\parallel$. The matrix elements are

$$\begin{aligned}
a^i_l &= \varepsilon^{ijk}\left(2\frac{\partial U}{\partial x^l}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \frac{\partial U}{\partial x^j}\frac{\partial}{\partial x^l}\frac{B_k}{\omega_c}\right) \qquad \text{for} \qquad 1 \le i, l \le 3, \\
a^i_4 &= \varepsilon^{ijk}\frac{\partial A_k}{\partial x^j} \qquad \text{for} \qquad 1 \le i \le 3, \\
a^4_l &= 0 \qquad \text{for} \qquad 1 \le l \le 3, \\
a^4_4 &= \varepsilon^{ijk}\frac{\partial U}{\partial x^i}\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c},
\end{aligned} \tag{3.6}$$

and components of vector b^i are

$$\begin{aligned}
b^i &= \varepsilon^{ijk}\left(2U_0\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c} + \left(\frac{B_k}{\omega_c}\right)_0\frac{\partial U}{\partial x^j}\right) \qquad \text{for} \qquad 1 \le i \le 3, \\
b^4 &= \varepsilon^{ijk}\frac{\partial U}{\partial x^i}\frac{\partial A_k}{\partial x^j},
\end{aligned} \tag{3.7}$$

where quantities with zero mean the value at the origin of the coordinates shifted in each tetrahedral cell to one of the cell's vertices,"[1]

$$U = U_0 + x^i\frac{\partial U}{\partial x^i}, \qquad \frac{B_k}{\omega_c} = \left(\frac{B_k}{\omega_c}\right)_0 + x^i\frac{\partial}{\partial x^i}\frac{B_k}{\omega_c}. \tag{3.8}$$

The derivatives in Eqs.(3.6)-(3.8) are taken by using the coefficients of three planar fits at the faces adjacent to the vertex to which the origin of the coordinates is shifted. The resulting sets of linear algebraic equations for derivatives are solved and stored before the actual orbit tracing.

It is worth to mention that the special combination of $v_\parallel$ being either an independent variable or a function of x^i and w in Eqs. (3.4) is chosen in order to obtain constant coefficients a^i_l and b^i inside the cells. It is not possible to achieve this with either $v_\parallel$ or w being independent coordinates alone. The coefficients a^i_l and b^i are constant due to the fact that the derivatives of linearly interpolated values are constants. In particular, on the right hand sides of (3.6) and (3.7), $v_\parallel$ does not occur explicitly as an independent coordinate but only implicitly via U. Further, U is only a function of

position (second of (3.2)) and, thus, its gradient is also given inside the tetrahedral cell, because ω_c and Φ are linear functions inside the cell.

"Since the piecewise constant coefficients of set (3.5) are discontinuous at the cell boundaries, orbit intersections with tetrahedra faces must be computed exactly when integrating particle trajectories.

In fact, a linear approximation of field quantities which locally breaks the physical connection between them does not destroy the Hamiltonian nature of the original set (3.1). Indeed, despite the approximation made, equation set (3.4) can still be cast to the non-canonical Hamiltonian form

$$\frac{\mathrm{d}z^i}{\mathrm{d}\tau} = \Lambda^{ij}\frac{\partial H}{\partial z^j}, \qquad \Lambda^{ij}(\mathbf{z}) = \left\{z^i, z^j\right\}_\tau, \tag{3.9}$$

where the Hamiltonian function is $H(\mathbf{z}) = v_\parallel^2/2 - U(\mathbf{x})$ and $\Lambda^{ij}(\mathbf{z})$ is an antisymmetric Poisson matrix. The latter is linked to Poisson brackets that are slightly re-defined from those in Ref. [45] due to a different orbit parameter,

$$\{f,g\}_\tau = b_*^i\left(\frac{\partial f}{\partial x^i}\frac{\partial g}{\partial v_\parallel} - \frac{\partial g}{\partial x^i}\frac{\partial f}{\partial v_\parallel}\right) + \varepsilon^{ijk}\frac{\partial g}{\partial x^i}\frac{\partial f}{\partial x^j}\frac{B_k}{\omega_c}, \qquad b_*^i = \varepsilon^{ijk}\left(\frac{\partial A_k}{\partial x^j} + v_\parallel\frac{\partial}{\partial x^j}\frac{B_k}{\omega_c}\right). \tag{3.10}$$

In the derivation above, one occurrence of $v_\parallel^2$ has been replaced by $2U(\mathbf{x})$ to obtain equation set (3.4). As long as this equality in the first of (3.2) is kept exact by a numerical scheme, this formal violation of the Poisson structure doesn't affect the final result.

The presented method integrates equations of motion to computer accuracy and thus exactly conserves invariants with respect to piecewise linear fields in order to fulfil this requirement. The invariants with respect to the original smooth fields oscillate within fixed bounds in a similar way as for conventional symplectic integrators[24]."[1]

Conservation properties and symplectic features of the locally linear equation set (3.5) are discussed in more detail in the sections 3.3 and 3.4.

It is worth to mention that despite the very accurate integration of the linear ODE set (3.5) certain errors are nevertheless present in that solution and can be accumulated over long time evolutions. If the total energy is kept exactly constant, this may lead to the consequence that the different two formal definitions of $v_\parallel$ (as an independent variable or as a function of w and x^i) result in different values. Therefore, everytime when a particle enters a tetrahedral cell, the total energy w is computed at the entry point from the guiding-center position x^i and the parallel velocity $v_\parallel$. This total energy value goes in the quantity U_0 (Eq. (3.8)) and is kept constant during the local integration of Eq. (3.5). Thus, w is renewed every time the particle enters a new cell,

and the errors due to inaccuracy of the integration of the local linear ODE sets do not lead to divergence between the two definitions of $v_\parallel$.

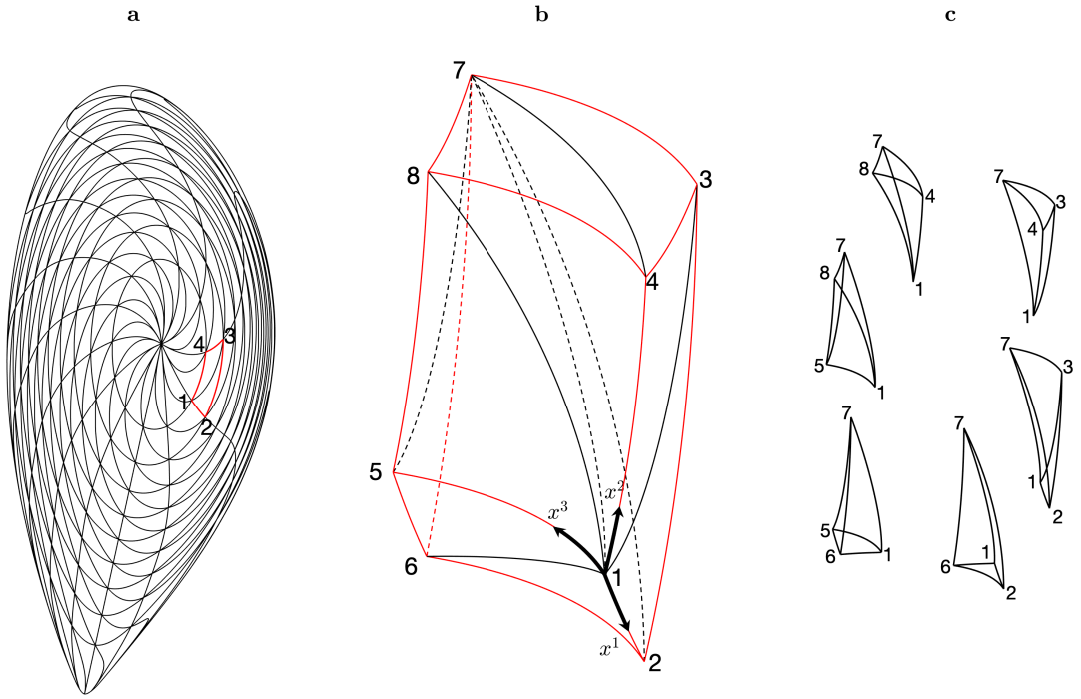

Figure 3.1: Real space illustration of field-aligned grid for the plasma core of a toroidal fusion device. Magnetic field lines are traced in general curvilinear coordinates (x^1,x^2,x^3) on N_s flux surfaces. At equidistant spacing in toroidal and poloidal direction a 3D grid consisting of $N_s \times N_\vartheta \times N_\varphi$ hexahedra is created.
(a) 2D poloidal projection of 3D grid with a marked hexahedron cell. (b) Magnification of marked hexahedron in (a) with indication how each hexahedron is split into six tetrahedral cells. The symmetry direction is along the coordinate x^3. (c) Six individual tetrahedral cells compose stackable hexahedron such that adjacent tetrahedra faces are congruent.

3.2 Numerical solution of locally linear guiding-center equations

"An approximate formal solution of set (3.5) in a single cell is given as a polynomial series of the orbit parameter,

$$\mathbf{z}(\tau) = \mathbf{z}_0 + \sum_{k=1}^{K} \frac{\tau^k}{k!} \left(\hat{\mathbf{a}}^{k-1} \cdot \mathbf{b} + \hat{\mathbf{a}}^k \cdot \mathbf{z}_0 \right), \tag{3.11}$$

where $\hat{\mathbf{a}}$ and $\mathbf{b}$ stand for matrix a^i_l and vector b^i, respectively, $\mathbf{z}(0) = \mathbf{z}_0$ is a starting point, and the exact solution is obtained in the limit $K \to \infty$. In the case $k = 1$ the

matrix $\hat{\mathbf{a}}^{k-1} = \hat{\mathbf{a}}^0$ is the identity matrix. It should be noted that for mild electric fields (validity domain of Eq. (3.1)) where the potential energy $e_\alpha \Phi$ is of the order of the kinetic energy (sub-sonic rotations) all elements of matrix $\hat{\mathbf{a}}$ except for a_4^i scale linearly with the Larmor radius, and in the zero Larmor radius limit the series expansion of the order $K = 2$ does already provide an exact solution. Therefore, the series expansion (3.11) converges rapidly in the case that the Larmor radius is small in comparison with the spatial scale of the electromagnetic field (see Eq. (3.13) below). Since intersections with cell boundaries (tetrahedra faces) must be computed exactly, the particle is pushed from cell boundary to cell boundary, if no intermediate position inside a cell, e.g. after a pre-defined time step, is required deliberately. In the latter case, the positions before and after the time step are located inside the cells, while the time step itself includes tracing of all orbit intersections with cell boundaries between these positions.

An orbit intersection with a tetrahedron is found on exit as an intersection with one of four planes

$$F^\alpha(\mathbf{z}) \equiv n_i^{(\alpha)} \left(x^i - x_{(\alpha)}^i \right) = 0, \qquad \alpha = 1, \ldots, 4 \tag{3.12}$$

reached in the smallest positive time, "exit time", from the starting position $x^i(\tau_0)$ located either on the cell boundary or inside the cell , see Fig. 3.2. Here, $n_i^{(\alpha)}$ and $x_{(\alpha)}^i$ are the (constant) normal to the plane containing tetrahedron face α and coordinates of some vertex on that face, respectively. In the case that the starting position is located on the cell boundary (due to boundary-boundary-pushings as in Fig. 3.2), the exit time coincides with the "dwell time" of the particle inside the cell. If an intermediate stop (inside the cell) is deliberately required, the dwell time is the sum of the time to reach this stop ("entry time") and the exit time.

With substitution of the orbit, $\mathbf{z} = \mathbf{z}(\tau)$, Eqs. (3.12) are nonlinear equations with respect to the orbit parameter, $F^\alpha(\mathbf{z}(\tau)) = 0$, which should be solved numerically. They become algebraic and can be solved analytically if an approximate solution (3.11) is used with $K \leq 4$, as explained below in detail.

Both cases have been implemented in a Fortran program with the name **G**uiding-center **OR**bit **I**ntegration with **L**ocal **L**inearization **A**pproach (**GORILLA**).

As a matter of completeness, it is worth mentioning that a third similar to Ref. [35] variant exists to solve the linear equation set (3.5) exactly in terms of exponential functions of eigenvalues in the eigenvector basis (as any other linear equation set). Formally, this solution corresponds to the limit $K \to \infty$ in Eq. (3.11) where infinite sums can be expressed in terms of the exponential function $\exp(\tau \hat{a})$. This method, however, has two drawbacks. First, Eqs. (3.12) turn into nonlinear transcendental equations which have to be solved numerically up to computer accuracy. Second, the

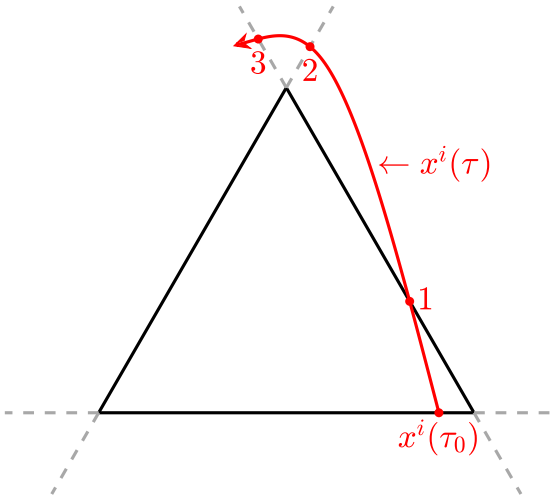

Figure 3.2: Illustration of intersections of the orbit $x^i(\tau)$ with planes confining the cell. The three-dimensional tetrahedral cell is depicted as a two-dimensional triangle in the interest of simplification. The particle enters the cell at $x^i(\tau_0)$ and leaves the cell at ①. Other intersections at the points ② and ③ are not realized. The correct orbit-tetrahedron intersection ① is reached in the smallest positive time among all intersections.

analytical solution results in strong cancellation errors in the case of small Larmor radii which is exactly the case where the finite series expansion results in negligible errors. Therefore, we refrain from such a solution.

In its first variant (GORILLA RK4), the orbit intersections with tetrahedra faces are computed numerically by solving equation set (3.5) in each cell with a single step of the Runge-Kutta 4 method embedded into an iterative scheme using Newton's method to obtain the integration time $\Delta\tau$ required to reach the cell boundary (exit time $\tau_e = \Delta\tau$). This iterative scheme predominantly converges after roughly two Newton iterations, due to an analytic estimation for the necessary initial step length using an approximate parabolic solution of ODE set (3.5) taken in zeroth order in Larmor radius, $a_l^i = 0$ for $1 \leq i, j \leq 3$ and $a_4^4 v_\parallel(\tau) = a_4^4 v_{\parallel,0}$.

Nevertheless, in the case of numerically challenging orbits (tangential to a tetrahedron face or almost intersecting with a tetrahedron's edges or vertices), several special cases appear in which the iterative scheme does not converge, and those cases must be treated separately in a computationally more expensive manner. Since such cases appear only rarely, the additionally required computational effort is negligible in comparison to the standard procedure.

A single RK4 integration step per iteration is sufficient because the magnetic field is uniform within a cell. Respectively, the error of the RK4 method strongly scales with the Larmor radius ρ and can be brought below computer accuracy by a moderate grid refinement. Namely, in a tokamak geometry the error of a single step traversing the cell can be estimated as

$$\frac{\delta R(\Delta\tau)}{a} \sim \frac{\rho^3}{q^4 R^3} \Delta\varphi^5, \tag{3.13}$$

where R, a, q and $\Delta\varphi$ denote major radius, plasma radius, safety factor and toroidal cell length, respectively.

Due to the scaling of the error with ρ, for particles with mild energies (thermal electrons and ions) a less elaborate algorithm (GORILLA Poly) can be realized. By truncating the summation of Eq. (3.11) at $K = 2$, 3 or 4 one obtains approximate solutions of various orders in Larmor radius. With these solutions, equations for the exit time τ_e are algebraic equations which are solved analytically up to order $K = 4$, by finding the smallest positive root of

$$\mathbf{n}^\alpha \cdot \left(\mathbf{z}_0 + \sum_{k=1}^{K} \frac{\tau_e^k}{k!} \left(\hat{\mathbf{a}}^{k-1} \cdot \mathbf{b} + \hat{\mathbf{a}}^k \cdot \mathbf{z}_0 \right) - \mathbf{z}^\alpha \right) = 0, \tag{3.14}$$

where $\mathbf{n}^\alpha = \left(n_1^{(\alpha)}, n_2^{(\alpha)}, n_3^{(\alpha)}, 0 \right)$ are face normals (see above) and $\mathbf{z}^\alpha = \left(x_{(\alpha)}^1, x_{(\alpha)}^2, x_{(\alpha)}^3, 0 \right)$. Furthermore, by using the quadratic polynomial solution $K = 2$ of Eq. (3.14), the appropriate orbit intersection plane α can be predicted for higher orders $K = 3$ or 4, which predominantly reduces the number of higher order root finding operations from four to one. In the numerical implementation, series (3.11) which contains matrix products is not directly evaluated as written above. Instead, various sub-products are preliminarily evaluated and stored for grid cells, minimizing the number of matrix products that need to be evaluated directly. The finite series solution (3.11) allows to analytically evaluate also various integrals over the dwell time needed for scoring of macroscopic plasma plasma parameters in Monte Carlo procedures. In Appendix **??**, such integrals of $v_\parallel$ and of $v_\perp^2$ and $v_\parallel^2$ are given, which respectively determine parallel plasma flow and components of the pressure tensor in Chew-Goldberger-Low form essential for computation of equilibrium plasma currents."[1]

3.3 Hamiltonian structure of locally linear guiding-center equations

3.3.1 Liouville's theorem

"Let us check that Liouville's theorem is fulfilled for the piecewise linear set represented locally by Eq. (3.5) despite the discontinuities of the phase-space velocity at cell boundaries and the fact that the vector potential is not exactly linked with magnetic field, i.e. that the relation $\omega_c \left(\mathbf{B}/\omega_c \right) = \nabla \times \mathbf{A}$ is not fulfilled anymore by the piecewise linear approximations $\omega_c^{(L)}$, $\left(B_i/\omega_c \right)^{(L)}$ and $A_i^{(L)}$ of the respective magnetic field parameters within the exact curvilinear coordinate metrics. (In this section we use sub- or superscripts (L) for quantities representing the piecewise linear electromagnetic field.) Formally, independent interpolation errors in these originally related quantities

result in a slight distortion of the metric tensor, but produce a consistent set of equations of motion with a divergence-free modified magnetic field $\mathbf{B}^*_{(L)}$ being a necessary condition for correct geometric properties. Namely, $\mathbf{B}^*_{(L)} = \nabla \times \mathbf{A}^\star_{(L)}$ remains divergence-free by using an interpolation of covariant components A_k and covariant unit vector components $B_k/B \propto B_k/\omega_c$ that are both given in a curl-compatible representation. The gyrofrequency $\omega_c \propto B$ just plays the role of a scalar potential in Eq. (3.2) and doesn't affect the symplectic structure.
First, we notice (see Eq. (44) of Ref. [5]) that the phase-space Jacobian of the coordinate set $\mathbf{y} = (\mathbf{x}, J_\perp, \phi, v_\parallel)$ where ϕ is the gyrophase is

$$J = \frac{\partial(\mathbf{r}, \mathbf{p})}{\partial(\mathbf{x}, J_\perp, \phi, v_\parallel)} = \frac{m_\alpha e_\alpha}{c}\sqrt{g}B^*_\parallel. \tag{3.15}$$

Liouville's theorem states that divergence of the phase-space flow velocity is identically zero, $J^{-1}\partial\left(J\dot{y}^i\right)/\partial y^i \equiv 0$, yielding

$$\frac{\partial}{\partial x^i}J\dot{x}^i_{(L)} + \frac{\partial}{\partial J_\perp}J\dot{J}_\perp + \frac{\partial}{\partial \phi}J\dot{\phi} + \frac{\partial}{\partial v_\parallel}J\dot{v}^{(L)}_\parallel = \frac{\partial}{\partial x^i}J\dot{x}^i_{(L)} + \frac{\partial}{\partial v_\parallel}J\dot{v}^{(L)}_\parallel = 0, \tag{3.16}$$

where two terms vanished due to $\dot{J}_\perp = 0$ and the independence of $\dot{\phi}$ of the gyrophase. The remaining phase-space velocity components are defined in accordance with (3.4) and a subsequent piecewise linear approximation as

$$\dot{x}^i_{(L)} = \frac{1}{\sqrt{g}B^*_\parallel}\left(\frac{\mathrm{d}x^i}{\mathrm{d}\tau}\right)_{(L)}, \qquad \dot{v}^{(L)}_\parallel = \frac{1}{\sqrt{g}B^*_\parallel}\left(\frac{\mathrm{d}v_\parallel}{\mathrm{d}\tau}\right)_{(L)}. \tag{3.17}$$

Here, $B^*_\parallel$ corresponds to the exact field while the derivatives with respect to τ are given for the linearized field,

$$\begin{aligned}
\left(\frac{\mathrm{d}x^i}{\mathrm{d}\tau}\right)_{(L)} &= \varepsilon^{ijk}\left(v_\parallel\frac{\partial A^{(L)}_k}{\partial x^j} + v^2_\parallel\frac{\partial}{\partial x^j}\left(\frac{B_k}{\omega_c}\right)^{(L)} + \left(\frac{B_k}{\omega_c}\right)^{(L)}\frac{\partial U^{(L)}}{\partial x^j}\right), \\
\left(\frac{\mathrm{d}v_\parallel}{\mathrm{d}\tau}\right)_{(L)} &= \varepsilon^{ijk}\frac{\partial U^{(L)}}{\partial x^i}\left(\frac{\partial A^{(L)}_k}{\partial x^j} + v_\parallel\frac{\partial}{\partial x^j}\left(\frac{B_k}{\omega_c}\right)^{(L)}\right).
\end{aligned} \tag{3.18}$$

Note that here we had to replace the quantity $U^{(L)}$ (but not its spatial derivatives computed for constant w) back with its expression (3.2) via $v_\parallel$ (which remains exact also for the linearized field), because $U^{(L)}$ is the only quantity containing the total energy w which is not a constant parameter but a function of phase-space coordinates for the derivatives in (3.16). Substitution of (3.18) in (3.17) and subsequently (3.17) in (3.16) yields

$$\frac{m_\alpha e_\alpha}{c}\left(\frac{\partial}{\partial x^i}\left(\frac{\mathrm{d}x^i}{\mathrm{d}\tau}\right)_{(L)}+\frac{\partial}{\partial v_\parallel}\left(\frac{\mathrm{d}v_\parallel}{\mathrm{d}\tau}\right)_{(L)}\right)=$$
$$=\frac{m_\alpha e_\alpha}{c}\varepsilon^{ijk}\left(v_\parallel\frac{\partial^2 A_k^{(L)}}{\partial x^i\partial x^j}+v_\parallel^2\frac{\partial^2}{\partial x^i\partial x^j}\left(\frac{B_k}{\omega_c}\right)^{(L)}\right.$$
$$\left.+\left(\frac{B_k}{\omega_c}\right)^{(L)}\frac{\partial^2 U^{(L)}}{\partial x^i\partial x^j}+\frac{\partial U^{(L)}}{\partial x^j}\frac{\partial}{\partial x^i}\left(\frac{B_k}{\omega_c}\right)^{(L)}+\frac{\partial U^{(L)}}{\partial x^i}\frac{\partial}{\partial x^j}\left(\frac{B_k}{\omega_c}\right)^{(L)}\right)=$$
$$=0 \tag{3.19}$$

which proves the theorem due to the symmetry of the expression in parentheses over i and j. Note that generally the second derivatives in (3.19) formally contain Dirac δ functions, because the first derivatives of the piecewise linear functions are discontinuous at the cell boundaries. However, the δ functions do not appear at a given boundary if one linearly transforms the spatial variables x^i so that one of the coordinate planes, e.g. $x^1 = \text{const.}$, contains the respective tetrahedral cell face. This makes it evident that the normal component of the spatial velocity $(\mathrm{d}x^1/\mathrm{d}\tau)_{(L)}$ is continuous at the cell boundary since it does not contain discontinuous derivatives over x^1 while the discontinuous tangential components $(\mathrm{d}x^{2,3}/\mathrm{d}\tau)_{(L)}$ are not differentiated in (3.19) across the boundary (over x^1)."[1]

3.3.2 Symplecticity

"Let us show that the piecewise linearization of the electromagnetic field does not affect the symplectic properties of the orbit geometry by using a similar to Ref. [5] derivation of the guiding-center equations. First, we introduce the Lagrangian for the piecewise linear field,

$$L^{(L)}=\frac{e_\alpha}{c}A_i^{*(L)}\dot{x}^i-J_\perp\dot{\phi}-H^{(L)}, \tag{3.20}$$

where the independent phase-space variables are $\mathbf{y}=(\mathbf{x}, J_\perp, \phi, v_\parallel)$ and

$$A_i^{*(L)}=A_i^{(L)}+v_\parallel\left(\frac{B_i}{\omega_c}\right)^{(L)}, \qquad H^{(L)}=\omega_c^{(L)}J_\perp+\frac{m_\alpha v_\parallel^2}{2}+e_\alpha\Phi^{(L)}. \tag{3.21}$$

The corresponding Euler-Lagrange equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L^{(L)}}{\partial\dot{y}^i}=\frac{\partial L^{(L)}}{\partial y^i}, \tag{3.22}$$

are explicitly given as

$$\frac{e_\alpha}{c}\dot{x}^j\left(\frac{\partial A_i^{*(L)}}{\partial x^j}-\frac{\partial A_j^{*(L)}}{\partial x^i}\right)+\frac{e_\alpha}{c}\left(\frac{B_i}{\omega_c}\right)^{(L)}\dot{v}_\parallel+\frac{\partial}{\partial x^i}\left(\omega_c^{(L)}J_\perp+e_\alpha\Phi^{(L)}\right)=0,$$

$$\dot{\phi}+\omega_c^{(L)}=0,\qquad \dot{J}_\perp=0,\qquad \frac{e_\alpha}{c}\left(\frac{B_i}{\omega_c}\right)^{(L)}\dot{x}^i-m_\alpha v_\parallel=0. \tag{3.23}$$

Making a convolution of the first vector equation of (3.23) with the tensor $\varepsilon^{ikl}\left(B_k/\omega_c\right)^{(L)}$ and using the last of Eqs. (3.23) one obtains an explicit expression for $\dot{x}^i$. The convolution of the same vector equation with the vector $\varepsilon^{ikl}\partial A_l^{*(L)}/\partial x^k$ yields an expression for $\dot{v}_\parallel$. All these phase-space velocity components are expressed via (3.18) as follows

$$\dot{x}^i=\frac{1}{\left(\sqrt{g}B_\parallel^*\right)^{(L)}}\left(\frac{\mathrm{d}x^i}{\mathrm{d}\tau}\right)_{(L)},\qquad \dot{v}_\parallel=\frac{1}{\left(\sqrt{g}B_\parallel^*\right)^{(L)}}\left(\frac{\mathrm{d}v_\parallel}{\mathrm{d}\tau}\right)_{(L)}, \tag{3.24}$$

and differ from (3.17) by the first factor, where

$$\left(\sqrt{g}B_\parallel^*\right)^{(L)}=\frac{e_\alpha}{m_\alpha c}\varepsilon^{ijk}\left(\frac{B_i}{\omega_c}\right)^{(L)}\frac{\partial A_k^{*(L)}}{\partial x^j}. \tag{3.25}$$

By replacing in (3.15) the exact expression $\sqrt{g}B_\parallel^*$ with the linearized one of (3.25), we obtain the Jacobian of the phase-space coordinates $\mathbf{y}$ which formally have slightly modified dependence on $(\mathbf{r},\mathbf{p})$.

