Department of Mathematics
TUM School of Computation, Information and Technology
Technical University of Munich

Adding a Gyrokinetic Turbulence Model with Adiabatic Electrons in Struphy

Ipek Akbatur

Thesis for the attainment of the academic degree

Master of Science

at the TUM School of Computation, Information and Technology of the Technical University of Munich

Supervisor:

Prof. Dr. Eric Sonnendrücker

Advisor:

Dr. Stefan Possanner

Submitted:

Munich, 29. May 2024

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated.
I have only used the resources given in the list of references.

Munich, 29. May 2024 Ipek Akbatur

Acknowledgments

Firstly, I want to thank my advisor, Stefan Possaner, for his support and patience throughout the whole procedure. This thesis is written by guidance of him. I feel thankful for writing my master's thesis in the Max Planck Institute of Plasma Physics.

Secondly, I want to thank my family in Turkey and friends who are part of my family here for always being by my side.

To mom and dad

Abstract

This thesis focuses on the discretization and implementation of the Poisson and a gyrokinetic turbulence model with adiabatic electrons within the Struphy framework. Struphy is an open source Python package to solve plasma physics PDEs. Models are discretized and implemented with the finite element exterior calculus (FEEC) and particle-in-cell (PIC) methods. Initially, we discretize the Poisson equation using splines demonstrating high-order accuracy and computational efficiency. The core of the thesis involves the gyrokinetic model, which describes the behavior of charged particles in a magnetized plasma. We derive the equations of motion and ensure that the discretized system respects the fundamental symmetries and semi-discrete energy conservation. Furthermore, we validate the gyrokinetic model through a Landau damping test, confirming its capability to accurately capture essential plasma phenomena. The results highlight the effectiveness of the Hamiltonian discretization approach and the practical applicability of the implemented numerical methods in plasma physics.

Contents

1 Introduction

The use of plasma in daily life has become typical which includes examples such as neon tubes and plasma displays. There are also several industrial applications: amplifiers in telecommunications satellites, plasma etching in microelectronics, production of X-rays. These technological uses demonstrate how plasma can be applied to a wide range of applications. Furthermore, plasma research is driving innovation and advancement in many industries. For example, the uses of plasma technology in medicine are examined, and new treatment methods are developed. In addition, the potential of plasma in energy production is also being investigated. Therefore, plasma research plays an important role in both daily life and industrial applications.

Since the main topic in this thesis is to discretize and implement a kinetic plasma model, let us dive into what a kinetic model is. In a kinetic model, each particle species s in the plasma is characterized by a distribution function $\mathbf{f}_s(x, v, t)$. Phase space consists of the subspace of $\mathbb{R}^6$ containing all possible positions and velocities of the particles. For any given volume V, $\int_V \mathbf{f}_s d\mathbf{x} d\mathbf{v}$ yields the average number of particles of species s whose positions and velocities fall within V. To define the likelihood of finding a particle of species s at a particular point (x,v) in phase space, density function f is normalized to unity, and becomes a probability density function. Instead of determining the exact path of each particle, it may be more convenient to understand the probability of a particle being found in a particular region of the (x, v)-phase space. The corresponding probability distribution function is called $f_s(t, x, v)$ for type "s" and is often called the "distribution function". This function satisfies the Vlasov equation:

$$\frac{\partial f_s}{\partial t} + v \cdot \nabla f_s + \frac{q_s}{m_s}(E + v \times B) \cdot \frac{\partial f_s}{\partial v} = 0, \quad f_s(0) = f_{s0}(x, v). \tag{1.1}$$

This equation is a transport equation in (x, v)-phase space and the initial condition is satisfied by $f_{s0}(x, v)$. f_s satisfies the above equation because it is constant along a path, and this is found by taking the time derivative of f_s at a point (x, v) in phase space. The kinetic model that will be introduced as a gyrokinetic model mainly satisfies similar equations with some differences.

Gyrokinetics is a fundamental field of research in the study of fluctuations in magnetized plasmas. The importance of gyrokinetics originates from the wide range of time and space scales available in many laboratory and space plasma configurations. For instance, large tokamaks like ITER are built to study

fusion plasmas, and one of the main causes of unexpected heat and particle movement in these facilities is turbulence caused by ion temperature gradients (ITG). Since the phase space is generally six-dimensional, this is still too expensive for real-world problems, even as supercomputers are reaching exascale flops. This is due to the multiscale nature