The preservation of the symplectic properties of the phase-space flow by a piecewise linearization of the field is obvious in case of 3D toroidal fields with embedded flux surfaces. Using the canonical straight field line flux coordinate system[28] $\mathbf{x}=(r,\vartheta,\varphi)$ where $A_r=B_r=0$ and, respectively, $A_r^*=0$ one can introduce canonical momenta $\mathbf{P}\equiv(P_1,P_2,P_3)=(P_\vartheta,P_\varphi,J_\perp)$ which are conjugates to the coordinates $\mathbf{Q}\equiv(Q^1,Q^2,Q^3)=(\vartheta,\varphi,-\phi)$ with

$$P_\vartheta=\frac{e_\alpha}{c}A_\vartheta^{*(L)},\qquad P_\varphi=\frac{e_\alpha}{c}A_\varphi^{*(L)}. \tag{3.26}$$

Then the Lagrangian (3.20) and the Hamiltonian (3.21) respectively take the form

$$\begin{aligned}L^{(L)}&=P_i\dot{Q}^i-H^{(L)},\\ H^{(L)}&=\omega_c^{(L)}J_\perp+\frac{m_\alpha}{2}\left(\left(\frac{B_\varphi}{\omega_c}\right)^{(L)}\right)^{-2}\left(\frac{c}{e_\alpha}P_\varphi-A_\varphi^{*(L)}\right)^2+e_\alpha\Phi^{(L)}.\end{aligned} \tag{3.27}$$

Consequently, the Euler-Lagrange equations (3.22) for the variable set $\mathbf{y}=(\mathbf{Q},\mathbf{P})$

result in Hamilton's equations. Note that due to the continuity of the piecewise linearization the mutual relations between canonical and non-canonical variables and the Hamiltonian are continuous at the cell boundaries. Respectively continuous are the orbits which fulfill the usual symplectic relations of mapping in time.
Note that both, Eqs. (3.17) and Eqs. (3.24) result in the same phase space orbit geometry with the latter having Hamiltonian time dynamics described by

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = \left(\sqrt{g}B_{\parallel}^{*}\right)^{(L)}. \tag{3.28}$$

High accuracy of this dynamics, however, is not important for the steady state problems which are here of main interest. Since none of the coefficients of the kinetic equation depend in this case on the time variable, it can be replaced by τ resulting in

$$\frac{\partial f}{\partial \tau} + \{f, H\}_{\tau} = -\frac{\partial}{\partial y^i} J F_C^i(f) + Jq, \tag{3.29}$$

where $\{a, b\}_{\tau}$ are modified Poisson brackets (3.10), J is the Jacobian (3.15), $F_C^i(f)$ is phase-space flux density due to collisions, and q is some phase-space particle source. Obviously, the subsequent linearization of the fields does not violate the conservation properties of Eq. (3.29). Moreover, Boltzmann's distribution $f = f_B(H)$ remains to be a steady state solution since it commutes with the Hamiltonian in the Vlasov part and results in $F_C^i(f_B) = 0$ for the background in the thermodynamic equilibrium."[1]

3.4 Analytical integrals of velocity powers over the dwell time

"In this section it is presented how analytical expressions for the dwell time integrals of $v_{\perp}^2$, $v_{\parallel}$ and $v_{\parallel}^2$ can be obtained. Furthermore, the latter quantity is exemplary given as an explicit expression.
We start with the exact polynomial series solution of ODE set (3.5)

$$\mathbf{z} = \mathbf{z}_0 + \sum_{k=1}^{\infty} \frac{\tau^k}{k!} \left(\hat{\mathbf{a}}^{k-1} \cdot \mathbf{b} + \hat{\mathbf{a}}^k \cdot \mathbf{z}_0\right), \tag{3.30}$$

already given in Eq. (3.11). For the parallel velocity $v_{\parallel} = z^4$ this series can be written in form of a shifted exponential function

$$v_{\parallel}(\tau) = e^{\alpha\tau}\left(v_{\parallel,0} + \frac{\beta}{\alpha}\right) - \frac{\beta}{\alpha}, \tag{3.31}$$

where α, β and $v_{\parallel,0}$ stand for the matrix element a_{44}, the vector component b^4 and the initial value for the parallel velocity at the cell entry, respectively. After squaring Eq. (3.31), one can perform a Taylor series expansion of the orbit parameter up to the 4^{th} order,

$$\begin{aligned} v_\parallel^2(\tau) &\approx v_{\parallel,0}^2 + \tau(2\beta v_{\parallel,0} + 2\alpha v_{\parallel,0}^2) + \tau^2\left(\beta^2 + 3\alpha\beta v_{\parallel,0} + 2\alpha^2 v_{\parallel,0}^2\right) \\ &+ \tau^3\left(\alpha\beta^2 + \frac{7}{3}\alpha^2\beta v_{\parallel,0} + \frac{4}{3}\alpha^3 v_{\parallel,0}^2\right) + \frac{1}{12}\tau^4\left(7\alpha^2\beta^2 + 15\alpha^3\beta v_{\parallel,0} + 8\alpha^4 v_{\parallel,0}^2\right). \end{aligned} \tag{3.32}$$

This is the highest order, where an algebraic expression of the dwell time t_d to pass a cell can be found. In order to obtain the dwell time integral of $v_\parallel^2$, its polynomial representation can simply be integrated

$$\begin{aligned} \int_0^{t_d} v_\parallel^2(t)\mathrm{d}t &\approx C\left(\tau_d v_{\parallel,0}^2 + \frac{1}{2}\tau_d^2(2\beta v_{\parallel,0} + 2\alpha v_{\parallel,0}^2)\right. \\ &+ \frac{1}{3}\tau_d^3\left(\beta^2 + 3\alpha\beta v_{\parallel,0} + 2\alpha^2 v_{\parallel,0}^2\right) \\ &+ \frac{1}{4}\tau_d^4\left(\alpha\beta^2 + \frac{7}{3}\alpha^2\beta v_{\parallel,0} + \frac{4}{3}\alpha^3 v_{\parallel,0}^2\right) \\ &+ \left.\frac{1}{60}\tau_d^5\left(7\alpha^2\beta^2 + 15\alpha^3\beta v_{\parallel,0} + 8\alpha^4 v_{\parallel,0}^2\right)\right), \end{aligned} \tag{3.33}$$

where $C = \mathrm{d}t/\mathrm{d}\tau = B_\parallel^*\sqrt{g} = \text{const.}$ within the cell. Here, this lowest order approximation of C is intended, because for the moments of a steady state (3.29) only the integrals over the orbit parameter τ are needed to be computed accurately. The reason is that ther error of such an approximation is always small as long as the field quantities are well resolved by the grid. In turn, the cross-field drift terms which formally scale with the Larmor radius must be accounted accurately in τ_d as well as the respective integrals because they determine the orbit dynamics near the banana tips. In the case that higher order accuracy of the moments with respect to the grid size is required the derivative $\mathrm{d}t/\mathrm{d}\tau$ must be used in its form (3.28). Due to Eq. (3.25) this derivative is a product of a linear function of the coordinates and a linear function of $v_\parallel$. Therefore, the dwell time t_d and the time integrals can still be computed analytically leading to somewhat more complicated expressions. Note that here the dwell time τ_d may be a sum of entry and exit times in the case of a pre-defined time step.

Clearly, the dwell time integral of $v_\parallel$ requires to omit squaring of Eq. (3.31) and to proceed straightforwardly in the same manner.

Moreover, the squared perpendicular velocity $v_\perp^2$ is purely a function of position inside a tetrahedral cell due to its proportionality with the cyclotron frequency, $v_\perp^2 = 2\omega_c\frac{J_\perp}{m}$,

which is a piecewise linear function of position in the quasi-geometric orbit integration formulation. Thus, the dwell time integral of $v_\perp^2$ is obtained via an integral along the orbit which can easily be computed by using its polynomial representation given in Eq. (3.30)."[1]

Chapter 4

Implementation of the guiding-center orbit code GORILLA

The quasi-geometric integration method derived in chapter 3 has been implemented in a Fortran program with the name **G**uiding-center **OR**bit **I**ntegration with **L**ocal **L**inearization **A**pproach (**GORILLA**). The code has been made publicly available on GitHub as scientific open source software under the MIT license; see Ref. [2]. Further, the source code for GORILLA has been archived on Zenodo with the linked DOI of Ref. [46]. GORILLA has been developed in collaboration of TU Graz (ÖAW), IPP Garching, and IPP Kharkov (KIPT), whereby the author contributed as main developer.

A first proof of concept of GORILLA's underlying algorithm was originally presented in detail in the author's Master's thesis; see Ref. [44]. In this original realization, the guiding-center orbits were integrated in cylindrical coordinates. Furthermore, the tetrahedral grid which is necessary for this approach was not field-aligned, instead the vertices of the tetrahedra were uniformly distributed along the coordinate contours of a cylindrical coordinate system. Due to structural limitations and deficiencies of the original algorithm, as well as new insights, this code was jointly re-factored and extended in a cooperation between L. Bauer and the author, including:

- introduction of flux coordinates
- generation of a field-aligned grid
- complete re-factoring of the numerical approach with Runge-Kutta
- implementation of an analytical approach with polynomials
- suitable code preparation for physics applications
- conditioning for analysis of the underlying quasi-geometric integration method

In the following chapter, an overview of the structure of this guiding-center orbit code and the working principle of its modules and subroutines is shown with the intention of presenting the code in a concise and unified manner. Thus, this chapter can also be found in a slightly modified version in the documentation of GORILLA; see Ref. [46]. Some parts of the content of this chapter were already published before in the Master's thesis of L. Bauer, see Ref. [47], which was co-supervised by the author. Modifications (including some figures), extensions and conflations were performed by the author. Whenever already published content of Ref. [47] which was firmly co-developed by the author seemed not improvable with respect to compactness and clarity, this content is literally quoted in this thesis.

4.1 General structure of GORILLA

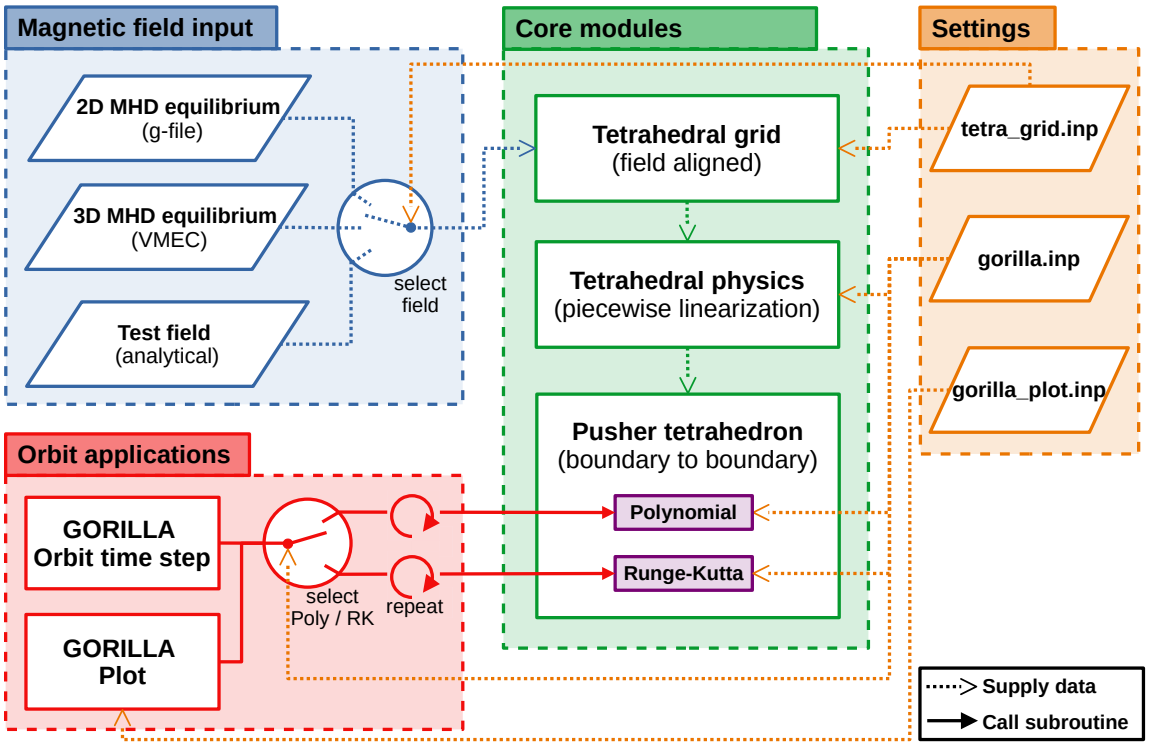

Figure 4.1: General overview of the structure of GORILLA: Core modules, magnetic field input, settings and orbit applications. Dotted lines with line-arrows indicate the supply of data for the subsequent execution of subroutines. Solid lines with filled arrows show the direction of the subroutine calls. (Data flow of called subroutine return values is implied in the opposite direction.)

For various simulations in magnetic confinement fusion, direct modeling of guiding-center particle orbits is utilized, e.g. global kinetic computations of quasi-steady plasma parameters or fast alpha particle loss estimation for stellarator optimization; see introduction of this thesis. In such complex simulations a simple interface for the

guiding-center orbit integration part is needed. Specifically, the initial condition in five-dimensional phase-space is provided (e.g. guiding-center position, parallel and perpendicular velocity) and the main interest is in the condition after a prescribed time step while the integration process itself is irrelevant. Such a pure "orbit time step routine" acting as an interface with a plasma physics simulation is provided ("*GORILLA orbit time step*"). In the context of this work, however, the integration process itself is of great interest as well and a program allowing the detailed analysis of guiding-center orbits, their respective invariants of motion and Poincaré plots is thus provided as well ("*GORILLA Plot*"). Both orbit application routines can be used independently from each other, whereby they utilize the same underlying modules and subroutines.

In Fig. 4.1 a general overview of the structure of GORILLA is depicted, in which these two orbit application routines can be seen in a red box. The blue box shows the input options for the magnetic field, which can be either provided by magnetohydrodynamics (MHD) equilibria in 2D (e.g. EFIT[48,49]) and in 3D (e.g. VMEC[50,51], GPEC[52,53]) or by an analytical test field.

The major part of GORILLA can then be divided in three separate core modules, which are depicted in green. First, the *tetrahedral grid module* loads the magnetic field data and generates a 3D field-aligned grid consisting of tetrahedra. Second, in the *tetrahedral physics module* the electromagnetic field is linearly interpolated within the tetrahedra of the grid such that globally (in the plasma core) the field is represented by piecewise linear functions. Further, normal vectors describing the planes which confine the tetrahedra and which are later needed are computed. Third, the actual guiding-center orbit integration is performed in the *pusher tetrahedron module*, where "a particle is pushed from tetrahedral cell boundary to cell boundary, if no intermediate position inside a cell, e.g. after a pre-defined time step, is required deliberately. The positions before and after the time step are, in general, located inside the cells while the time step itself includes tracing of all orbit intersections with cell boundaries between these positions."[1] A numerical *Runge-Kutta* and a *Polynomial* option, as briefly described in section 3.2, are implemented for the pushing of a particle in this manner. These options are depicted in violet boxes. Eventually, one of the two pushing options is repeatedly called in a loop (within the orbit applications) until the prescribed time has elapsed.

All settings for the grid generation, the orbit integration and the plotting of full guiding-center orbits, Poincaré sections and the time evolution of invariants of motion are governed by the input files `tetra_grid.inp`, `gorilla.inp` and `gorilla_plot.inp` which are shown in orange. E.g., a switch parameter in `gorilla.inp` determines

whether the *Runge-Kutta* or the *Polynomial* option is used.

4.2 Field-aligned tetrahedral grid with linearized electromagnetic field quantities

In this section an overview is given of how the grid which is used in GORILLA is generated. This grid consists of tetrahedral cells and is aligned to the magnetic field. A detailed description of the grid generation (with some modifications) including also a non-field-aligned grid can be found in Ref. [47]. Furthermore, inside the tetrahedral cells the quantities describing the electromagnetic field are independently approximated by linear functions.

Magnetic field input:
As stated in section 4.1, the magnetic field must be either provided by an MHD equilibrium or by an analytical test field. As shown below, for the utilization in GORILLA a field-aligned grid has an inherent benefit with respect to the linear interpolation of the covariant component of the vector potential when this is represented in flux coordinates. Without going into details, the working principle of GORILLA requires neither flux coordinates nor a field-aligned grid. However, non-flux coordinates, e.g. cylindrical coordinates, affect the shape of guiding-center orbits significantly and can also lead to artificial chaos in the case of three-dimensional fields. These circumstances are explained and illustrated in detail in chapter 5.
Without loss of generality, we use (straight field-line) symmetry flux coordinates[54] (s, ϑ, φ) here, where s denotes the flux label (normalized poloidal or toroidal flux), ϑ the symmetry flux poloidal angle and φ the toroidal angle, respectively. A prerequisite for the generation of a field-aligned grid to be used in GORILLA is that the following quantities which represent the electromagnetic field are available as functions of arbitrary positions inside the plasma core: A_k, B_k, ω_c, Φ and $\sqrt{g}$, which are the covariant components of the vector potential and the magnetic field, the cyclotron frequency, the electrostatic potential and the metric determinant, respectively. If these are not inherently available through analytic functions (in the case of a test field), interpolation of data points (in the case of MHD equilibria) and appropriate transformation to symmetry flux coordinates (SFC) are needed. Here, it should be mentioned that the divergence freeness of the magnetic field must not be violated. Since it is not an inherent part of this work, a description how to achieve such a transformation and interpolation is omitted, instead we assume that the above mentioned quantities are available as functions of SFC. A detailed description can be found in Ref. [47].

Requirements and structure of the tetrahedral grid:
Several requirements exist that must hold for a tetrahedral grid in order to function correctly while being used in GORILLA. "These requirements are:

1. The three-dimensional spatial domain, which is relevant for calculations, must be fully covered by non-overlapping tetrahedra. In this application tetrahedra are necessary since field quantities are used in a piecewise linearized form, meaning they are saved as a scalar value at a reference point and a corresponding gradient of the quantity. Such a linear representation has four independent parameters, therefore, the use of tetrahedra is ideal since the parameters of the linearized field quantity can be exactly defined via the field quantities at the four vertices of the tetrahedron. One might think that apart from tetrahedra other spatial objects with more vertices can still be used by fitting a linear function of the field quantity, however, such an application would destroy the important property of the field quantities being continuous through adjacent faces of tetrahedra. Furthermore, for vector quantities each vector component is independently linearized.

2. All the tetrahedra edges must coincide with the edges of the neighboring tetrahedra. Edges that lie on the faces of neighboring tetrahedra (these are termed *hanging nodes*) or crossings between edges are not permitted. This requirement is given by the continuity condition of linearized field quantities through the faces of each tetrahedron.

3. Each tetrahedron must be defined via four corner vertices in a given coordinate system. The coordinate values of each vertex are stored in an array and each vertex is identified by its array index. The index of the four vertices belonging to a specific tetrahedron has to be stored in a 4×1 array and can be accessed by indices 1 to 4 within each tetrahedron.

4. The tetrahedra are stored in an array of tetrahedron objects and identified by their index. Each tetrahedron has four defined faces labeled face 1 to 4. Each face i ($i = 1, 2, 3, 4$) is spanned by the vertices of the tetrahedron excluding tetrahedron-vertex i. For instance, face 3 will be spanned by tetrahedron-vertices 1,2,4. This implies that the tetrahedron-vertex i will be the only tetrahedron-vertex not lying on face i.

5. For a given tetrahedron, the neighboring tetrahedron which is separated by the i-th face of the current tetrahedron will be considered the neighbor i to

the current tetrahedron with its global labeling index being saved in the i-th position of a 4×1 array.

6. In addition to saving the four faces and neighbors of each tetrahedron, the index of the intersecting face between the original tetrahedron and its neighbor in the index system of the neighbor will be saved with the original tetrahedron. This means that the face through which a particle enters a neighboring tetrahedron can be determined by knowing through which face it is leaving the current one.

7. Tetrahedra at the outer boundary of the grid will not have neighboring tetrahedra at the boundary face, the neighboring tetrahedron index as well as the index of the face in the index system of the neighboring tetrahedron will be set to -1.

8. The normal vectors corresponding to each face of all tetrahedra must be explicitly calculated and saved together with a reference point. This enables the calculation of normal distances of any arbitrary point to all faces of each tetrahedron.

9. Grids that are made in a coordinate system with a periodic coordinate need to have an additional property set for tetrahedra faces lying at the periodic boundary, depending on which side the face lies. This determines in which direction the coordinate needs to be shifted when the particle passes through the boundary."[47] In the case that an approach with Cartesian coordinates is used for the particle position exchange between adjacent tetrahedral cells, see section 4.3.3, this requirement can be neglected.

10. In the case of axisymmetry of the electromagnetic field (with respect to the toroidal direction, e.g. field inside a tokamak), this symmetry property must not be violated when approximating field quantities with piecewise linear functions.

Generation of the vertices of the tetrahedral grid:

"The first step in implementing this grid is to generate the vertices that will define the corner points for the tetrahedral cells filling a given space. In order to achieve this, the domain which will be covered by the grid needs to be specified in the coordinate system where the grid will be generated."[47] In this case, we choose SFC as described above. In its most simple form, the grid will have vertices equidistantly spaced in s, ϑ and φ direction with intervals for s being $[0, 1]$, $[0, 2\pi]$ for ϑ and $[0, 2\pi/N_{\mathrm{fp}}]$ for φ, where N_{fp} is the number of field periods of the toroidal fusion device.

"Each coordinate x_i will now be discretized into N_i equidistant values for each given interval. Using nested loops, these discretized values will be connected to $N_s \times N_\vartheta \times N_\varphi$

unique triples representing the coordinates of the individual vertices of the grid."[47] As a result, one obtains the vertices of a regular grid consisting of hexahedra, if all vertices are connected to their nearest neighbors, assuming that all intervals are normalized to the same value. These hexahedral cells are subsequently split into tetrahedral cells as shown below.
The coarseness of the hexahedral grid, which also implies the total number of tetrahedral cells, is defined by N_i. Thus, we call $N_s \times N_\vartheta \times N_\varphi$ the "grid size" of the tetrahedral grid. However, in order to obtain the number of tetrahedral cells the product of this grid size must be multiplied by the number of tetrahedral cells per hexahedron.

Generation of tetrahedral cells and linearized electromagnetic field quantities:

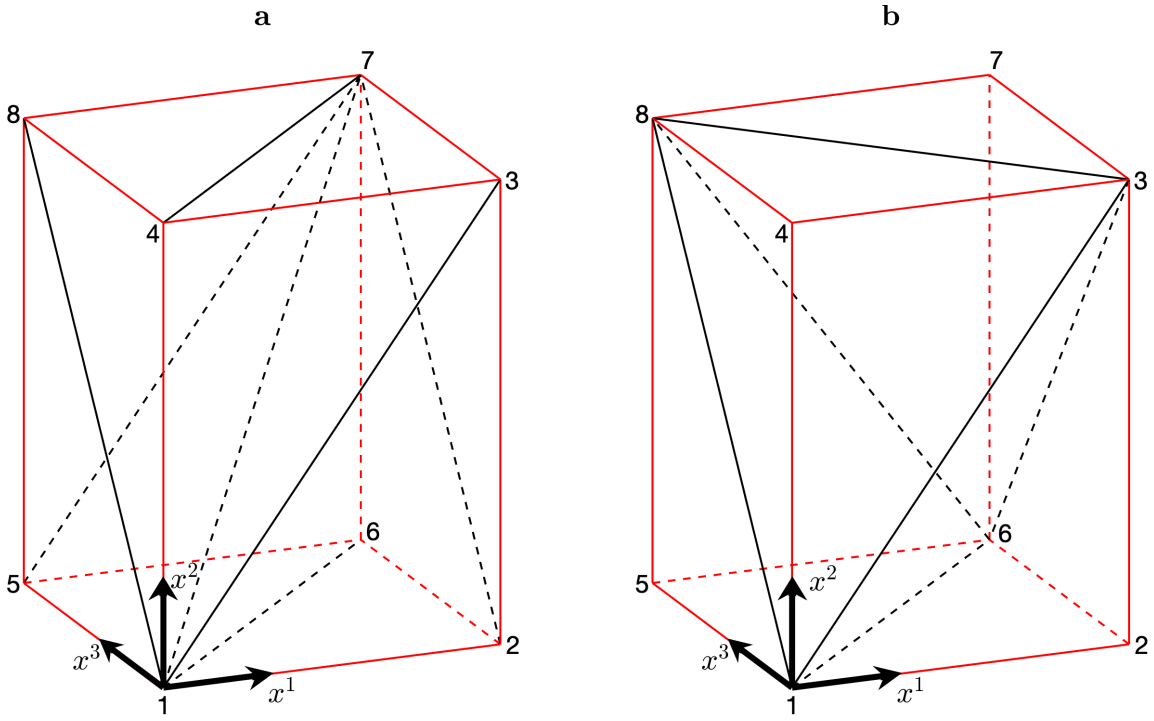

Figure 4.2: Flux coordinate space illustration of two realizations of a hexahedron (red) being split into tetrahedral cells. The symmetry direction is along the coordinate x^3.
(a) The hexahedron is split into six tetrahedral cells such that the axisymmetry property of the electromagnetic field is preserved upon linearization of field quantities.
(b) The hexahedron is split into five tetrahedral cells without preserving axisymmetry upon linearization. *Figure (a) is taken from Ref. [3]*

As stated in section 3.1, the field quantities A_k, B_k/ω_c, ω_c and Φ are independently approximated by continuous piecewise linear functions. This approximation is achieved by linearly interpolating the field quantities inside the tetrahedral cells of the described grid. Here, it should be mentioned that the exact field values are retained on the

tetrahedral cell's vertices.
In the case that the electromagnetic field is axisymmetric with respect to the toroidal direction, this symmetry property must not be violated by the interpolation. Thus, the tetrahedral cells must be specially oriented in order to preserve this symmetry and its respective invariant of motion of the guiding-center equations; see section 5.1. "Namely, this is achieved with stackable hexahedra each consisting of two triangular prisms, which are both subsequently split into 3 tetrahedral cells. The prisms are oriented such that all triangular prism faces lie on $x^3 = \text{const.}$ planes where x^3 is the symmetry direction. Thus, each tetrahedron face which lies on a $x^3 = \text{const.}$ plane is congruent with all other tetrahedron faces opposing it in the symmetry direction."[1] This specific splitting realization can be seen in a real space illustration of the field-aligned grid in Fig. 3.1. Furthermore, Fig. 4.2 (a) shows this specific splitting realization which is used in GORILLA in flux coordinate space. For a better understanding of the preservation of axisymmetry, let us assume that the modulus of the magnetic field B, which is proportional to the cyclotron frequency, $\omega_c = eB/mc$, is exactly given on the 8 vertices of the hexahedron in Fig. 4.2 (a). In an axisymmetric field, the respective values at vertices which oppose in the symmetry direction x^3 remain constant, e.g. B has the same value at vertex 1 and 5. Due to the linearization, the value of B at a random position (x_a^1, x_a^2, x_a^3) on the triangular tetrahedron face confined by the vertices $1, 2$ and 3 is a linear superposition of the values of B at these vertices. If we keep x_a^1, x_a^2 constant and make a variation of the component x_a^3, the point describing the position will cross the tetrahedral cell boundary and eventually reach the triangular tetrahedron face confined by the vertices $5, 6$ and 7. Throughout this variation, the value of $B(x_a^1, x_a^2, x_a^3)$ remains constant, since the values at the vertices opposing in the symmetry direction remain constant, while the vertices themselves, which are the basis of the superposition, change when entering an adjacent tetrahedron.
In contrast, Fig. 4.2 (b) shows an alternative splitting realization, where the field property of axisymmetry is not preserved. The triangular tetrahedron face confined by the vertices $1, 2$ and 3 is not opposed by a congruent triangular face in the symmetry direction. E.g., the value of B at a random position (x_b^1, x_b^2, x_b^3) on the triangular face with the vertices $1, 2$ and 3 is in general affected by the value at vertex 3, while the non-congruent opposing triangular face with the vertices $5, 6$ and 8 has the same B values at vertices 5 and 6 but not at vertex 8. If we make the same variation as above, specifically by keeping x_b^1, x_b^2 constant and change only x_b^3, the value of $B(x_b^1, x_b^2, x_b^3)$ will in general change throughout this variation.
Moreover, if one carefully observes the splitting realization of Fig. 4.2 (a), one can see that symmetry is not only preserved in the x^3 direction, but that the criterion for

conservation (upon linearization) is also valid for the directions x^1, x^2 in the case that these directions possess a symmetry property. In particular, in the case of transport optimized stellarators[55] a grid with this specific splitting realization should preserve the quasi-axisymmetry (B is independent of the toroidal angle φ) and quasi-poloidal symmetry (B is independent of the poloidal angle ϑ).
Finally, for the utilization in GORILLA a field-aligned grid has an inherent benefit with respect to the linear interpolation of the covariant component of the vector potential A_k when this is represented in flux coordinates. With the appropriate gauge transformation, A_k can be denoted in SFC as $A_s = 0$, $A_\vartheta = s\psi_{\text{tor}}^{\text{max}}$ and $A_\varphi = \psi_{\text{pol}}$, where s, $\psi_{\text{tor}}^{\text{max}}$ and ψ_{pol} are the normalized toroidal flux surface label, the toroidal flux at the separatrix and the poloidal flux, respectively. In the described linearization approach, for vector quantities each vector component is independently linearized. Thus, with this representation of A_k in SFC, no interpolation errors in the poloidal and toroidal direction are introduced, since none of the covariant components of A_k are dependent on ϑ or φ. As a consequence, the existence of flux surfaces is not violated in piecewise linear toroidal fields. However, due to the s-dependence of ψ_{pol}, the rotational transform, $\iota = \psi'_{\text{pol}}/\psi'_{\text{tor}}$, is a piecewise constant function in this approach. Associated consequences with respect to artificial chaotization are discussed in detail in section 5.2.

Numerical implementation:
In the numerical implementation of GORILLA, the *tetrahedral grid module* loads the magnetic field data and generates the field-aligned tetrahedral grid with the (symmetry preserving) splitting procedure of a regular hexahedral grid in flux coordinate space. Further, the linear interpolation of the electromagnetic field quantities is performed within the *tetrahedral physics module* as well as the construction of normal vectors that describe planes which confine these tetrahedra and which are essentially needed in the subroutines for appropriately pushing particles through the tetrahedral cells; see below.
The technical details of the remaining implementation for a logic that fulfils the requirements from above, specifically the indexing of tetrahedra, their faces and the indexing of the respective neighboring tetrahedra and their adjacent faces, etc., are omitted in this work, since this chapter is intended to present the code in a concise and unified manner. A detailed description can be found in Ref. [47].

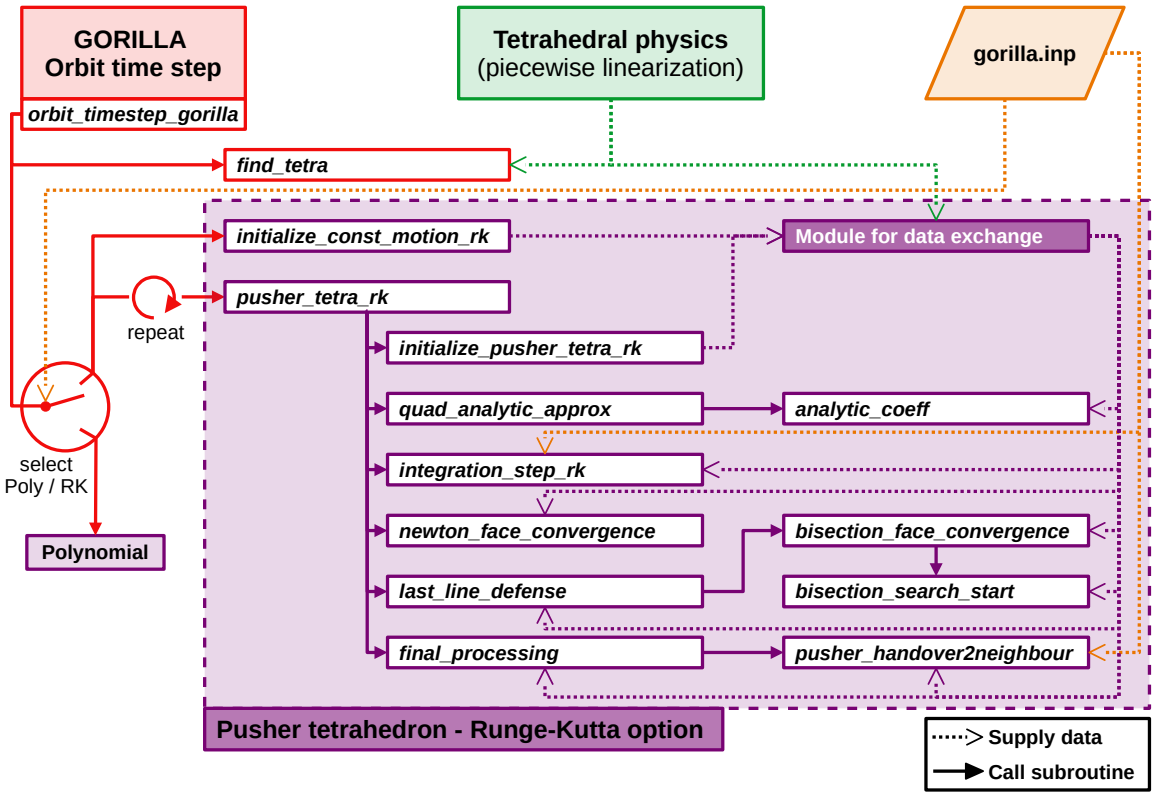

Figure 4.3: Overview of the structure of the *Runge-Kutta option* belonging to the *pusher tetrahedron module* of GORILLA. Dotted lines with line-arrows indicate the supply of data for the subsequent execution of subroutines. Solid lines with filled arrows show the direction of the subroutine calls. (Data flow of called subroutine return values is implied in the opposite direction.) *This figure is taken from Ref. [47] and modified and extended.*

4.3 Algorithm for pushing particles through tetrahedral cells

In this section an overview of the algorithm to integrate a guiding-center orbit by consecutively pushing a particle through tetrahedral cells (with its two options *Runge-Kutta* and *Polynomial*) is presented. As can be seen in Fig. 4.1, this is done within the *pusher tetrahedron module*. The working principle of the required routines and auxiliary methods is briefly explained. Without loss of generality, here, we limit to only one orbit application, namely *GORILLA orbit time step*. The second orbit application (*GORILLA Plot*), which is intended for the analysis of the quasi-geometric integration method itself, works in the same manner with some additional diagnostic extensions intended for visualization.

"The focus of the *pusher tetrahedron module* lies, however, not only on the computation of the trajectory and the calculation of the next tetrahedron intersection but rather on finding a numerically inexpensive scheme that allows to save computational cost while

reliably yielding accurate results for the intersection of the orbit with the tetrahedron. While the results are in theory equivalent for both pushing options, the approaches are completely independent and thus may vary in both computational efficiency and numerical accuracy, depending on the polynomial order K of Eq. (3.11) up to which the orbit is computed."[47]

Due to the interchangeability of these two options, we will now demonstrate the common algorithmic approach by referencing only the *Runge-Kutta option*. The distinctive details of these two options are subsequently discussed in separate sections. Fig. 4.3 shows the overview of the structure of the *Runge-Kutta option*, belonging to the *pusher tetrahedron module* of GORILLA, and also how this module is called from the orbit application *GORILLA orbit time step*. The actual *Runge-Kutta option* is depicted in a violet box.

A prerequisite for the functionality of this module is the pre-computed linearized electromagnetic field, as well as the normal vectors describing the planes which confine the tetrahedra. Both are provided by the *tetrahedral physics module* (green in Fig. 4.3). All the necessary settings including the pushing option (*Runge-Kutta* or *Polynomial*) are governed by the input file (orange in Fig. 4.3).

The *GORILLA orbit time step* application is realized as a wrapping routine with the name `orbit_timestep_gorilla` (red in Fig. 4.3) which is given the initial conditions of the particle and returns the condition after the pre-defined time step.

"For starting a particle at a given position without knowing to which tetrahedron it belongs to in coordinate space, an auxiliary routine `find_tetra` was constructed for efficiently finding the corresponding tetrahedron index to start a calculation"[47]; see section 4.3.3. This routine must be called only at the very first time when a particle is started. Further subsequent orbit time steps, e.g. after a scattering process in a simulation, do not require to find the appropriate tetrahedron in the grid anymore, as long as the particle position stays unchanged between consecutive time steps, and thus, the information to which tetrahedron the particle belongs can be retained.

The first subroutine to be called for an orbit time step within the *pusher tetrahedron module* (and after the appropriate tetrahedron has been found) is `initialize_const_motion_rk` which sets the constants of motion (perpendicular adiabatic invariant $J_\perp$ and respective squared value) for the given initial conditions. "The constants of motion will retain their set values for a number of tetrahedral pushings until the next time step is executed. Usually, between time steps collision events will occur. As a consequence, the constants of motion may change and have to be defined anew for every orbit time step."[47]

Eventually, the subroutine `pusher_tetra_rk` which is responsible for the actual push-

ing through a tetrahedral cell is called repeatedly inside the wrapping routine until the pre-defined time is elapsed. Since the pre-defined time step for an orbit will, in general, lead to a particle position somewhere inside a final tetrahedron, the remaining time of the orbit to reach the pre-defined time must be given to the subroutine `pusher_tetra_rk` when being called repeatedly. "This subroutine then computes the time it takes until the particle exits the current tetrahedron and compares this value to the remaining time of the orbit time step. In the case that the time it takes to leave the tetrahedron is less than the remaining time of the orbit time step, the pushing is computed. The remaining time is then reduced by this value and the next pushing through the adjacent tetrahedron is started. In the case there is not sufficient time to complete the pushing, the orbit is instead integrated up to the value of the orbit parameter τ corresponding to the remaining time, leading to a final position somewhere inside the tetrahedron."[47]
The strategy how to achieve a pushing through a tetrahedral cell with the *Runge-Kutta* option (within the subroutine `pusher_tetra_rk`) and, specifically, in a computationally efficient manner is discussed in the following. The algorithm for the *Polynomial* option is identical until here, with the exception of calling the subroutines `initialize_const_ motion_poly` and `pusher_tetra_poly` instead. The latter subroutine realizing the alternative polynomial approach is discussed below as well.

4.3.1 Numerical approach with Runge-Kutta

The subroutine `pusher_tetra_rk` is called with the initial conditions, specifically guiding-center position and parallel and perpendicular velocity, as well as the required tetrahedron information, specifically the indices of the appropriate tetrahedron and tetrahedron face. The face index takes the values $1 \ldots 4$, meaning that the particle's position lies in on one of the four tetrahedra faces, or 0 in the case the particle's position is somewhere inside the tetrahedron. In addition, the remaining time of the orbit time step prior the pushing is also given to `pusher_tetra_rk`. In the violet box in Fig. 4.3 the subroutines used within the *Runge-Kutta option* for pushing a particle through a tetrahedron can be seen. After the execution, the subroutine returns the new initial conditions for a subsequent pushing through a tetrahedron, as well as the remaining time of the orbit time step after the pushing.

Initialize the module for data exchange:
"In the `pusher_tetra_rk` subroutine, first an initialization subroutine `initialize_pusher_tetra_rk` is called. Here, the initial conditions are used to compute the coefficients a_l^i, b^i for the ordinary differential equation set (3.5), representing the equations

of motion in the specific tetrahedron. It should be emphasized here that this ODE set is solved within a shifted coordinate system, where the coordinate origin lies on the first vertex of the given tetrahedron."[47]

Initial guess of exit time and exit face:
Now, the intersection between the orbit and the tetrahedral cell, which can be reached in the smallest positive time from the initial position ("exit time") and its appropriate tetrahedral face ("exit face"), needs to be computed. (A representative illustration can be seen in Fig. 3.2.) "Since this must be done efficiently, a numerically inexpensive approximative quadratic solution is first evaluated by subroutine `quad_analytic_approx` to compute the guess for the orbit parameter τ at the first intersection of the particle trajectory with the cell boundary. Based on the approximative (quadratic) result for the orbit parameter τ, an integration step is performed for the given step length"[47] using `integration_step_rk`. This subroutine solves the initial value problem for the set (3.5) either, determined by the user, with a single step of the Runge-Kutta 4 method (RK4) or with the adaptive Runge-Kutta 4/5 method (RK4/5) and returns the evaluated position.
"In general, due to inaccuracies in the approximation, this value does not correspond to a "converged" orbit position. In the context of the subroutine `pusher_tetra_rk`, converged simply means that the particle position is within a defined convergence distance to a given tetrahedral plane. This distance can be set manually. Its default value is 10^{-10} times the normal distance of the first vertex within a given tetrahedron to its opposing cell boundary spanned by vertices 2,3 and 4 of the given tetrahedron. In addition, the normal velocity, which can also be computed from the output of the RK integration step, must have a negative sign in order for the convergence to be valid."[47] (By definition, the normal vectors describing the planes which confine the tetrahedra are pointing inwards.) "The negative sign assures here, that only outflowing particles (i.e. with negative normal velocity) are accepted as a solution. Since the orbit position is generally not yet converged after the quadratic approximation, the next step is to apply Newton's method for the convergence of the orbit position on the tetrahedron face by calling the subroutine `newton_face_convergence`."[47] Runge-Kutta integration steps are there embedded into an iterative scheme using Newton's method in various orders. A detailed description of this approach is given in Ref. [44].

Convergence and validation loop (`conv_val_loop`):
"What has been obtained so far is a proposal for the exit plane and a converged orbit position on this plane. There may still be some problems, however, since for example

Newton's method can fail if the orbit in fact turns before it intersects with the plane. Furthermore, it might happen that it does converge on the suggested plane but at the point of convergence it had already left the tetrahedron through another plane which is not allowed as it would be the actually correct exit plane instead of the proposed one. Such cases need to be checked and handled appropriately. For this purpose, the *convergence and validation loop* (`conv_val_loop`) was implemented. This loop starts directly after the above mentioned quadratic approximation just before the convergence using Newton's method. Here, in each iteration of the loop, the algorithm tries to converge the particle orbit position on the currently proposed intersection face. Next, if convergence is reached the algorithm checks for the remaining planes if the particle lies outside the tetrahedron. If this is not the case, furthermore, the normal velocity is checked to see if the particle flies outside. In case this is also correct, the particle is considered "converged" and, hence, accepted.
In any case, where an error is detected, an appropriate approach is suggested. In most cases this involves using the quadratic approximation to suggest a different face, however, in some special cases this is not sufficient. Therefore, one calls inside `conv_val_loop` an additional convergence routine `last_line_defense`, which no longer opts for high computational efficiency but rather for a reliable way of finding the intersection point.
This subroutine is very comprehensive, but a central piece of it is a bisection scheme. For a short overview, in this scheme (`bisection_face_convergence`) one computes the relative particle positions to all four faces and furthermore checks the normal velocities. Here, if the particle is inside the tetrahedron, the current step length is doubled and an integration step is performed. If the particle is now outside the tetrahedron, the last integration step is halved and integrated back in negative τ-direction. This is done in an iterative scheme until a converged particle position has been found."[47] In extremely rare cases (less than one in 100 million pushings) an unsuitable starting position for the bisection procedure can lead to failure with respect to face convergence. In such a case, the orbit is stepwise integrated in the subroutine `bisection_search_start` in order to find an appropriate starting position for the consecutive bisection procedure, leading to face convergence.
"Albeit the subroutine `last_line_defense` is computationally expensive, it remains an indispensable element of the algorithm due to its high reliability. This effect on performance remains small, however, as only a small portion of particle pushings actually need to be solved by `last_line_defense`.
Due to the used approach of guessing the exit face with an approximation instead of computing all intersections with all four planes, which would be computationally

much more expensive, many special cases of particle trajectories had to be taken into account in order that the logic can deal with them correctly."[47] It should be mentioned that much thought has gone into developing such an efficient and reliable logic.

Final processing:
Eventually, the last steps of the orbit integration are to first check if the computed time is in fact smaller than the remaining time of the orbit time step. When this is true, the remaining time of the orbit time step which was given to `pusher_tetra_rk` is reduced by the exit time computed in this pushing. By calling the subroutine `pusher_handover2neighbour`, the tetrahedron and face index of the tetrahedron which is adjacent to the exit face will be evaluated. Due to periodic boundaries of adjacent tetrahedra, particle positions might be updated in `pusher_handover2neighbour` which is explained in section 4.3.3. Finally, the orbit condition (position and velocities) and the tetrahedron information for the subsequent pushing are returned to the orbit application, i.e. `orbit_timestep_gorilla`.
If the computed time is in fact greater than the remaining time of the orbit time step, the corresponding position at the remaining time is evaluated instead of the converged orbit position. In this case the current tetrahedron index will be returned while the face index is manually set to 0, meaning that the particles is somewhere inside the tetrahedron.

4.3.2 Analytical approach with polynomials

The subroutine `pusher_tetra_poly` is called in the same manner as its counterpart `pusher_tetra_rk`, specifically with the initial conditions of the particle, as well as the required tetrahedron information and the remaining time of the orbit time step. After the execution, the subroutine returns the new initial conditions for a subsequent pushing through an (adjacent) tetrahedron, as well as the remaining time of the orbit time step after the pushing. In the following, an overview of the working principle of the analytical approach with polynomials, which differs significantly from the numerical approach with Runge-Kutta and Newton's method, is given.
As already stated in section 3.2, by truncating the summation of the formal solution of the locally linear guiding-center equations (3.11) at $K = 2$, 3 or 4, one obtains approximate solutions of various orders in Larmor radius. With these solutions, the equations for the exit time, which is the time it takes for the particle to reach the tetrahedral cell boundary, are algebraic equations. Thus, we call this approach

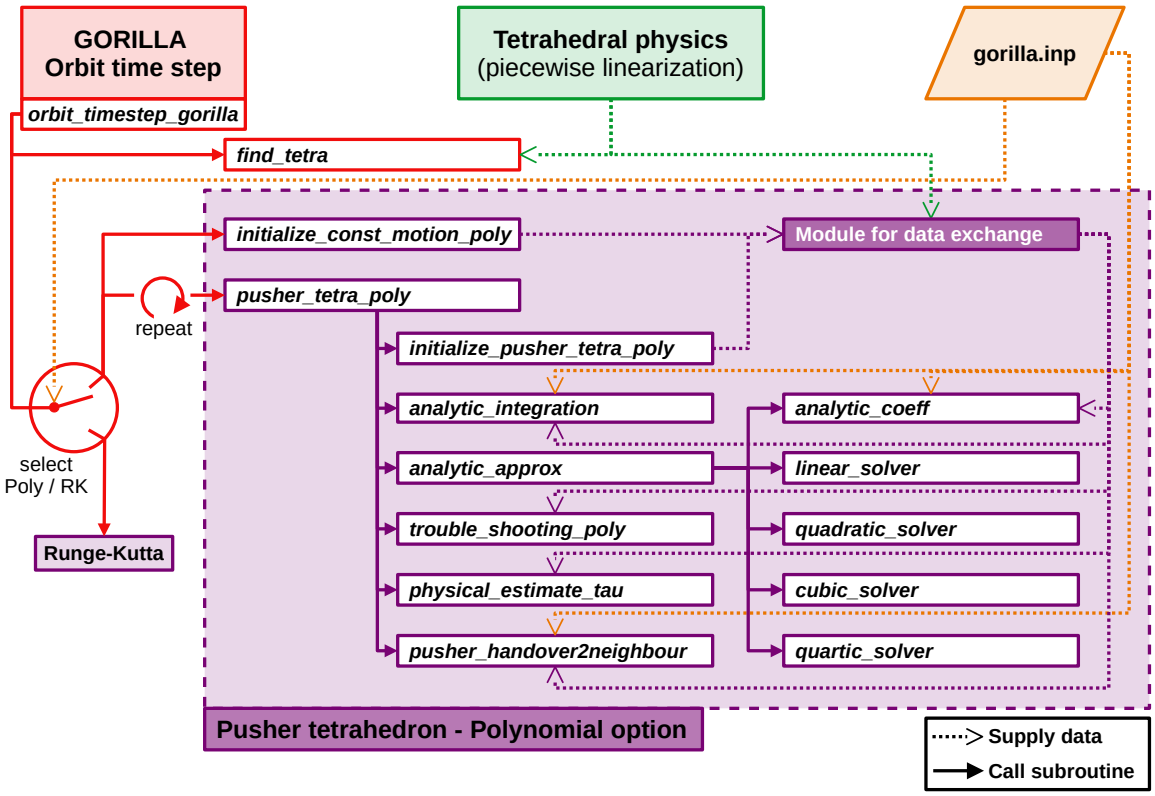

Figure 4.4: Overview of the structure of the *Polynomial option* belonging to the *pusher tetrahedron module* of GORILLA. Dotted lines with line-arrows indicate the supply of data for the subsequent execution of subroutines. Solid lines with filled arrows show the direction of the subroutine calls. (Data flow of called subroutine return values is implied in the opposite direction.)

"analytical".

In general, the orbit can be computed separately for all orders $K = 2$, 3 or 4, however, since computational efficiency is also aspired to in this approach, the quadratic polynomial solution $K = 2$ is used in a manner similar to the RK-approach, for predicting the exit face, in the event that the orbit is computed with the orders $K = 3$ or 4. The predominant effect of this is the reduction of the number of higher order root finding operations from four to one, while demanding an algorithm that guarantees that the "correct" face is found. In the context of the subroutine `pusher_tetra_poly`, correct simply means that the particle position is within a negligible convergence distance to the plane, which requires an accurate polynomial root finding procedure, and that the particle flies outwards the tetrahedron with respect to the exit plane.

In the violet box in Fig. 4.4 one can see the subroutines used within this *Polynomial option*. Similarly as in the *Runge-Kutta option*, by calling the initialization subroutine `initialize_pusher_tetra_poly` the initial conditions for orbit integration within the specific tetrahedron are preprocessed, e.g. local coordinates, while the coordinate

handling for subsequent pushing in an adjacent tetrahedron is performed within `pusher_handover2neighbour` at the end of the pushing procedure.
In the context of the actual orbit integration let us recall, as stated in Eq. 3.14, that for computing the appropriate exit time, we need to find the smallest positive root of

$$\mathbf{n}^\alpha \cdot \left(\mathbf{z}_0 + \sum_{k=1}^{K} \frac{\tau_e^k}{k!} \left(\hat{\mathbf{a}}^{k-1} \cdot \mathbf{b} + \hat{\mathbf{a}}^k \cdot \mathbf{z}_0 \right) - \mathbf{z}^\alpha \right) = 0 \quad \text{with} \quad K = 2, 3, 4, \tag{4.1}$$

where $\mathbf{n}^\alpha = \left(n_1^{(\alpha)}, n_2^{(\alpha)}, n_3^{(\alpha)}, 0 \right)$ are face normals and $\mathbf{z}^\alpha = \left(x_{(\alpha)}^1, x_{(\alpha)}^2, x_{(\alpha)}^3, 0 \right)$. Depending on the aspired polynomial order K, the coefficients of this polynomial are then computed in the subroutine `analytic_coeff` for the initial condition $\mathbf{z}_0$. Although the roots of such a polynomial can be exactly solved in an analytical sense, quite some difficulties arise when implementing such an analytical root solving procedure due to ill-conditioned coefficients that lead to numerical cancellations. Thus, the algorithm of Ref. [56] was applied for accurately and efficiently obtaining all roots of general real quadratic, cubic and quartic polynomials. The respective subroutines are `linear_solver`, `quadratic_solver`, `cubic_solver` and `quartic_solver`.
However, such a pure root solver is not sufficient in our case due to three major reasons. First, if a particle orbit is started from one of the four faces, which is predominantly the case, one is not interested in this intersection of the orbit with the tetrahedral face, specifically the actual starting position at time zero. Since this root is already known, the order of the polynomial is reduced by one. Second, the coefficients of the polynomial can turn zero (or almost zero in a numerical sense) due to scattering events of the particles or by definition of the user, e.g. if the perpendicular velocity of the particle is set to zero by the user. Also in this case the order of the polynomial is reduced. The third case is rather technical and deals with ill-conditioned coefficients. The root solving algorithm of Ref. [56] expects the coefficient of the highest order polynomial term to be normalized to one, but such a normalization can lead to severe numerical cancellations since the orders of magnitude of the individual coefficients differ quite drastically. This issue is solved by rescaling the polynomial, then finding the roots of this rescaled polynomial and finally scaling back the appropriate smallest positive root. E.g., let us assume we have a fourth order polynomial for the orbit parameter τ. We introduce the scaling parameter λ by multiplying every polynomial term with one and subsequently perform the transformation $\tilde{\tau} = \lambda\tau$,

$$\frac{a}{\lambda^4} \underbrace{\lambda^4 \tau^4}_{\tilde{\tau}^4} + \frac{b}{\lambda^3} \underbrace{\lambda^3 \tau^3}_{\tilde{\tau}^3} + \frac{c}{\lambda^2} \underbrace{\lambda^2 \tau^2}_{\tilde{\tau}^2} + \frac{d}{\lambda} \underbrace{\lambda\tau}_{\tilde{\tau}} + e = 0. \tag{4.2}$$

If we demand that the coefficient of the highest order polynomial term is normalized

to one,

$$\tilde{\tau}^4 + \frac{b}{a}\lambda\tilde{\tau}^3 + \frac{c}{a}\lambda^2\tilde{\tau}^2 + \frac{d}{a}\lambda^3\tilde{\tau} + \frac{e}{a}\lambda^4 = 0, \tag{4.3}$$

then the scaling parameter λ gives us the freedom to rebalance the ill-conditioned coefficients by choosing that, e.g., $\frac{e}{a}\lambda^4 \stackrel{!}{=} \frac{d}{a}\lambda^3$. For the sake of completeness, rescaling of the polynomial can also be performed without the required normalization, in the case a different root solving procedure is applied.

The subroutine `analytic_approx` takes care of the appropriate reduction of the order of the polynomial, wherever necessary, and also scales the polynomial before and after the root finding procedure. Since there are infinite choices for how to set the scaling parameter λ, the subroutine `trouble_shooting_poly` systematically varies this choice with the result that the root finding procedure does not suffer from numerical cancellations. It should be mentioned, that this subroutine is very seldom needed, since in the vast majority of the root finding operations one choice of λ is sufficient. Moreover, without going into the details, more cases arise for particle pushing with the polynomial approach that need to be handled within the subroutine `pusher_tetra_poly`. E.g., let us assume that the position of a trapped particle's banana-tip is exactly at the face of a tetrahedral cell. The correct root ("exit position") has been found, but instead of entering the adjacent tetrahedron, the particle actually enters again at the very same tetrahedron after turning exactly at the boundary.

For appropriately dealing with all the cases, an essential tool is to estimate the dwell time with some physical assumptions in order to obtain a maximum value for the exit time, e.g. 10 times the estimated dwell time. This is realized within the subroutine `physical_estimate_tau`.

When implementing the *Polynomial option*, however, the number of special cases that need to be treated is much smaller than in the case of the *Runge-Kutta option*, in which an appropriate root is found in a numerical iterative scheme with Newton's method and bisection. This is due to the analytic approach in which the smallest real positive root can be found in an (almost) straight forward form up to the order $K = 4$.

4.3.3 Auxiliary methods

The subroutine `find_tetra`:

"Since a user generally defines a particle starting position in global coordinates, rather than specifying a tetrahedron index and a local position, the search routine `find_tetra` is implemented for finding the corresponding tetrahedron index. Moreover, a function `is_inside` is implemented for this purpose, which allows the checking

of whether the particle position lies inside a proposed tetrahedron. This function uses the Hesse normal form to compute the distances to all tetrahedral planes. Due to the axisymmetry of the grid, based on the current φ position one can vastly reduce the number of possible tetrahedra by allowing only tetrahedra of the current φ-slice. Now, a loop over all possible tetrahedron indices is implemented to check if the particle lies inside."[47]

Once a tetrahedron fulfilling this criterion has been found, the distances to the four planes need to be checked for convergence on a plane. If this is the case for one or more planes, the normal velocities with respect to these planes where the orbit position is converged must be evaluated. When a particle pushing through a tetrahedral cell is initiated, both pushing routines (*Runge-Kutta* or *Polynomial*) assume that a given particle flies inwards. If a particle is started on the face of a tetrahedron but, indeed, immediately flies outwards, this can lead to errors in the pushing logic.

Thus, in `find_tetra` a tetrahedron is found, where the particle position is located inside the tetrahedron and the normal velocities with respect to the planes where the orbit position is converged has a positive sign, meaning that the particle is flying inwards the tetrahedral cell. This circumstance and the implied importance of the algorithm can easily be made clear, if one imagines a particle starting exactly on the vertex of a tetrahedron. Such a vertex position is shared by multiple neighbouring tetrahedra, whereby only one specific tetrahedron among these can be taken into account to initiate particle pushing.

It should be mentioned, that generally in a simulation the particle positions for starting orbits are uniformly distributed in the respective coordinate space and, therefore, the probability that such a position is converged on a face or even on a vertex is negligibly small. In the majority of these cases only the criterion of whether the particle lies inside the tetrahedron, which is computationally much less expensive, is thus evaluated.

The subroutine `pusher_handover2neighbour`:

After the orbit position is converged on the exit face of a tetrahedron, the indices of the adjacent tetrahedron and the respective "entry face" (belonging to the adjacent tetrahedron; congruent and adjacent to the exit face) are needed for a consecutive pushing. The subroutine `pusher_handover2neighbour` provides this information and furthermore, in the event that the intersection between the orbit and the face is at a periodic boundary of the coordinate system, also takes care of appropriate handling of the particle position with respect to such a periodic boundary.

For this purpose, two options are implemented. First, the period length is either added to or subtracted from the respective coordinate component depending on the direction

from which the periodic boundary is crossed. E.g., in the case of symmetry flux coordinates[54] the period length for poloidal component ϑ is 2π, and for the toroidal component φ it is $2\pi/N_{\mathrm{fp}}$, where N_{fp} is the number of field periods of the toroidal fusion device. A prerequisite for this is, however, that every tetrahedral cell also stores the information regarding its faces if they are adjacent to any periodic boundary and if this is the case, on which side of the periodic boundary they are located. Let us now assume that we are dealing with a coordinate with a periodic boundary at $0 = 2\pi$. Integers for each face and every periodic coordinate can take the values 0, 1 and -1, meaning, respectively, that the face is not on a periodic boundary, or that the face is either on the periodic boundary 2π or 0.

The second implementation manages the handling of the position with respect to the adjacent tetrahedron without cell-stored information of this kind and also omits addition and subtraction of the period length to the periodic component. Instead of this, the particle position is exchanged via Cartesian coordinates. The only prerequisite here is that the periodic boundary coincides with the tetrahedral cell boundary and does not lie inside a tetrahedral cell. We denote the Cartesian exchange coordinates with (x, y, z) and the "internal" coordinates used for the pushing with (s, ϑ, φ); allowing these to be symmetry flux coordinates[54] (SFC) in this example without loss of generality. For every vertex of a tetrahedron the coordinates are stored in two sets, both SFC and Cartesian coordinates. Specifically, the tetrahedron vertices $1 \ldots 4$ take the Cartesian coordinates $(x_1, y_1, z_1), \ldots, (x_4, y_4, z_4)$ and, respectively, the SFC $(s_1, \vartheta_1, \varphi_1), \ldots, (s_4, \vartheta_4, \varphi_4)$.

Now, let us express in the internal SFC the particle position (s, ϑ, φ), which is converged on the exit face, as a linear combination of vectors which span the tetrahedron. In this case, let us assume that the vertex $(s_4, \vartheta_4, \varphi_4)$ is contained in the exit plane. For this purpose, we need to solve for the coefficients ξ_i the linear equation set,

$$\begin{aligned} \xi_1(s_1 - s_4) + \xi_2(s_2 - s_4) + \xi_3(s_3 - s_4) &= s - s_4, \\ \xi_1(\vartheta_1 - \vartheta_4) + \xi_2(\vartheta_2 - \vartheta_4) + \xi_3(\vartheta_3 - \vartheta_4) &= \vartheta - \vartheta_4, \\ \xi_1(\varphi_1 - \varphi_4) + \xi_2(\varphi_2 - \varphi_4) + \xi_3(\varphi_3 - \varphi_4) &= \varphi - \varphi_4, \end{aligned} \tag{4.4}$$

which can also be denoted by $\hat{\mathbf{A}}_{\mathrm{SFC}} \cdot \xi = \mathbf{b}_{\mathrm{SFC}}$, where $\hat{\mathbf{A}}_{\mathrm{SFC}}$ is a matrix containing the vectors which span the tetrahedron in SFC and $\mathbf{b}_{\mathrm{SFC}}$ is the right hand side of Eqs. (4.4). We use the coefficients ξ_i to approximately express the particle coordinate (s, ϑ, φ) in Cartesian coordinates (x, y, z) by simply making a linear combination of vectors which span the tetrahedron in Cartesian coordinates, but which are identically

constructed from the same vertices as their counterparts in SFC,

$$\begin{aligned} x &= x_4 + \xi_1(x_1 - x_4) + \xi_2(x_2 - x_4) + \xi_3(x_3 - x_4), \\ y &= y_4 + \xi_1(y_1 - y_4) + \xi_2(y_2 - y_4) + \xi_3(y_3 - y_4), \\ z &= z_4 + \xi_1(z_1 - z_4) + \xi_2(z_2 - z_4) + \xi_3(z_3 - z_4). \end{aligned} \tag{4.5}$$

Here, a matrix $\hat{\mathbf{A}}_{\text{Cart.}}$ can be defined which contains the vectors spanning the tetrahedron in Cartesian coordinates. The particle position in Cartesian coordinates can thus be computed by $(x, y, z)^T = (x_4, y_4, z_4)^T + \hat{\mathbf{A}}_{\text{Cart.}} \cdot (\hat{\mathbf{A}}_{\text{SFC}}^{-\mathbf{1}} \cdot \mathbf{b}_{\text{SFC}})$. It is clear that the transformation is only approximative and not exact, since only the local basis is in fact directly exchanged while the coefficients remain unchanged. This transformation, however, will subsequently also be performed in the opposite direction cancelling this transformation error. It should be mentioned that the matrices $\hat{\mathbf{A}}_{\text{Cart.}}$ and $\hat{\mathbf{A}}_{\text{SFC}}^{-\mathbf{1}}$ as well as their respective matrix multiplication $\hat{\mathbf{A}}_{\text{Cart.}} \cdot \hat{\mathbf{A}}_{\text{SFC}}^{-\mathbf{1}}$ can be precomputed for every tetrahedron face due to the associativity of matrix multiplications. Thus, the transformation is in reality only a multiplication of a matrix with a vector resulting in a total amount of 9 multiplications.
In order to obtain the internal (SFC-)position of the particle in the adjacent tetrahedron, which is needed for consecutive pushing, the Cartesian coordinates of the particle are transformed to SFC in a similar way as above, but with (x, y, z) and (s, ϑ, φ) being switched.

Part III

Applications, results and discussion

Chapter 5

Collisionless guiding-center orbits

The content of this chapter including the figures can also be found in section III of Ref. [1] formulated by the author and including some minor modifications.

5.1 Guiding-center orbits in an axisymmetric tokamak field

"In this section the results of quasi-geometric orbit integration computed with GORILLA and the comparison to an exact orbit computed with a usual adaptive RK4/5 integrator are presented for an axisymmetric tokamak field configuration of ASDEX Upgrade (shot 26884 at 4300 ms) described in Ref. [57]. The adaptive RK4/5 integrator requires high-order interpolation of electromagnetic fields with continuous derivatives, e.g. with help of 3D-splines, instead of continuous piecewise linear functions as in the case of the quasi-geometric integration method.
In axisymmetric configurations, the shape of the orbit is fully determined by three conservation laws, $p_\varphi = \text{const.}$, $J_\perp = \text{const.}$ and $w = \text{const.}$, where

$$p_\varphi = \frac{e}{c} A^*_\varphi = m v_\parallel \frac{B_\varphi}{B} + \frac{e}{c} A_\varphi \tag{5.1}$$

is the canonical toroidal angular momentum. The conservation of p_φ is obvious from Eq. (3.1) since after its substitution in $\dot{p}_\varphi = \dot{x}^i \partial p_\varphi / \partial x^i$ all the terms in the resulting expression are proportional to partial derivatives over φ of various A^*_k components.
In geometric/symplectic numerical integration schemes these conservation properties are retained[24], which means that orbits must remain closed in the poloidal projection. In order to preserve p_φ upon linearization, the tetrahedral cells of the method's underlying grid must be specially oriented with respect to the symmetry direction;"[1] see section 4.2. "Consequently, the canonical toroidal angular momentum remains invariant in the presented method, since the linearization of the electromagnetic field within a tetrahedral cell does not introduce an interpolation error in the symmetry

direction. Namely, partial derivatives of the field quantities with respect to the symmetry direction remain zero. In the following, cylindrical (R, φ, Z) and symmetry flux coordinates (s, ϑ, φ) of Ref. [54] are used, where $\varphi = x^3$ in both coordinate systems.

Fig. 5.1 depicts Poincaré plots ($\varphi = 0$) of trapped thermal ion orbits making 10^7 toroidal turns which are integrated by different methods from the same starting conditions. Quasi-geometric integration using an iterative scheme with RK4 integration and Newton steps has been performed in cylindrical and symmetry flux coordinates and is compared to the exact orbit. It can be seen, that the coarseness of the grid leads to slightly differently shaped orbits obtained in different coordinate systems, whereas the effect of the integration error (3.13) is negligible.

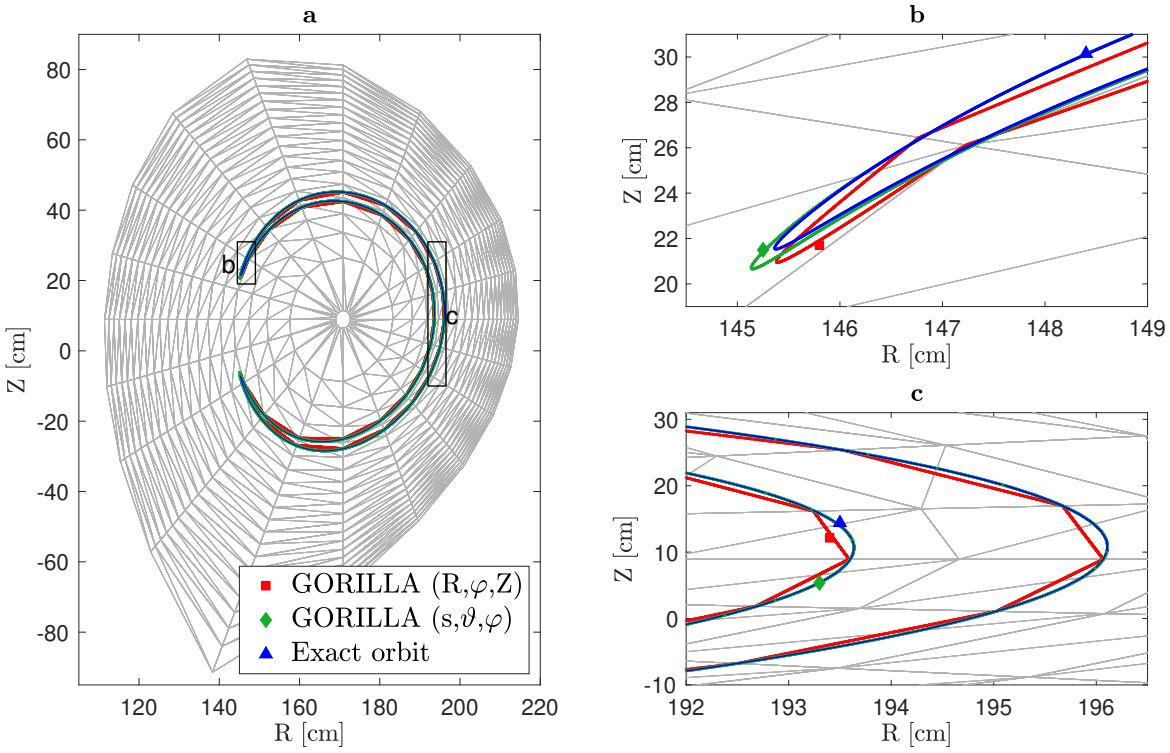

Figure 5.1: (a) Poincaré plot ($\varphi = 0$, 10^7 toroidal mappings) of a trapped 1.5 keV D ion in axisymmetric ASDEX Upgrade configuration with a tetrahedral grid size of 20x20x20. Two-dimensional Poincaré sections of orbits obtained with different integration methods are indicated with markers: Exact orbit (adaptive RK4/5): ▲, GORILLA with cylindrical coordinates: ■, GORILLA with symmetry flux coordinates: ♦. (b) and (c) are magnifications of the pertinent zones in (a).

In Fig. 5.2 results of the quasi-geometric integration of a passing high energy ion (300 keV) using the polynomial series solution (3.11) are shown for the same axisymmetric geometry and compared to the exact orbit. Symmetry flux coordinates are used for the quasi-geometric integration, the results are then converted to cylindrical

coordinates in the first plot (a). In general, the quartic polynomial solution ($K = 4$) of ODE set (3.5) is equivalent to the numerical solution using Runge-Kutta 4, thus, the result of the latter is omitted in the figure. Moreover, it can be seen that even for high energy ions the series expansions of third and fourth order are already accurate enough in order to fulfill the condition $p_\varphi = \text{const.}$ over 10^6 toroidal mappings. However, the second order series expansion shows a convective behavior, due to a systematic error from solving the ODE set (3.5). Nevertheless, the second order series expansion is sufficient for electrons which have much smaller Larmor radii.

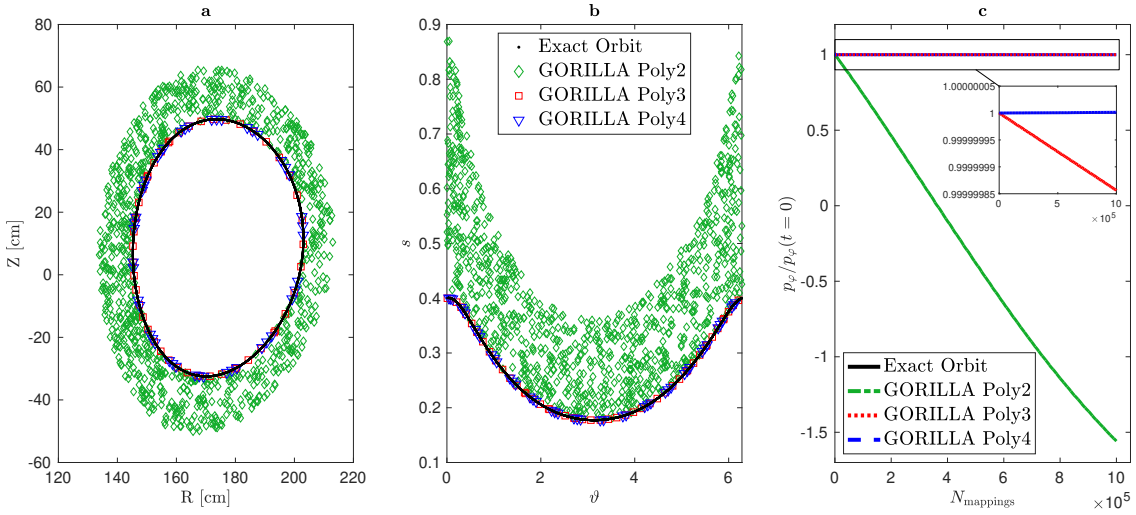

Figure 5.2: (a & b) Poincaré plot ($\varphi = 0$, 10^6 toroidal mappings) of a passing 300 keV D ion in axisymmetric ASDEX Upgrade configuration evaluated by GORILLA with the analytical solution in form of a polynomial series truncated at $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$) compared to the exact orbit ($\bullet$). Cylindrical coordinates are used for the Poincaré plot in (a), whereas symmetry flux coordinates are used in (b).
(c) Canonical toroidal angular momentum p_φ normalized to the value at $t = 0$ is evaluated at the Poincaré sections in (a): Exact result (solid) is compared to polynomial series truncated at $K = 2$ (dash-dotted), $K = 3$ (dotted) and $K = 4$ (dashed).

To obtain the orbits of thermal ions shown in Fig. 5.3, uniformly distributed axisymmetric random noise ($\xi = 0 \ldots 1$) is added to the electrostatic potential $\Phi^{\text{noisy}} = \Phi(1+\epsilon_\Phi \xi)$, to the vector potential $A_k^{\text{noisy}} = A_k(1 + \epsilon_A \xi)$ and simultaneously to both quantities, respectively. Here, ϵ is the relative magnitude of added noise. The guiding-center orbits are evaluated in symmetry flux coordinates by GORILLA with the analytical solution in form of a polynomial series truncated at $K = 4$, the results are then converted to cylindrical coordinates. Even though relatively high noise (up to 30 %) is added, the orbits keep a similar shape in comparison with the unperturbed ones and remain closed in the poloidal projection, meaning the condition $p_\varphi = \text{const.}$ is still fulfilled.
Therefore, the quasi-geometric integration method is suitable for self-consistent Monte

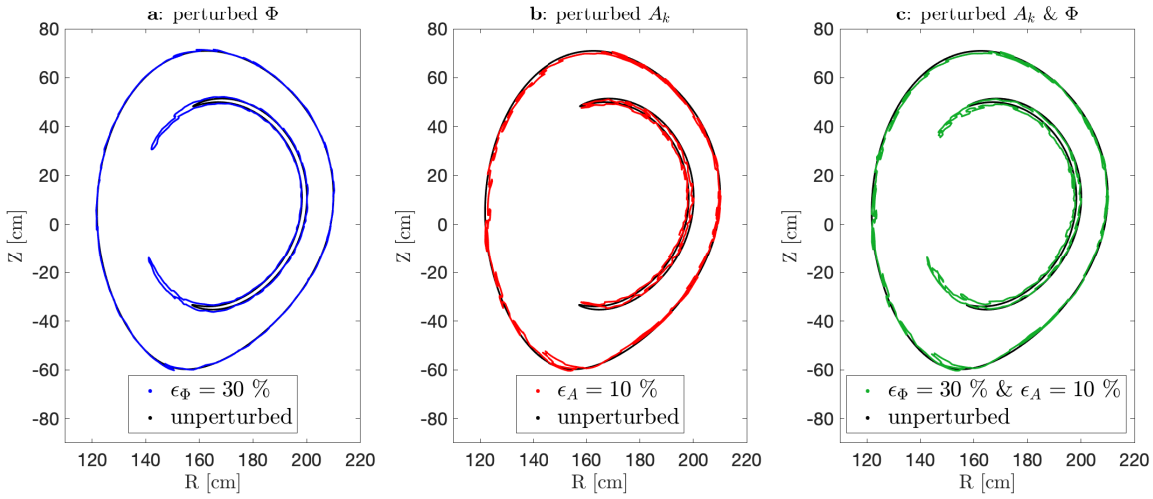

Figure 5.3: Poincaré plot ($\varphi = 0$) of trapped and passing 1.5 keV D ions in axisymmetric ASDEX Upgrade configuration with axisymmetric perturbation of electrostatic potential Φ and vector potential A_k. Orbits are evaluated by GORILLA with the analytical solution in form of a polynomial series truncated at $K = 4$.
The Poincaré plot for unperturbed electromagnetic fields is depicted in black as a reference. Uniformly distributed axisymmetric random noise ($\xi = 0 \ldots 1$) is added in (a) to the electrostatic potential $\Phi^{\text{noisy}} = \Phi(1 + \epsilon_\Phi \xi)$, in (b) to the vector potential $A_k^{\text{noisy}} = A_k(1 + \epsilon_A \xi)$ and in (c) to both quantities.

Carlo modeling of 2D equilibrium plasma parameters and electromagnetic fields computed from those parameters. Further, it should be noted that the computational efficiency of the quasi-geometric integrator is not affected by the presence of small-scale noise. However, this would be the case for an adaptive RK4/5 integration in combination with high order smooth interpolation where the noise leads to small-scale oscillations.

The Poincaré plots ($\varphi = 0$) of Fig. 5.4 (a) & (b) correspond to a trapped high energy ion (300 keV), where the orbits are integrated using the polynomial series expansion in several orders and compared to the exact orbit. Again, symmetry flux coordinates are used for the quasi-geometric integration, the results are then converted to cylindrical coordinates in the first plot (a). Despite the same starting conditions for all orbits, the magnification in (b) clearly depicts differences in the shape of the orbits caused by a finite grid size ($100 \times 100 \times 100$) and different orders of the series expansion.

Fig. 5.4 (c) shows the corresponding time evolution of the parallel adiabatic invariant $J_\parallel$, which is defined by an integral over the distance l passed along the field line during a single bounce period t_b by a trapped particle as follows,

$$J_\parallel = m \oint v_\parallel \mathrm{d}l = m \int_0^{t_b} v_\parallel^2 \left(t\right) \mathrm{d}t. \tag{5.2}$$

For an exact orbit, $J_\parallel$ is a conserved quantity. Symplectic orbit integration does

not lead to an error accumulation in the invariants of motion[24] while systematic changes can only arise from numerical errors in solving the ODE. Even for high energy ions the series expansions of third and fourth order are already accurate enough in order to fulfill the condition $J_\parallel = \text{const.}$ for 10^6 bounce periods. Hence, in an axisymmetric configuration excellent long-term orbit dynamic is demonstrated by the quasi-geometric orbit integration method, as long as the ODE set (3.5) is solved accurately."[1]

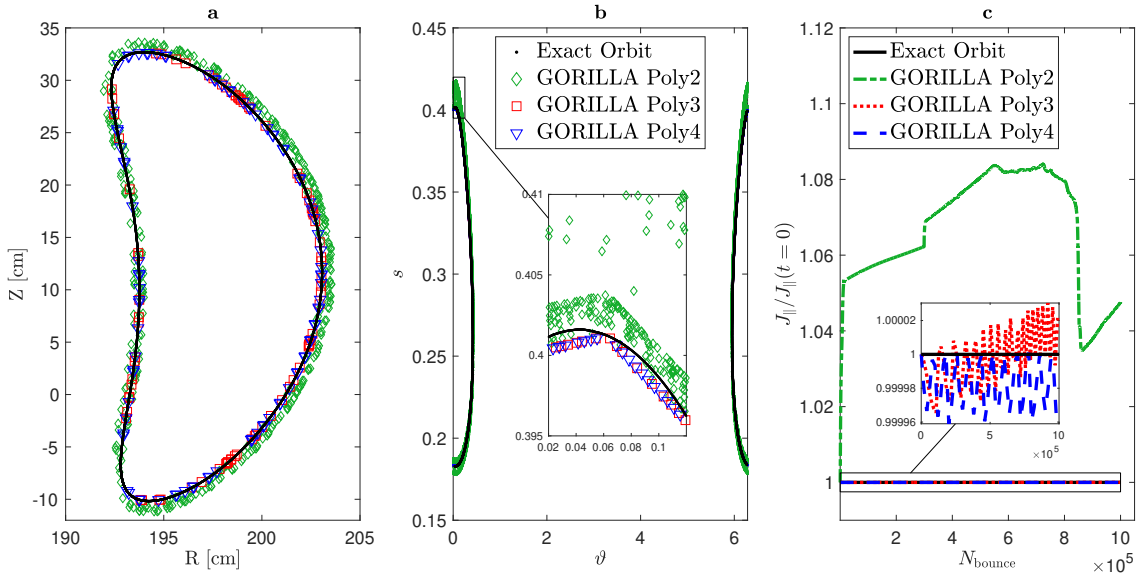

Figure 5.4: (a & b) Poincaré plot ($\varphi = 0$, 10^6 bounce periods) of a trapped 300 keV D ion in axisymmetric ASDEX Upgrade configuration evaluated by GORILLA with the analytical solution in form of a polynomial series truncated at $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$) compared to the exact orbit ($\bullet$). Cylindrical coordinates are used for the Poincaré plot in (a), whereas symmetry flux coordinates are used in (b). The magnification in (b) shows the banana tip and clearly depicts the difference of polynomial orders.
(c) The parallel adiabatic invariant $J_\parallel$ normalized to the value at $t = 0$ is depicted for 10^6 bounce periods: Exact result (solid) is compared to polynomial series truncated at $K = 2$ (dash-dotted), $K = 3$ (dotted) and $K = 4$ (dashed).

5.2 Guiding-center orbits in three-dimensional fields

"It can be seen from Eqs. (3.5) and (3.6) that in the field line limit $\omega_c \to \infty$ guiding-center orbits are straight within spatial cells and magnetic field lines are represented by polygonal chains, respectively. In case of 3D magnetic fields described in non-aligned spatial coordinates, the existence of embedded KAM surfaces is not obvious for such an approximate representation of the field even in the case where these surfaces exist in the exact system. In order to study artificial chaos induced by the linearization, quasi-geometric orbit integration has been performed for low energy particles with

negligible finite Larmor radius (FLR) effects in symmetry flux coordinates (s, ϑ, φ) associated with the axisymmetric tokamak field of the previous section with a harmonic perturbation added to the toroidal co-variant component of the axisymmetric vector potential,

$$A_\varphi = \psi_{\mathrm{pol}}(s)(1 + \varepsilon_M \cos(m_0\vartheta + n_0\varphi)). \tag{5.3}$$

The harmonic indices $m_0 = n_0 = 2$ used in the testing correspond to a non-resonant perturbation which leads only to a corrugation of the magnetic surfaces, with the effect that they are no longer aligned with the coordinate surfaces $s = \text{const.}$.
In Fig. 5.5, Poincaré plots of magnetic field lines obtained by the quasi-geometric integration method for this perturbed configuration are shown at the cross section $\varphi = 0$ together with a cross section of one exact corrugated flux surface. It can be seen that the orbits from the quasi-geometric integrator become more chaotic with increasing perturbation amplitude ε_M.

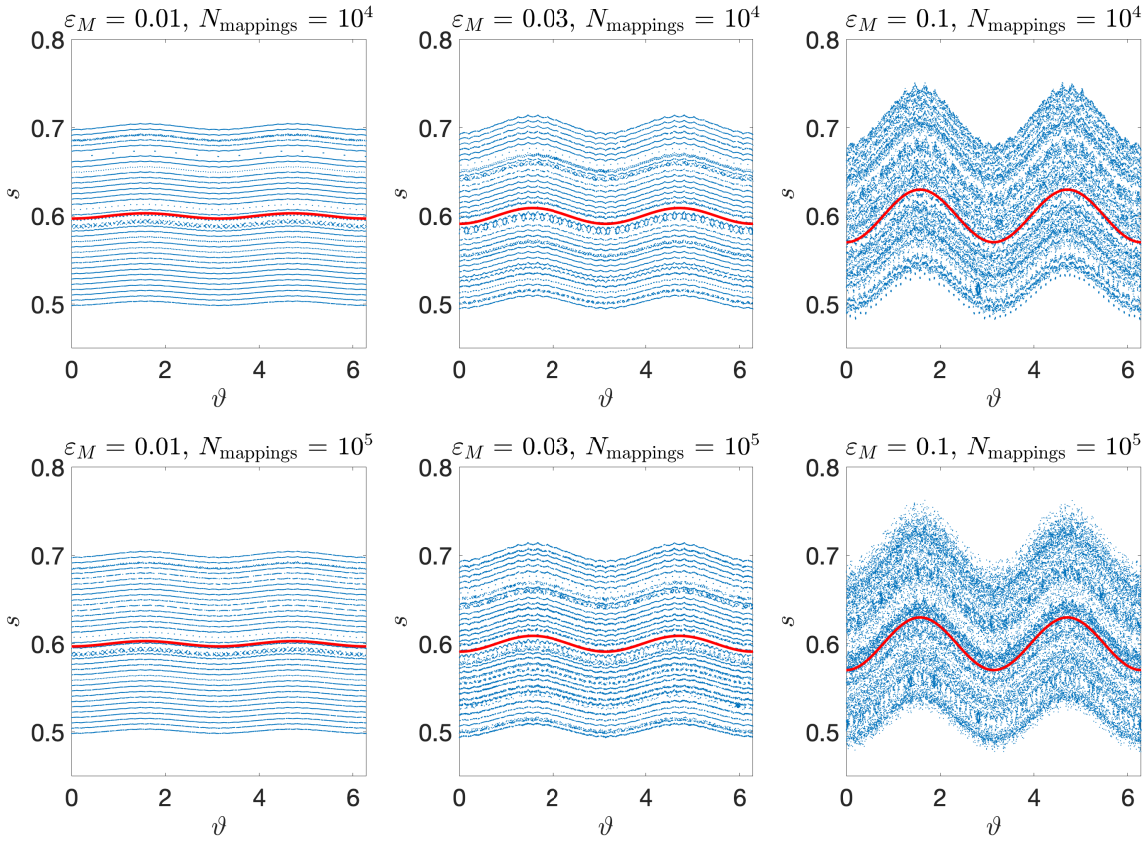

Figure 5.5: Poincaré plots of the orbits in zero Larmor radius limit (field lines) for 10^4 (upper row) and 10^5 (lower row) toroidal mappings and various perturbation amplitudes indicated in the titles. Orbits start at 34 equidistant flux surfaces between $s = 0.5$ and $s = 0.7$ and are evaluated by GORILLA with polynomial series truncated at $K = 2$ and angular grid size of $N_\vartheta = N_\varphi = 30$. Similar results are achieved for angular grid size with incommensurable dimensions, e.g. $N_\vartheta = 29, N_\varphi = 31$. Solid line shows a cross-section of one exact corrugated flux surface.

Diffusive behavior of orbits can be characterized by the variance of the normalized toroidal flux s, accumulated over the time for the ensemble of test particles starting from the same perturbed flux surface. This variance is described by the magnetic field line diffusion coefficient D_M^{ss} as $\langle \delta s^2 \rangle = 2 D_M^{ss} N$ where N is the number of toroidal orbit turns. The effective diffusion coefficient D_M^{ss} computed from the orbits has a strong inverse scaling with poloidal N_ϑ and toroidal N_φ grid sizes and, furthermore, shows in general a small magnitude of diffusion even at coarse angular grid resolution. E.g., at an angular grid size of $N_\vartheta = N_\varphi = 30$ the effective diffusion coefficient D_M^{ss} is of the order of 10^{-13}, 10^{-11} and 10^{-9} for relative perturbation amplitudes of $\varepsilon_M = 1$ %, 3 % and 10 %, respectively. This level of field line diffusion is roughly five orders of magnitude smaller than observed for the initial version of the code[4], and, in the worst case of $\varepsilon_M = 0.1$, results in stochastic diffusion of electrons with the coefficient $D_\perp \sim D_M^{ss} v_\parallel r^2/R \sim 100$ cm^{2}s^{-1}. For smaller ε_M values of the typical order for external magnetic perturbations in tokamaks this numerical diffusion is below the level of classical electron diffusion and, therefore, can be safely ignored.

In the case of field aligned coordinates, chaotization of passing orbits (lines of force of the effective field $\mathbf{B}^*$) can only be caused by the cross-field drift. Such a case is tested below for a strong violation of axial symmetry using as an example the stellarator field configuration described in Ref. [39], namely, a quasi-isodynamic reactor-scale device with five toroidal field periods and a major radius of 25 m. Here the magnetic field has been normalized so that its modulus averaged over Boozer coordinate angles on the starting surface is $B_{00} = 5$ T. Guiding-center orbits were computed with the quasi-geometric integration method in symmetry flux coordinates for strongly passing electrons and ions with $v_\parallel/v = 0.9$ at the starting point on the flux surface $s = 0.6$ with an energy of 3 keV. The numerical diffusion observed for a rather coarse angular grid with the size $N_\vartheta = N_\varphi = 30$ is roughly seven orders of magnitude smaller than the minimum level of the neoclassical diffusion with mono-energetic coefficient D_{11} evaluated in chapter 6 for the same device with particles of the same energy.

For the visualization of a trapped particle orbit, we use orbit footprints on Poincaré sections defined by the condition $v_\parallel(\tau) = 0$, i.e. phase-space hypersurfaces containing orbit turning points. From the two types of these surfaces, those are chosen in which the sign of $v_\parallel$ changes from negative to positive. Fig. 5.6 depicts a poloidal projection of orbit footprints corresponding to a trapped ion with an energy of 3 keV. The orbits have been integrated in symmetry flux coordinates using the polynomial series expansion in several orders and compared to the exact orbit computed with an adaptive RK4/5 integrator. Further, the poloidal coordinates of the footprints have also been converted to cylindrical coordinates and visualized in both coordinate systems,

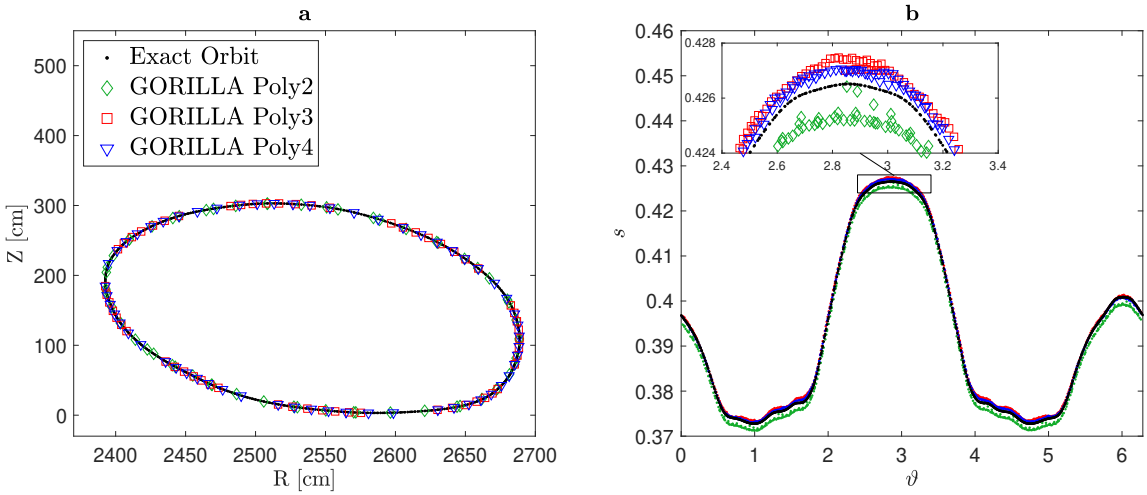

Figure 5.6: (a & b) Poloidal projection of Poincaré sections at $v_\parallel = 0$ switching sign from $-$ to $+$ of a trapped 3 keV D ion in 3D stellarator field configuration. Orbits evaluated by GORILLA with the analytical solution in form of a polynomial series truncated at $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$) are compared to the exact orbit ($\bullet$). Poincaré plot is depicted in cylindrical coordinates (a) and symmetry flux coordinates (b).

respectively. Despite the same starting conditions for all orbits, the magnification in (b) clearly depicts slight differences in the shape of the orbits caused by a finite grid size ($100 \times 100 \times 100$) and different orders of the series expansion.

In contrast to the axisymmetric tokamak field of the previous section, the parallel adiabatic invariant $J_\parallel$ is not an exact invariant in a stellarator. Nevertheless, it should be well preserved as long as the trapped orbit stays within the same class (traverses the same number of field minima over its bounce period) which is the case here. Fig. 5.7 (a) shows the time evolution of $J_\parallel$ for $4 \cdot 10^5$ bounce periods of the corresponding orbits of Fig. 5.6. It can be seen that the results for the exact orbit shows no visible deviation for this configuration. The truncation of the polynomial series at $K = 2$ causes a violation of the system's Hamiltonian structure. This is manifested by an attractor in phase-space, what is clearly visible in Fig. 5.7 (a). A detailed analysis of the corresponding Poincaré sections when the attractor is already fully established (last 10000 evaluations) reveals that the orbit strictly follows a continuous curve, staying in the same class.

However, truncation at higher orders ($K = 3$ & $K = 4$) violates the Hamiltonian structure only negligibly and thus does not lead to non-Hamiltonian features. Nevertheless, in contrast to the exact orbit, $J_\parallel$ evaluated by GORILLA with these polynomial orders is not accurately conserved. In particular, the value of $J_\parallel$ meanders randomly around the exact value which is caused by the diffusive behavior of the orbit induced by the piecewise linearization of the electromagnetic field.

Fig. 5.7 (b) shows the modulus of the relative error of the parallel adiabatic invariant

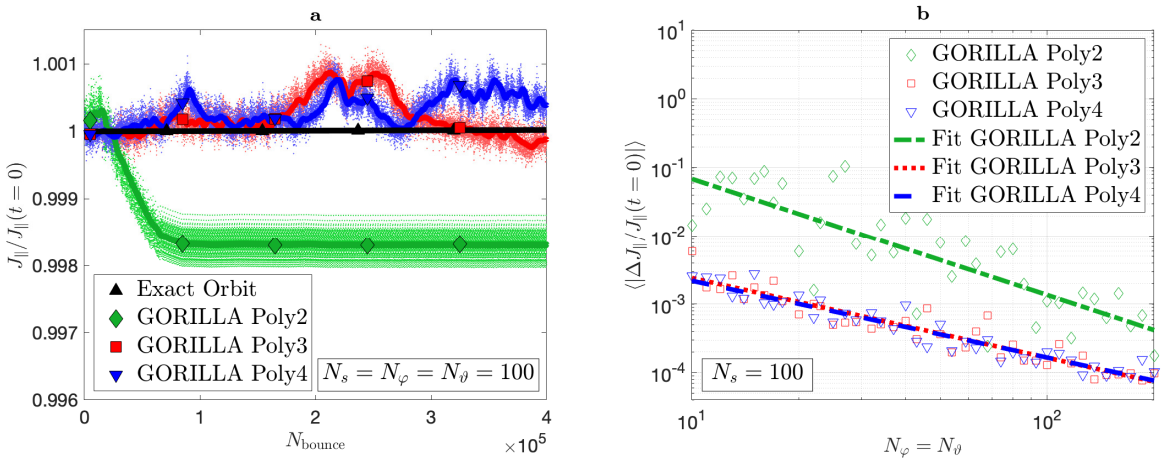

Figure 5.7: (a) Evolution of the parallel adiabatic invariant $J_\parallel$ normalized to the value at $t = 0$ of the trapped particle orbit in Fig. 5.6 over $4 \cdot 10^5$ bounce times. Lines in (a) represent window filtered trend of the $J_\parallel$ data points: Exact result ($\triangle$) is compared to polynomial series truncated at $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$).
(b) Modulus of the relative error of the parallel adiabatic invariant $J_\parallel$ averaged over 10^5 bounce times as a function of the angular grid size $N_\varphi = N_\vartheta$. Orbits are evaluated by GORILLA with the analytical solution in form of a polynomial series truncated at $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$). The fits of the results are depicted with lines in accordance with the legend.

$J_\parallel$ averaged over 10^5 bounce times as a function of the angular grid size $N_\varphi = N_\vartheta$ varied from 10 to 200. At a moderate angular grid size of $N_\vartheta = N_\varphi = 28$ (in accordance with the Nyquist-Shannon sampling theorem[58,59] explained below) this mean relative error stays already below 10^{-3} for the orders $K = 3$ & $K = 4$. Thus, the finite grid size used in the quasi-geometric orbit computation does not lead to a significant error accumulation. With regard to the comparatively large number of bounce times, even the order $K = 2$ which is naturally the fastest with respect to CPU time yields quite accurate results at sufficient grid resolution."[1]

Chapter 6

Monte Carlo evaluation of neoclassical transport coefficients and performance benchmark

The content of this chapter including the figures can also be found in section IV of Ref. [1] formulated by the author and including some minor modifications.

"Evaluation of neoclassical transport coefficients using the Monte Carlo method[8,9] is widely used for stellarators and tokamaks with 3D perturbations of the magnetic field[11–16,60]. An advantage of this method in its original, full-f form is the use of test particle guiding-center orbits without requiring the model simplifications needed in (more efficient) local approaches. The Monte Carlo methods thus provide an unbiased reference point in cases where those simplifications affect the transport in a manner such as that for regimes with significant role of the tangential magnetic drift[61–63]. An obvious disadvantage is that for realistic magnetic configurations Monte Carlo methods are CPU-intensive with most of the CPU time spent for the integration of the guiding-center motion. The application of the proposed quasi-geometric integration method for this purpose instead of the usual Runge-Kutta method results in a visible speed-up of the computations without significantly biasing the results. Here, this application is made for benchmarking purposes on the assumption that the inaccuracies in orbit integration which are tolerable in computations of transport coefficients are also tolerable in global modelling of macroscopic plasma parameters.
The proposed orbit integration method is applied within a standard Monte Carlo algorithm[8] using the Lorentz collision model for the evaluation of the mono-energetic radial diffusion coefficient D_{11}. The latter is determined via the average square deviation of the normalized toroidal flux s from its starting value s_0 as follows,

$$D_{11} = \frac{1}{2t}\langle (s(t) - s_0)^2 \rangle. \tag{6.1}$$

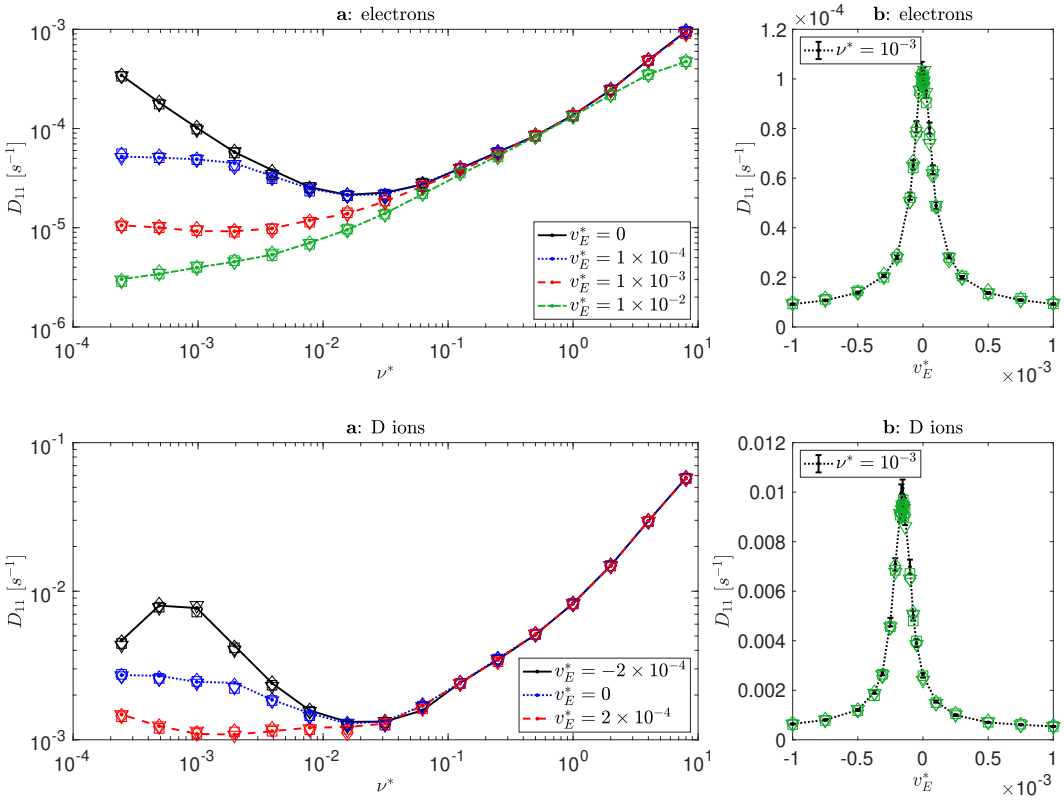

Figure 6.1: Mono-energetic radial diffusion coefficients D_{11} for electrons (top) and deuterium ions (bottom) as functions of (a) normalized collisionality ν^* and (b) Mach number v_E^*. Lines of various styles (see the legends) - reference computation, markers - results of quasi-geometric integration with polynomial solution of the order K for $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$). Error bars indicate 95 % confidence interval.

Here, angle brackets $\langle \dots \rangle$ denote an ensemble average, $s(0) = s_0$, and the test particle tracing time t is chosen to be longer than the local distribution function relaxation time τ_{rel} but shorter than the radial transport time, $t = 10\tau_{\mathrm{rel}}$. A Monte Carlo collision operator identical to that of Ref. [8] is applied here between constant collisionless orbit integration steps Δt. These steps are small enough compared to the typical bounce time τ_{b} and collision time τ_{c},

$$\Delta t = \min\left(\frac{\tau_{\mathrm{b}}}{20}, \frac{\tau_{\mathrm{c}}}{20}\right). \tag{6.2}$$

Here, $\tau_{\mathrm{c}} = 1/\nu$ and $\tau_{\mathrm{b}} = 2\pi R_0/(v N_{\mathrm{tor.}})$ with ν, R_0, v and N_{tor} denoting collisional deflection frequency, major radius, particle velocity and number of toroidal field periods, respectively. The relaxation time τ_{rel} is determined as the largest of τ_{c} and $\tau_{\mathrm{b}}^2/\tau_{\mathrm{c}}$.

In the present example, the mono-energetic radial diffusion coefficient has been

evaluated for the quasi-isodynamic stellarator configuration[39] used also for collisionless orbits in section 5.2. Guiding-center orbits were computed with the quasi-geometric integration method in symmetry flux coordinates using polynomial series solutions of various orders K. The grid size $N_s \times N_\vartheta \times N_\varphi = 100 \times 60 \times 60$ was selected to be appropriate for minimizing the numerical diffusion (see the previous section.) In a reference computation, guiding-center equations in symmetry flux variables with electromagnetic field interpolated by 3D cubic splines were integrated by an adaptive RK4/5 integrator. Computations were performed for a large ensemble size of 10000 particles in order to minimize statistical errors.

The results for D_{11} computed for 3 keV electrons and ions at $s_0 = 0.6$ are presented in Fig. 6.1. Values of radial electric field E_r and deflection frequency ν, which determine transport regimes, are respectively characterized here by two dimensionless parameters[64], Mach number $v_E^* = cE_r/(vB_0)$ and collisionality $\nu^* = (R_0\nu)/(\iota v)$, where ι is the rotational transform. For the ions, in addition to the $\mathbf{E} \times \mathbf{B}$ rotation, also the tangential magnetic drift plays a significant role which can be seen from the shift of the D_{11} maximum on v_E^* dependence. The results of quasi-geometric integration stay in agreement with the reference computation within the 95 % confidence interval in all cases even for the lowest order polynomial solution $K = 2$. Therefore, in calculations of this kind a significant gain in the computational efficiency can be obtained as shown below.

Moreover, we compare the performance and scaling for parallel computation of guiding-center orbits using the quasi-geometric orbit integration method with computations using standard reference integrators (RK4 and adaptive RK4/5). For this, different integrators have been used within D_{11} computation described above for a particular choice of dimensionless parameters, $v_E^* = 10^{-3}$ and $\nu^* = 10^{-3}$, and an increased ensemble size of 30000 test particles. The numerical experiment has been performed on a single node of the COBRA cluster of MPCDF with 40 CPU cores (Intel Xeon Gold 6126) running 80 concurrent threads with hyperthreading.

The reference value for the transport coefficient, $D_{11,\mathrm{ref}}$, and the reference CPU time are obtained by orbit integration with an adaptive RK4/5 integrator with a relative tolerance of 10^{-9}. The accuracy of the D_{11} evaluation using different computation parameter settings is represented by the relative error $\delta D_{11}/D_{11,\mathrm{ref}}$ where $\delta D_{11} = |D_{11} - D_{11,\mathrm{ref}}|$. The CPU time purely used for orbit integration serves as a measure for the computational effort of the methods. This given CPU time does not contain any overhead operations, e.g. the construction of the grid, generation of random numbers for pitch-angle scattering and computation of D_{11} by evaluating Eq. 6.1 with the help of a least-squares regression.

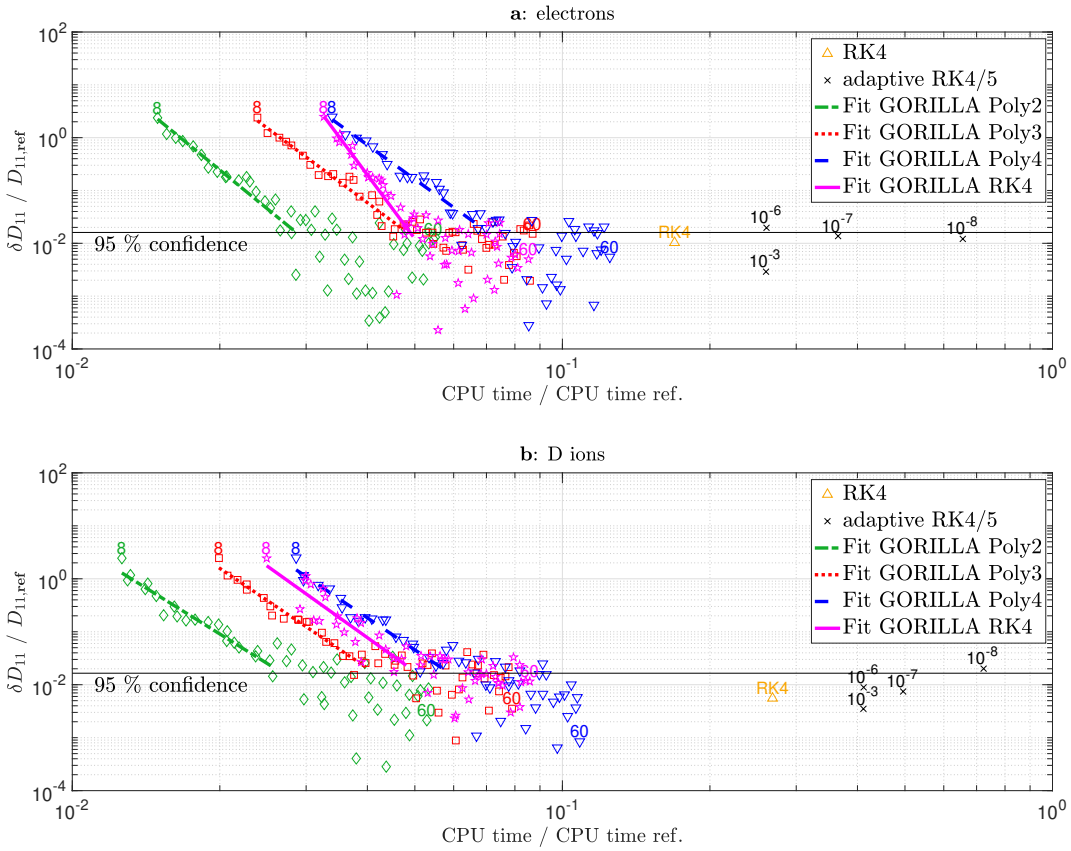

Figure 6.2: Relative error of mono-energetic radial transport coefficient D_{11} of electrons (top) and D ions (bottom) vs. relative CPU time. The compared orbit integration methods are: Runge-Kutta 4 ($\star$), Adaptive RK4/5 with various relative errors indicated in the plot ($\times$), quasi-geometric integration with polynomial solution (GORILLA Poly) of the order $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$), and with RK4 solution (GORILLA RK4, $\triangle$). The fits of the results are depicted with lines in accordance with the legend. The random error of the reference result, $D_{11,\mathrm{ref}}$, is depicted as a horizontal line limiting its 95 % confidence interval.

Fig. 6.2 shows the relative error of the mono-energetic radial transport coefficient versus the relative CPU time of computations using the quasi-geometric orbit integration method with the polynomial series solution of various orders, GORILLA Poly, and the iterative scheme with RK4 integration and Newton steps, GORILLA RK4. Accuracy and CPU time of quasi-geometric orbit integrations have been varied by mutually changing the angular grid size $N_\vartheta \times N_\varphi$ from 8×8 to 60×60 while keeping the radial grid size constant at $N_s = 100$. In the stellarator configuration of Ref. [39] used here, the number of toroidal harmonic modes per field period is 14, leading to a minimum toroidal grid size $N_\varphi = 28$ in order to satisfy the Nyquist-Shannon sampling theorem[58,59]. Therefore, regression lines are drawn for the range of data points with grid sizes from 8×8 until 28×28, clearly showing a convergent behavior of D_{11} with increasing grid refinement. Furthermore, the adaptive RK4/5 integration

is additionally performed with relative tolerances of 10^{-3}, 10^{-6}, 10^{-7} and 10^{-8}, respectively. Note that the computational speed of the adaptive RK4/5 integration with a relative tolerance of 10^{-6} cannot be increased by higher relative tolerances, e.g. 10^{-3}, since the macroscopic Monte Carlo time step, Δt Eq. (6.2), is already elapsed within a single RK4/5 step with sufficient accuracy. Hence, also the non-adaptive Runge-Kutta 4 method is tested, which naturally needs one field evaluation less per time step than RK4/5. In all cases, the relative error of the RK4/5 and Runge-Kutta 4 results is determined here mainly by statistical deviations, with a random error dominating the bias.

Apart from statistical errors due to Monte Carlo sampling, a limit for capturing all toroidal and poloidal field harmonics is given by a minimum grid size of two points per period due to the Nyquist–Shannon sampling theorem. Fig. 6.2 visibly shows that statistical fluctuations already dominate the bias of all variants of the quasi-geometric integration method above this sampling threshold, despite the large ensemble size of 30000 particles. To avoid possible sampling artifacts at an even higher particle count, we consider the quasi-geometric orbit integration method at the toroidal grid size N_φ of at minimum twice the number of toroidal modes in the magnetic field configurations. The variant with the polynomial series solution truncated at $K = 2$ (GORILLA Poly 2) at this grid resolution can be considered the fastest sufficiently accurate tested method to compute D_{11} for thermal ions and electrons. In the case of D ions with an energy of 3 keV this method is one order of magnitude faster than the Runge-Kutta 4 integrator which is the fastest reference method."[1]

Chapter 7

Confinement of fusion alpha particles in a realistic stellarator configuration and performance benchmark

"The space in which stellarators are optimized has about 50 degrees of freedom."[65]

When designing stellarators, one must be careful to ensure the existence of magnetic surfaces within a large volume[66] as well as a maximum confinement of single particle trajectories.[67] Furthermore, neoclassical transport must be optimized[55] and also practical considerations must be taken into account. Finding a stellarator configuration with favourable properties that satisfy these conditions poses a high-dimensional multi-objective optimization problem.[68]

In particular, one major objective among these is to minimize the losses of energetic fusion alpha particles (kinetic energy of 3.5 MeV) over their slowing-down time. This is important for heating the bulk plasma of a reactor instead of losing valuable kinetic energy to the first wall and at the same time risking potential damages of the reactor device.

In the past, proxy models[69,70] were developed to estimate fusion alpha losses with the drawback of not being able to capture the full physics of drift orbits. In contrast, direct tracing of guiding-center orbits and subsequent statistical estimation of losses leads to accurate results.[71,72] However, direct massive computation of guiding-center orbits with usual numerical methods is relatively time consuming in comparison with other computations in the stellarator optimization loop, e.g., finding 3D magnetohydrodynamic equilibria.[68]

Recently, the symplectic integrator for guiding-center orbits of Ref. [28] was applied to alpha loss computation in stellarator configurations, where a significant speed up of a factor 3-6 compared to an adaptive Runge-Kutta integrator was achieved. Furthermore, with an early classification into regular and chaotic orbit types a computational speed up of approximately one order of magnitude can be reached. (Regular guiding-center

orbits stay confined to the drift-surface and only chaotic orbits have to be traced up to the end, as their behaviour is unpredictable.) Thus, the simulation of fusion alpha losses via direct guiding-center orbit computations as part of the stellarator optimization loop comes within reach.[68]
Due to the promising performance benchmark of the presented quasi-geometric integration method, which was demonstrated in chapter 6, it is the goal of this chapter to study its potential application to the confinement simulation of fusion alpha particles in a realistic stellarator configuration.
Here, the quasi-isodynamic stellarator field configuration described in Ref. [39], which was already used in section 5.2 and chapter 6, will serve as an optimized realistic reactor-scale device.

However, there are two major concerns why the computation of fusion alpha particle losses with the help of the quasi-geometric integration method is not trivially feasible. On the one hand, the Runge-Kutta 4 integration error of a single step traversing the tetrahedral cell scales with the third power of the Larmor radius; see Eq. (3.13). Due to this reason, the analytical solution in form of a truncated polynomial series was originally developed for mild energies (thermal electrons and ions). But in the context of tracing fusion alphas, the Larmor radius of a 3.5 MeV alpha particle is roughly 24 times larger than that of a 3 keV D ion. Consequently, inaccurate integration of the linear ODE set inside a tetrahedral cell, Eqs. (3.5), produces a systematic error and results in non-Hamiltonian behavior of the guiding-center orbits. A similar behavior was already demonstrated in section 5.1 where the solution for a 3 keV D ion in form of the polynomial series was already truncated at $K = 2$ and as a consequence the system's Hamiltonian nature was violated.
On the other hand, the linearization of the electromagnetic field within the tetrahedral cells causes inherent artificial chaos when applying the quasi-geometric integration method in 3D fields. In section 5.2 this chaos was studied for low energy particles with negligible finite Larmor radius (FLR) effects in non-aligned coordinates and for 3 keV D ions in the above described stellarator configuration utilizing symmetry flux coordinates. In this chapter all computations with the quasi-geometric integration method are performed in symmetry flux coordinates (s, ϑ, φ), since thereby chaos is strongly reduced (in comparison to non-aligned coordinates). As mentioned in section 5.2, when utilizing field-aligned coordinates, the chaotization of passing orbits (lines of force of the effective field $\mathbf{B}^*$) are caused by the cross-field drift. The latter quantity is dependent on the particle's Larmor radius. E.g., if we consider $\mathbf{B} \times \nabla B$ drift motion, the normalized drift velocity scales roughly linearly with the Larmor

radius ρ, namely $v_d/v \approx \mathcal{O}(\rho/L)$, where v_d, v and L are the guiding-center drift velocity, the particle's absolute velocity and the typical scale length of the variation of $B(x)$, respectively.[73]

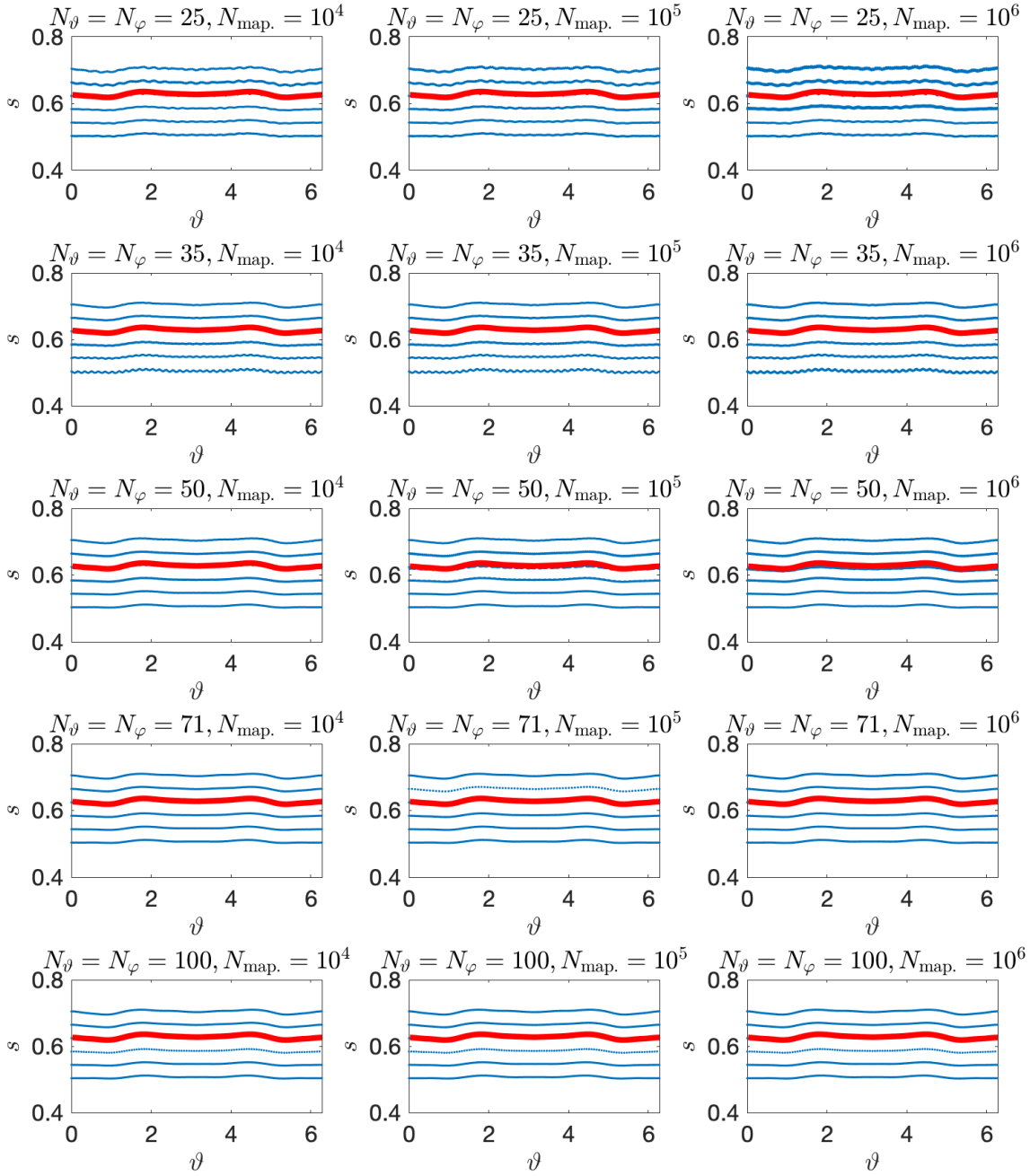

Figure 7.1: Poincaré plot ($\varphi = 0$) of strongly passing 3.5 MeV fusion alpha particles in 3D stellarator field configuration. The guiding-center orbits are started from 6 equidistant flux surfaces between $s = 0.5$ and $s = 0.7$ with a pitch parameter $\lambda = 0.95$. Orbits are evaluated by GORILLA with the polynomial solution truncated at $K = 4$ and a radial grid size of $N_s = 100$. Similar results are achieved by GORILLA with the numerical option utilizing an adaptive Runge-Kutta 4/5 integrator in order to minimize integration errors. The angular grid size ($N_\vartheta \times N_\varphi$) is mutually increased from top ($N_\vartheta = N_\varphi = 25$) to bottom ($N_\vartheta = N_\varphi = 100$). The number of toroidal mappings is increased from left ($N_{\text{mappings}} = 10^4$) to right ($N_{\text{mappings}} = 10^6$). Solid (red) lines show exact drift-surfaces obtained by reference guiding-center orbit computations.

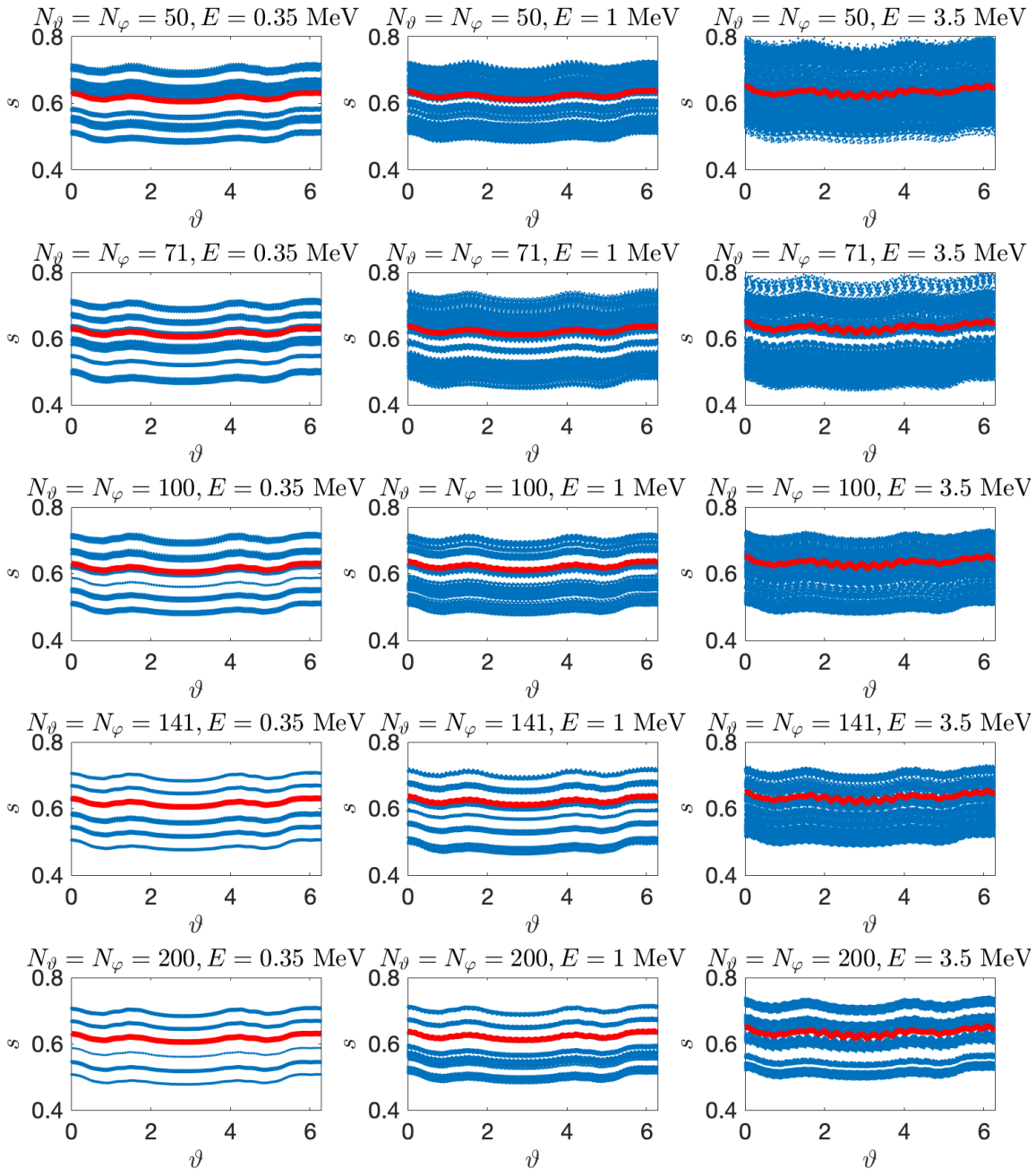

Figure 7.2: Poloidal projection of Poincaré sections at $v_\parallel = 0$ switching sign from $-$ to $+$ of deeply trapped alpha particles in 3D stellarator field configuration. The guiding-center orbits are started from 6 equidistant flux surfaces between $s = 0.5$ and $s = 0.7$ with a pitch parameter $\lambda = 0.2$. Orbits are evaluated by GORILLA with the polynomial solution truncated at $K = 4$ and a radial grid size of $N_s = 100$ for 10^4 bounces. Similar results are achieved by GORILLA with the numerical option utilizing an adaptive Runge-Kutta 4/5 integrator in order to minimize integration errors. The angular grid size ($N_\vartheta \times N_\varphi$) is mutually increased from top ($N_\vartheta = N_\varphi = 50$) to bottom ($N_\vartheta = N_\varphi = 200$). The kinetic energy of the alpha particles is increased from left (0.35 MeV) to right (3.5 MeV). Solid (red) lines show exact poloidal projections of Poincaré sections obtained by reference guiding-center orbit computations.

Thus, the level of artificial chaos due to the linearization is a function of the particle's Larmor radius and might be quite severe in the case of high-energetic fusion alpha particles.
Nevertheless, regarding the first concern, the occurring integration error can be made in principle negligibly small by performing the integration of the linear ODE set, Eq. (3.5), with an adaptive Runge-Kutta 4/5 integrator, yet computational efficiency will decrease.
For a detailed investigation of the second concern, numerical experiments are performed. First of all, Poincaré plots at the cross section $\varphi = 0$ of strongly passing 3.5 MeV fusion alpha particles are obtained for six different starting positions by the quasi-geometric integration method with the polynomial solution truncated at $K = 4$. These Poincaré plots are depicted in Fig. 7.1 for 10^4, 10^5 and 10^6 toroidal mappings. Furthermore, the angular grid size ($N_\vartheta \times N_\varphi$) is mutually increased from $N_\vartheta = N_\varphi = 25$ to $N_\vartheta = N_\varphi = 100$ while the radial grid size is kept constant at $N_s = 100$. In addition, exact drift-surfaces obtained by reference guiding-center orbit computations (adaptive Runge-Kutta 4/5 in splined fields) are visible for comparison for one of these six starting positions. If one carefully observes these Poincaré plots, some tiny chaotic features become visible at low grid resolution ($N_\vartheta = N_\varphi = 25$), whereas at moderate grid resolution ($N_\vartheta = N_\varphi = 70$) the guiding-center orbits seem to be perfectly confined to the drift-surface and appear undistinguishable to the reference orbits.
For the visualization of trapped alpha particle guiding-center orbits, we use orbit footprints on Poincaré sections defined by the condition $v_\parallel(\tau) = 0$, i.e. phase-space hypersurfaces containing orbit turning points. From the two types of these surfaces, those are chosen in which the sign of $v_\parallel$ changes from negative to positive. Fig. 7.1 depicts poloidal projections of orbit footprints corresponding to deeply trapped alpha particles with energies of 0.35 MeV, 1 MeV and 3.5 MeV. Again, the guiding-center orbits are obtained for six different starting positions by the quasi-geometric integration method with the polynomial solution truncated at $K = 4$ and for one of these six starting positions, exact drift-surfaces are obtained by reference guiding-center orbit computations. Here, the angular grid size ($N_\vartheta \times N_\varphi$) is mutually increased from $N_\vartheta = N_\varphi = 50$ to $N_\vartheta = N_\varphi = 200$ while the radial grid size is kept constant at $N_s = 100$. All Poincaré projections are shown for 10^4 bounces.
It is clearly visible that the level of the artifical chaos is much stronger for trapped particles than for passing particles. In the latter case the chaos remains evident even for the comparatively high angular grid resolution of $N_\vartheta = N_\varphi = 200$. Furthermore, one can see that the chaos increases with the particle energy and scales inversely with the grid size. For both, passing and trapped particles (Figs. 7.1 and 7.2) all results

of the quasi-geometric integration method with the polynomial solution truncated at $K = 4$ were also compared to results obtained by the quasi-geometric integration method utilizing and adaptive Runge-Kutta 4/5 integrator and did not show any significant differences. Thus, the observed chaos is most likely induced by the linearization of the electromagnetic field and not by the truncation of the polynomial series solution.

In section 5.2 an effective diffusion coefficient D_M^{ss} was used to characterize the level of the artificial chaos. Due to the comparatively large Larmor radius in the case of fusion alpha particles, such a computation cannot be performed here, because the drift motion will cover the chaotic motion. In principle, one could develop an effective diffusion coefficient utilizing Poincaré plots and projections. Such a measure would allow to investigate in detail how the chaos scales with the pitch parameter and the energy of the particle, but this is a topic for future investigations.
However, in this chapter the goal is to examine the potential application for the confinement simulation of fusion alpha particles in a realistic stellarator configuration. Thus, instead of computing an effective diffusion coefficient it appears to be more straight-forward to compute the confinement fraction f_c, which is the fraction of fusion alpha particles that stay confined in the stellarator configuration after a given trace time. Such a measure computed with GORILLA can then be compared to the results obtained by reference guiding-center orbit computations. It is clear that in the case of diffusive behavior of the guiding-center orbits, the confinement fraction f_c will be strongly a non-linear function of the diffusion coefficient. Nevertheless, the confinement fraction f_c can be considered to be a sufficient measure for our purpose.

"Fig. 7.3 shows the confined fraction f_c of 3.5 MeV fusion alpha particles as a function of the trace time in the 3D stellarator field configuration of Ref. [39]. In Fig. 7.3 (a) 1000 particles are traced for 1 s, whereas in Fig. 7.3 (b) 10000 particles are traced for 0.01 s. All guiding-center orbits are started from $s = 0.6$ with a homogeneous distribution of the pitch parameter. The reference result is obtained by utilizing the "exact" guiding-center orbits which are computed with an adaptive Runge-Kutta 4/5 integrator with a relative tolerance of 10^{-9} in splined fields. The quasi-geometric integration method is performed with its *Polynomial 4* and *adaptive Runge-Kutta 4/5* options and in addition three different settings for the grid size are examined, namely $N_s = N_\vartheta = N_\varphi = 70$, $N_s = N_\vartheta = N_\varphi = 100$ and $N_s = N_\vartheta = N_\varphi = 200$."[3]
The estimated standard deviation σ scales inversely with the total number of particles

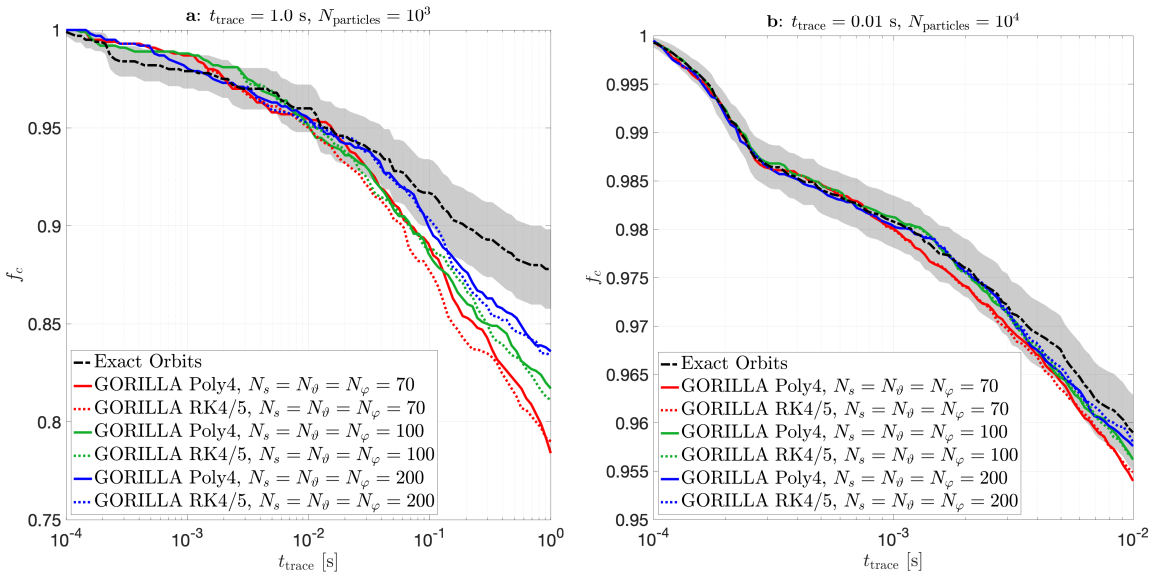

Figure 7.3: Confined fraction f_c of 3.5 MeV fusion alpha particles as a function of the trace time in 3D stellarator field configuration. (a) 1000 particles are traced for 1 s. (b) 10000 particles are traced for 0.01 s.
The guiding-center orbits are started from $s = 0.6$ with a homogeneous distribution of the pitch parameter. Orbits are evaluated by GORILLA and the results of f_c are compared to those obtained by exact guiding-center orbits from the reference guiding-center orbit computation. In the case of GORILLA, the method (*Polynomial* or *Runge-Kutta*) and the choice of the grid size are in accordance with the legend. Error bands at $\pm 1.96\sigma$ around the curve of f_c obtained with the exact guiding-center orbits describe the 95 % confidence interval due to the Monte Carlo error. *The figure is taken from Ref. [3]*

$N_{\mathrm{particles}}$ of the Monte Carlo computation as

$$\sigma = \sqrt{\frac{f_c(1-f_c)}{N_{\mathrm{particles}}}}. \tag{7.1}$$

The gray error bands at $\pm 1.96\sigma$ around the curve of f_c obtained with the exact guiding-center orbits describe the 95 % confidence interval due to the Monte Carlo error. The error bands for the computations with the quasi-geometric integration method are of similar size and not depicted due to clarity of the figure.
The trace time of 1 s in Fig. 7.3 (a) is in the order of the slowing-down time of 3.5 MeV fusion alpha particles. In the optimized stellarator configuration of Ref. [39] roughly 87 % of the particles stay confined in the fusion device after the trace time of 1 s. One can clearly see that none of the computations performed with the quasi-geometric integration method lie within the reference 95 % confidence interval after this time. Furthermore, it is worth mentioning, that the choice of the GORILLA method (*Polynomial 4* or *adaptive Runge-Kutta 4/5*) does not significantly influence the result.

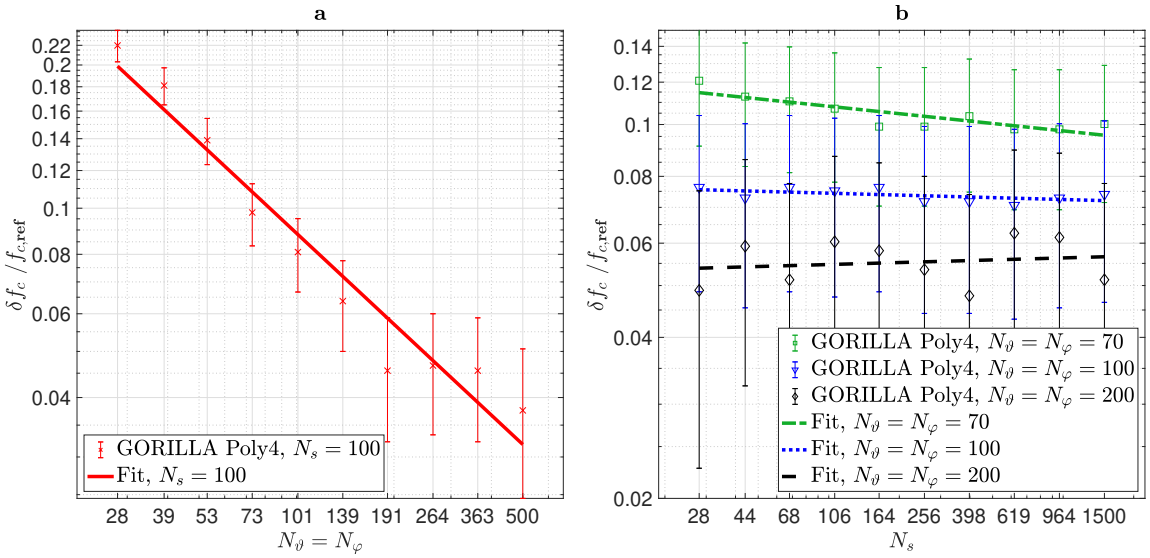

Figure 7.4: Relative error of the confined fraction f_c of 3.5 MeV fusion alpha particles at $t_{\mathrm{trace}} = 1$ s vs. the (a) angular and (b) radial grid size of GORILLA in 3D stellarator field configuration. The guiding-center orbits are evaluated by GORILLA with the polynomial solution truncated at $K = 4$. Orbits are started from $s = 0.6$ with a homogeneous distribution of the pitch parameter. Error bars indicate the 95% confidence of the relative error of the confined fraction due to the Monte Carlo error (1000 particles). The results as function of the grid size and their respective fits are depicted with markers and lines in accordance with the legend.

However, with an increased grid resolution the results obtained with GORILLA come closer to the reference confined fraction $f_{c,\mathbf{ref}}$. In order to examine the accuracy of the GORILLA results, the relative error of the confined fraction is defined as $\delta f_c / f_{c,\mathbf{ref}} = |f_c - f_{c,\mathbf{ref}}| / f_{c,\mathbf{ref}}$. Fig. 7.4 shows how this relative error of the confined fraction after 1 s of trace time scales with (a) the angular and (b) the radial grid size. Integration is performed with GORILLA and the polynomial solution truncated at $K = 4$. In Fig. 7.4 (a) one can see how the relative error scales inversely with the angular grid size, whereas in Fig. 7.4 (b) the variation of the radial grid size does not significantly influence the relative error at three given angular grid sizes. Albeit the relative error can be reduced with an increased angular grid size, the relative error stays at roughly 4 % for a comparatively high angular grid resolution of $N_\vartheta = N_\varphi = 500$. (In the chapter 6 an angular grid resolution of $N_\vartheta = N_\varphi = 28$ was sufficient to reach 1 % relative error at the D_{11} computation of 3 keV D ions.)
With this observation, it becomes clear that the quasi-geometric integration method is not suited to be used for tracing of high-energetic 3.5 MeV fusion alpha particles for the slowing-down time of 1 s. This is due to the inherent chaos of the method which is induced by the linearization of the electromagnetic field and which scales strongly with the particle's Larmor radius.

Yet, it can be clearly seen in Fig. 7.3 (b) that for the trace time of 0.01 s most of the confined fraction results computed with the quasi-geometric integration method lie within the reference 95 % confidence interval. Here, the number of particles was increased by one order of magnitude (in comparison with Fig. 7.3 (a)) and, therefore, the confidence interval became by roughly a factor 3 more narrow. Thus, it is worth to examine the behavior of the quasi-geometric integration method for a shorter trace time than 1 s. As mentioned above, an early classification into regular and chaotic orbit types, as it is done in Ref. [68], can result in a significant computational speed up for the computation of alpha losses. Hypothetically, for a short trace time (like 0.01 s) the inherent chaos of the quasi-geometric integration might only slightly affect the guiding-center orbits. Therefore, correct orbit classification would still be possible, while quasi-geometric guiding-center orbit computation needed for the classification might be computationally more efficient than the symplectic integration method of Ref. [28]. Hence, in order to be beneficial for the computation of alpha losses, the computational efficiency of the quasi-geometric integration method must be higher than a comparable integrator for the same achieved accuracy.

In accordance with Fig. 7.4 the control parameter for the accuracy of the (confined fraction) result is the angular grid size (in addition to the choice of the GORILLA polynomial order). At the same time the computational efficiency of GORILLA decreases with higher grid resolution, since more numerical operations must be performed. (In GORILLA, intersections between the guiding-center orbit and the cell boundaries are inherently computed and the number of these intersections increases linearly with the grid size.)

Fig. 7.5 shows for various polynomial orders that the CPU time needed for the computation of the confined fraction linearly increases with the angular grid size; as it was expected.

Furthermore, in a similar manner as in chapter 6 we compare the performance and scaling for parallel computation of guiding-center orbits using the quasi-geometric orbit integration method with computations using a standard reference integrator (adaptive RK4/5 in splined fields). For this purpose, different settings of GORILLA (grid size and polynomial order) have been used within the f_c computation and an increased ensemble size of 30000 test particles was chosen in order to decrease the Monte Carlo error. The numerical experiment has been performed on a single node with 20 CPU cores (Intel Xeon E5-2630 v4 @ 2.20 GHz) running 40 concurrent threads with hyperthreading.

The reference value for the confined fraction, $f_{c,\mathbf{ref}}$, and the reference CPU time are obtained by orbit integration with an adaptive RK4/5 integrator with a relative

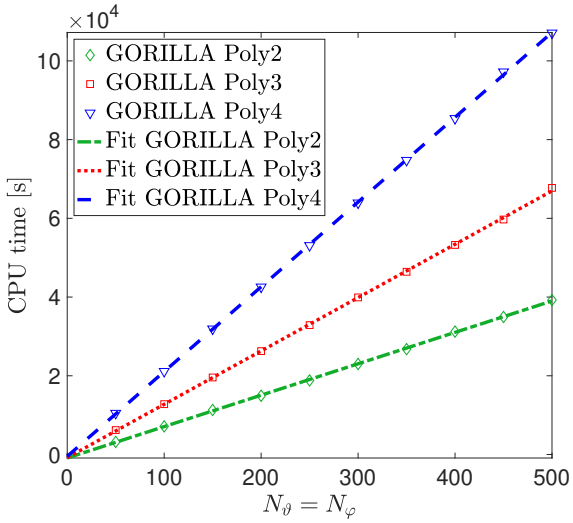

Figure 7.5: CPU time for the computation of the confined fraction f_c of 3.5 MeV fusion alpha particles after a trace time of 0.01 s vs. the angular grid size of GORILLA in 3D stellarator field configuration. The radial grid size is kept constant at $N_s = 100$. The guiding-center orbits of 10000 particles are evaluated by GORILLA with various polynomial orders in accordance with the legend. Orbits are started from $s = 0.6$ with a homogeneous distribution of the pitch parameter.

tolerance of 10^{-9}. The CPU time purely used for orbit integration serves as a measure for the computational effort of the methods. This given CPU time does not contain any overhead operations, e.g. the construction of the grid or generation of random numbers.

Fig. 7.6 shows the relative error of the confined fraction for a trace time of 0.01 s versus the relative CPU time of the computations using the quasi-geometric orbit integration method with the polynomial series solution of various orders, GORILLA Poly. Accuracy and CPU time of quasi-geometric orbit integrations have been varied by mutually changing the angular grid size $N_\vartheta \times N_\varphi$ from 28×28 to 200×200 while keeping the radial grid size constant at $N_s = 100$. Regression lines are drawn for the range of data points with grid sizes from 28×28 until 100×100, clearly showing a convergent behavior of f_c with increasing grid refinement. Furthermore, the adaptive RK4/5 integration is additionally performed with relative tolerances of 10^{-3}, 10^{-6}, 10^{-7} and 10^{-8}, respectively.

In all cases, the relative error of the RK4/5 results as well as of the GORILLA Poly results with angular grid sizes higher than 100×100 is determined here mainly by statistical deviations, with a random error dominating the bias. Therefore, we consider the quasi-geometric integration method with an angular grid resolution that is high enough to obtain a result for the confined fraction that lies within the 95 % confidence interval of the reference result, $f_{c,\mathbf{ref}}$, to be sufficiently accurate.

The fastest sufficiently accurate method depicted in Fig. 7.6 is GORILLA with the solution in form of a polynomial series truncated at $K = 2$ and an angular grid size of $N_\vartheta \times N_\varphi = 100 \times 100$. This method is roughly 3 times faster than the fastest reference method, namely the adaptive RK4/5 integrator with a relative tolerance of 10^{-3}. As stated in the introduction of this chapter, the symplectic integration

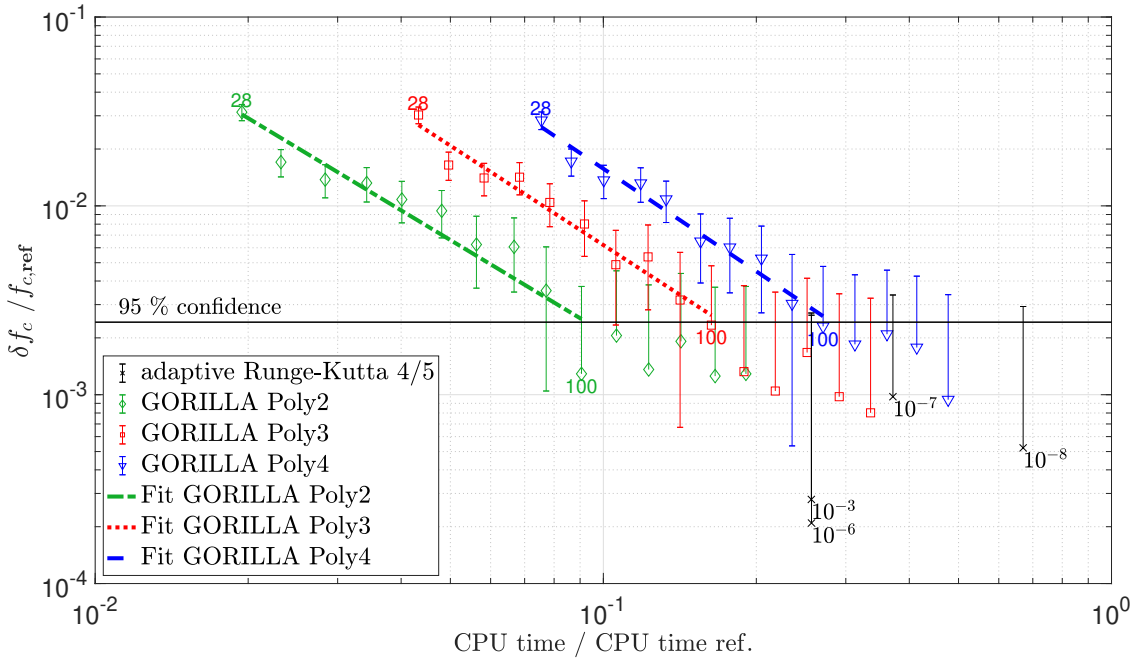

Figure 7.6: Relative error of the confined fraction f_c of 3.5 MeV fusion alpha particles after a trace time of 0.01 s vs. relative CPU time in 3D stellarator field configuration. The guiding center orbits of 30000 particles are started from $s = 0.6$ with a homogeneous distribution of the pitch parameter. The compared orbit integration methods are: GORILLA Poly of the order $K = 2$ ($\Diamond$), $K = 3$ ($\square$) and $K = 4$ ($\triangledown$), and adaptive RK4/5 with various relative errors indicated in the plot ($\times$). The fits of the results are depicted with lines in accordance with the legend. The random error of the reference result, $f_{c,\mathbf{ref}}$, is depicted as a horizontal line limiting its 95 % confidence interval.

method of Ref. [28] is 3-6 faster than usual adaptive RK4/5 integration. Thus, when tracing fusion alpha particles for 0.01 s the computational efficiency of the quasi-geometric integration method with sufficient accuracy is comparable to the fastest known methods for this purpose, despite its inherent deficiency of chaos induced by the piecewise linearization of the electromagnetic field.

Chapter 8

Summary, conclusions and outlook

The content of this chapter including the figures can partially also be found in section V of Ref. [1] formulated by the author and including some minor modifications.

"A quasi-geometric integration method for guiding-center orbits in general three-dimensional toroidal fields has been developed, implemented and presented here. This orbit integration procedure is based on a representation of the electromagnetic field by continuous piecewise linear functions using a spatial mesh."[1] The implementation has been made publicly available on GitHub as scientific open source software with the name **G**uiding-center **OR**bit **I**ntegration with **L**ocal **L**inearization **A**pproach (**GORILLA**) under the MIT license.

"Collisionless particle orbits in real space and magnetic coordinates and their respective invariants of motion have been studied in detail for axisymmetric and non-axisymmetric geometries. Due to the special formulation of the guiding-center equations, the magnetic moment and the total energy are conserved naturally. In the case of toroidal axisymmetry the canonical toroidal angular momentum is accurately preserved by the quasi-geometric method in third and fourth order series expansion over the orbit parameter. This is also true for the parallel adiabatic invariant. Thus, the property of such systems to ideally confine the orbits is retained. In order to evaluate the limitations of these confinement properties, the kinetic energy of ions was chosen to be 300 keV. Otherwise systematic ODE integration errors originating from truncating the series expansion already at the second order would not be visible, since this error is proportional to the particle energy.

For passing orbits in 3D fields, however, the piecewise linearization of the electromagnetic field introduces some artificial chaotic diffusion which, nevertheless, could be made negligibly small by spatial grid refinement. For trapped orbits in 3D fields, the approximate conservation of the parallel adiabatic invariant is not violated by significant error accumulation.

To assess the method's performance, the mono-energetic radial transport coefficient, D_{11}, which gives a main contribution to neoclassical transport, has been evaluated

for a quasi-isodynamic reactor-scale stellarator field[39] using the Monte Carlo method. For both, electrons and ions, the results obtained by quasi-geometric orbit integration are in good agreement with the results of adaptive RK4/5 integration with usual spline interpolation of electromagnetic fields. In the performance benchmark, we observe that the quasi-geometric orbit integration method with the polynomial series solution truncated at $K = 2$ is the fastest sufficiently accurate tested method to compute D_{11}. For the case of D ions with an energy of 3 keV the guiding-center orbit integration is one order of magnitude faster than 4th order Runge-Kutta integration in splined fields. Here, truncating the series expansion at the second order is not necessarily contradicting the result of collisionless guiding-center orbits, where a second order truncation leads to visible systematic errors at high kinetic energies. An appropriate choice of grid size and series expansion order strongly depends on the physical application, the kinetic particle energy and the complexity of the magnetic field (e.g. number of harmonic modes)."[1]

Furthermore, the quasi-geometric integration method has been also applied to the computation of fusion alpha particle losses in a realistic stellarator configuration. "It has been shown that the inherent artificial chaos of the method, which is induced by the linearization of the electromagnetic field, strongly scales with the particle's Larmor radius. Consequently, the quasi-geometric integration method is not suited to be used for tracing of high-energetic 3.5 MeV fusion alpha particles for the slowing-down time of 1 s. Nevertheless, for the trace time of 0.01 s the confined fraction results computed with the quasi-geometric integration method and an appropriate choice of the grid size lie within the 95 % confidence interval of the reference computation while showing a significant CPU speed-up. In particular, the fastest sufficiently accurate method is GORILLA with the solution in form of a polynomial series truncated at $K = 2$ and an angular grid size of $N_\vartheta \times N_\varphi = 100 \times 100$. This method is roughly 3 times faster than the fastest reference method, namely the adaptive RK4/5 integrator with a relative tolerance of 10^{-6}. The results for the confined fraction of this comparatively short trace time could be used for an early classification into regular and chaotic orbit types within fusion alpha loss computations."[3]

"For the application in global kinetic computations no extra effort is needed to obtain dwell times within spatial grid cells as these are computed automatically in the present approach. Additionally, integrals of velocity powers over these dwell times are available as analytical expressions. The latter quantities are required for statistical scoring of orbits in Monte Carlo computations of macroscopic parameters, such as plasma response currents and charges caused by external non-axisymmetric perturbations in tokamaks or parameters of the edge plasma in devices with 3D field geometry.

Moreover, similarly to the geometric integrator for axisymmetric two-dimensional fields described in Ref. [35], the presented method is less sensitive to noise in the electromagnetic field than procedures relying upon high order polynomial interpolation. These characteristics suggest additional overall performance enhancements in both numerical stability and computational efficiency, when the quasi-geometric orbit integration is applied to kinetic modeling.

The applicability of the inherently low-order method to particle-in-cell turbulence computations, where higher order schemes produce smoother solutions, is still an open question. Finally, it should be mentioned that the method facilitates the coupling with kinetic neutral particle codes such as EIRENE[74], where one needs to model particle conversion into neutrals and back with plasma and neutral particles described in different coordinate systems. The necessary transformation of coordinates does not require solving any implicit dependencies (nonlinear equations), since that is a linear operation in this approach and is therefore intrinsically fast."[1]

Due to its well-suited properties, GORILLA is part of the "EUROfusion Theory, Simulation, Validation and Verification Task for Impurity Sources, Transport, and Screening" where it is planned to be coupled to EIRENE and applied for the kinetic modelling of the impurity ion component. Furthermore, in the "EUROfusion Theory, Simulation, Validation and Verification Task for Stellarator Optimization" GORILLA is considered to support fusion alpha loss computations. Chapter 7 of this thesis is already a preliminary work for this task.

In addition, a grant from the Austrian Marshall Plan Foundation was awarded to the author for the proposed research project of "Global Monte Carlo computations of parallel equilibrium current density and other macroscopic parameters in an axisymmetric tokamak using the guiding-center orbit tracing code GORILLA" which shall be conducted at the Princeton Plasma Physics Laboratory (PPPL), Princeton University.

Bibliography

[1] M. Eder, C. G. Albert, L. M. P. Bauer, S. V. Kasilov, and W. Kernbichler, "Quasi-geometric integration of guiding-center orbits in piecewise linear toroidal fields," *Physics of Plasmas*, vol. 27, p. 122508, Dec. 2020. Publisher: American Institute of Physics.

[2] M. Eder, C. G. Albert, L. M. P. Bauer, S. V. Kasilov, and W. Kernbichler, "Gorilla: Guiding-center orbit integration with local linearization approach," *Public GitHub repository:* `https://github.com/itpplasma/GORILLA`; *submitted to Journal of Open Source Software; Preprint:* `https://github.com/openjournals/joss-papers/blob/joss.03116/joss.03116/10.21105.joss.03116.pdf` *[29 April 2021]*, 2021.

[3] M. Eder, C. G. Albert, S. V. Kasilov, W. Kernbichler, and M. Scheidt, "Integration of the guiding-center equations in toroidal fields utilizing a local linearization approach," *47th EPS Conference on Plasma Physics*, p. P3.1059, 2021.

[4] M. Eder, C. G. Albert, L. M. P. Bauer, S. V. Kasilov, and W. Kernbichler, "Three-dimensional geometric integrator for charged particle orbits in toroidal fusion devices," *46th EPS Conference on Plasma Physics*, vol. ECA Vol. 43C, p. P5.1100, 2019.

[5] R. G. Littlejohn, "Variational principles of guiding centre motion," *Journal of Plasma Physics*, vol. 29, pp. 111–125, Feb. 1983.

[6] A. H. Boozer, "Guiding center drift equations," *The Physics of Fluids*, vol. 23, pp. 904–908, May 1980.

[7] J. R. Cary and A. J. Brizard, "Hamiltonian theory of guiding-center motion," *Reviews of Modern Physics*, vol. 81, pp. 693–738, May 2009.

[8] A. H. Boozer and G. Kuo-Petravic, "Monte Carlo evaluation of transport coefficients," *The Physics of Fluids*, vol. 24, pp. 851–859, May 1981.

[9] W. Lotz and J. Nührenberg, "Monte Carlo computations of neoclassical transport," *Phys. Fluids*, vol. 31, no. 10, p. 2984, 1988.

[10] J. A. Heikkinen, T. P. Kiviniemi, T. Kurki-Suonio, A. G. Peeters, and S. K. Sipilä, "Particle Simulation of the Neoclassical Plasmas," *Journal of Computational Physics*, vol. 173, pp. 527–548, Nov. 2001.

[11] A. Wakasa, S. Murakami, and S.-i. Oikawa, "Study of Neoclassical Transport in LHD Plasmas by Applying the DCOM/NNW Neoclassical Transport Database," *Plasma and Fusion Research*, vol. 3, pp. S1030–S1030, 2008.

[12] V. Tribaldos, "Monte Carlo estimation of neoclassical transport for the TJ-II stellarator," *Phys. Plasmas*, vol. 8, no. 4, p. 1229, 2001.

[13] M. Y. Isaev, S. Brunner, W. A. Cooper, T. M. Tran, A. Bergmann, C. D. Beidler, J. Geiger, H. Maassberg, J. Nührenberg, and M. Schmidt, "VENUS+δf: A Bootstrap Current Calculation Module for 3-D Configurations," *Fusion Science and Technology*, vol. 50, pp. 440–446, Oct. 2006.

[14] K. Allmaier, S. V. Kasilov, W. Kernbichler, and G. O. Leitold, "Variance reduction in computations of neoclassical transport in stellarators using a δf method," *Physics of Plasmas*, vol. 15, p. 072512, July 2008.

[15] S. Satake, J.-K. Park, H. Sugama, and R. Kanno, "Neoclassical Toroidal Viscosity Calculations in Tokamaks Using a δ f Monte Carlo Simulation and Their Verifications," *Phys. Rev. Lett.*, vol. 107, p. 055001, July 2011.

[16] D. Pfefferlé, W. A. Cooper, J. P. Graves, and C. Misev, "VENUS-LEVIS and its spline-Fourier interpolation of 3D toroidal magnetic field representation for guiding-centre and full-orbit simulations of charged energetic particles," *Computer Physics Communications*, vol. 185, pp. 3127–3140, Dec. 2014.

[17] S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T. M. Tran, B. F. Mcmillan, O. Sauter, K. Appert, Y. Idomura, and L. Villard, "A global collisionless PIC code in magnetic coordinates," *Computer Physics Communications*, vol. 177, pp. 409–425, Sept. 2007.

[18] A. Mishchenko, R. Hatzky, and A. Könies, "Global particle-in-cell simulations of Alfvénic modes," *Physics of Plasmas*, vol. 15, p. 112106, Nov. 2008.

[19] R. Hager and C. S. Chang, "Gyrokinetic neoclassical study of the bootstrap current in the tokamak edge pedestal with fully non-linear Coulomb collisions," *Physics of Plasmas*, vol. 23, p. 042503, Apr. 2016.

[20] S. Ku, C. S. Chang, R. Hager, R. M. Churchill, G. R. Tynan, I. Cziegler, M. Greenwald, J. Hughes, S. E. Parker, M. F. Adams, E. D'Azevedo, and P. Worley, "A fast low-to-high confinement mode bifurcation dynamics in the boundary-plasma gyrokinetic code XGC1," *Physics of Plasmas*, vol. 25, p. 056107, Apr. 2018.

[21] C. G. Albert, M. F. Heyn, S. V. Kasilov, W. Kernbichler, A. F. Martitsch, and A. M. Runov, "Kinetic modeling of 3D equilibria in a tokamak," *J. Phys.: Conf. Ser.*, vol. 775, p. 012001, Nov. 2016.

[22] R. Hager, C. S. Chang, N. M. Ferraro, and R. Nazikian, "Gyrokinetic study of collisional resonant magnetic perturbation (RMP)-driven plasma density and heat transport in tokamak edge plasma using a magnetohydrodynamic screened RMP field," *Nucl. Fusion*, vol. 59, p. 126009, Sept. 2019.

[23] A. I. Morozov and L. S. Solov'ev, "The Structure of Magnetic Fields," *Reviews of Plasma Physics*, vol. 2, p. 1, 1966.

[24] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. May 2006.

[25] P. J. Morrison, "Structure and structure-preserving algorithms for plasma physics," *Phys. Plasmas*, vol. 24, no. 5, p. 055502, 2017.

[26] R. Zhang, J. Liu, Y. Tang, H. Qin, J. Xiao, and B. Zhu, "Canonicalization and symplectic simulation of the gyrocenter dynamics in time-independent magnetic fields," *Phys. Plasmas*, vol. 21, no. 3, p. 032504, 2014.

[27] B. Zhu, Z. Hu, Y. Tang, and R. Zhang, "Symmetric and symplectic methods for gyrocenter dynamics in time-independent magnetic fields," *Int. J. Model. Simulation, Sci. Comput.*, vol. 07, no. 02, p. 1650008, 2016.

[28] C. G. Albert, S. V. Kasilov, and W. Kernbichler, "Symplectic integration with non-canonical quadrature for guiding-center orbits in magnetic confinement devices," *Journal of Computational Physics*, vol. 403, p. 109065, Feb. 2020.

[29] H. Qin and X. Guan, "Variational symplectic integrator for long-time simulations of the guiding-center motion of charged particles in general magnetic fields," *Phys. Rev. Lett.*, vol. 100, no. 3, p. 035006, 2008.

[30] H. Qin, X. Guan, and W. M. Tang, "Variational symplectic algorithm for guiding center dynamics and its application in tokamak geometry," *Phys. Plasmas*, vol. 16, no. 4, p. 042510, 2009.

[31] J. W. Burby and C. L. Ellison, "Toroidal regularization of the guiding center Lagrangian," *Phys. Plasmas*, vol. 24, no. 11, p. 110703, 2017.

[32] M. Kraus, "Projected variational integrators for degenerate Lagrangian systems," *arXiv:1708.07356*, 2017.

[33] C. L. Ellison, J. M. Finn, J. W. Burby, M. Kraus, H. Qin, and W. M. Tang, "Degenerate variational integrators for magnetic field line flow and guiding center trajectories," *Physics of Plasmas*, vol. 25, p. 052502, May 2018.

[34] J. Xiao and H. Qin, "Slow manifolds of classical Pauli particle enable structure-preserving geometric algorithms for guiding center dynamics," *arXiv:2006.03818 [physics]*, June 2020.

[35] S. V. Kasilov, A. M. Runov, and W. Kernbichler, "Geometric integrator for charged particle orbits in axisymmetric fusion devices," *Computer Physics Communications*, vol. 207, pp. 282–286, Oct. 2016.

[36] J. Squire, H. Qin, and W. M. Tang, "Geometric integration of the Vlasov-Maxwell system with a variational particle-in-cell scheme," *Physics of Plasmas*, vol. 19, p. 084501, Aug. 2012.

[37] J. Xiao, H. Qin, J. Liu, Y. He, R. Zhang, and Y. Sun, "Explicit high-order non-canonical symplectic particle-in-cell algorithms for Vlasov-Maxwell systems," *Physics of Plasmas*, vol. 22, p. 112504, Nov. 2015.

[38] M. Kraus, K. Kormann, P. J. Morrison, and E. Sonnendrücker, "GEMPIC: geometric electromagnetic particle-in-cell methods," *Journal of Plasma Physics*, vol. 83, p. 905830401, Aug. 2017.

[39] M. Drevlak, C. D. Beidler, J. Geiger, P. Helander, and Y. Turkin, "Quasi-isodynamic Configuration with Improved Confinement," *41st EPS Conference on Plasma Physics*, vol. ECA Vol. 38F, p. P1.070, 2014.

[40] L. Landau and E. Lifschitz, *Lehrbuch der Theoretischen Physik. Band 1: Mechanik*. Akademie Verlag Berlin, 1984.

[41] H. Goldstein, *Classical Mechanics*. Pearson Education, Sept. 2002.

[42] H. Alfvén, "On the motion of a charged particle in a magnetic field," *Ark. Mat., Astron. Fys.*, no. 27A, 1940.

[43] C. G. Albert, *Guiding-center dynamics*. Plasmaphysik Skriptum, Technische Universität Graz, 2020.

[44] M. Eder, *Three-dimensional geometric integrator for charged particle orbits in toroidal fusion devices*. Master's Thesis, Technische Universität Graz, 2018.

[45] C. Grebogi and R. G. Littlejohn, "Relativistic ponderomotive Hamiltonian," *The Physics of Fluids*, vol. 27, pp. 1996–2004, Aug. 1984.

[46] M. Eder, C. G. Albert, L. M. P. Bauer, S. V. Kasilov, W. Kernbichler, and M. Meisterhofer, "GORILLA: Guiding-center ORbit Integration with Local Linearization Approach," Mar. 2021.

[47] L. Bauer, *Field-aligned mesh generation and enhancements of the guiding-center orbit code GORILLA*. Master's Thesis, Technische Universität Graz, 2020.

[48] L. L. Lao, H. S. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer, "Reconstruction of current profile parameters and plasma shapes in tokamaks," *Nuclear Fusion*, vol. 25, pp. 1611–1622, Nov. 1985. Publisher: IOP Publishing.

[49] L. L. Lao, J. R. Ferron, R. J. Groebner, W. Howl, H. S. John, E. J. Strait, and T. S. Taylor, "Equilibrium analysis of current profiles in tokamaks," *Nuclear Fusion*, vol. 30, pp. 1035–1049, June 1990. Publisher: IOP Publishing.

[50] S. P. Hirshman and J. C. Whitson, "Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria," *The Physics of Fluids*, vol. 26, pp. 3553–3568, Dec. 1983. Publisher: American Institute of Physics.

[51] S. P. Hirshman, W. I. van RIJ, and P. Merkel, "Three-dimensional free boundary calculations using a spectral Green's function method," *Computer Physics Communications*, vol. 43, pp. 143–155, Dec. 1986.

[52] J.-k. Park, A. H. Boozer, and A. H. Glasser, "Computation of three-dimensional tokamak and spherical torus equilibria," *Physics of Plasmas*, vol. 14, p. 052110, May 2007. Publisher: American Institute of Physics.

[53] N. C. Logan, J.-K. Park, K. Kim, Z. Wang, and J. W. Berkery, "Neoclassical toroidal viscosity in perturbed equilibria with general tokamak geometry," *Physics of Plasmas*, vol. 20, p. 122507, Dec. 2013. Publisher: American Institute of Physics.

[54] W. D. D'haeseleer, W. N. G. Hitchon, J. D. Callen, and J. L. Shohet, *Flux Coordinates and Magnetic Field Structure: A Guide to a Fundamental Tool of Plasma Theory*. Dec. 2012.

[55] H. E. Mynick, "Transport optimization in stellarators," *Physics of Plasmas*, vol. 13, p. 058102, May 2006.

[56] N. Flocke, "Algorithm 954: An Accurate and Efficient Cubic and Quartic Equation Solver for Physical Applications," *ACM Transactions on Mathematical Software*, vol. 41, pp. 30:1–30:24, Oct. 2015.

[57] M. F. Heyn, I. B. Ivanov, S. V. Kasilov, W. Kernbichler, P. Leitner, V. V. Nemov, and W. S. and, "Quasilinear modelling of RMP interaction with a tokamak plasma: application to ASDEX Upgrade ELM mitigation experiments," *Nucl. Fusion*, vol. 54, p. 064005, May 2014.

[58] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, pp. 617–644, Apr. 1928. Conference Name: Transactions of the American Institute of Electrical Engineers.

[59] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[60] J. L. Velasco, K. Allmaier, A. López-Fraguas, C. D. Beidler, H. Maassberg, W. Kernbichler, F. Castejón, and J. A. Jiménez, "Calculation of the bootstrap current profile for the TJ-II stellarator," *Plasma Phys. Control. Fusion*, vol. 53, p. 115014, Nov. 2011.

[61] S. Matsuoka, S. Satake, R. Kanno, and H. Sugama, "Effects of magnetic drift tangential to magnetic surfaces on neoclassical transport in non-axisymmetric plasmas," *Physics of Plasmas*, vol. 22, p. 072511, July 2015.

[62] B. Huang, S. Satake, R. Kanno, H. Sugama, and S. Matsuoka, "Benchmark of the local drift-kinetic models for neoclassical transport simulation in helical plasmas," *Phys. Plasmas*, vol. 24, p. 022503, Feb. 2017.

[63] J. L. Velasco, I. Calvo, F. I. Parra, and J. M. García-Regaña, "KNOSOS: A fast orbit-averaging neoclassical code for stellarator geometry," *Journal of Computational Physics*, vol. 418, p. 109512, Oct. 2020.

[64] C. Beidler, K. Allmaier, M. Isaev, S. Kasilov, W. Kernbichler, G. Leitold, H. Maaßberg, D. Mikkelsen, S. Murakami, M. Schmidt, D. Spong, V. Tribaldos, and A. Wakasa, "Benchmarking of the mono-energetic transport coefficients—results from the International Collaboration on Neoclassical Transport in Stellarators (ICNTS)," *Nucl. Fusion*, vol. 51, p. 076001, July 2011.

[65] A. H. Boozer, "Curl-free magnetic fields for stellarator optimization," *Physics of Plasmas*, vol. 26, p. 102504, Oct. 2019. Publisher: American Institute of Physics.

[66] P. Helander, "Theory of plasma confinement in non-axisymmetric magnetic fields (2014 *Rep. Prog. Phys* . **77** 087001)," *Reports on Progress in Physics*, vol. 81, p. 099501, Sept. 2018.

[67] L.-M. Imbert-Gerard, E. J. Paul, and A. M. Wright, "An Introduction to Stellarators: From magnetic fields to symmetries and optimization," *arXiv:1908.05360 [physics]*, Aug. 2020. arXiv: 1908.05360 version: 2.

[68] C. G. Albert, S. V. Kasilov, and W. Kernbichler, "Accelerated methods for direct computation of fusion alpha particle losses within, stellarator optimization," *Journal of Plasma Physics*, vol. 86, p. 815860201, Apr. 2020.

[69] V. V. Nemov, S. V. Kasilov, W. Kernbichler, and G. O. Leitold, "Poloidal motion of trapped particle orbits in real-space coordinates," *Physics of Plasmas*, vol. 15, p. 052501, May 2008. Publisher: American Institute of Physics.

[70] V. V. Nemov, S. V. Kasilov, W. Kernbichler, and G. O. Leitold, "The ∇B drift velocity of trapped particles in stellarators," *Physics of Plasmas*, vol. 12, p. 112507, Nov. 2005. Publisher: American Institute of Physics.

[71] W. Lotz, P. Merkel, J. Nuhrenberg, and E. Strumberger, "Collisionless alpha -particle confinement in stellarators," *Plasma Physics and Controlled Fusion*, vol. 34, pp. 1037–1052, June 1992. Publisher: IOP Publishing.

[72] V. V. Nemov, S. V. Kasilov, and W. Kernbichler, "Collisionless high energy particle losses in optimized stellarators calculated in real-space coordinates," *Physics of Plasmas*, vol. 21, p. 062501, June 2014. Publisher: American Institute of Physics.

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

[74] D. Reiter, M. Baelmans, and P. Börner, "The EIRENE and B2-EIRENE Codes," *Fusion Science and Technology*, vol. 47, pp. 172–186, Feb. 2005.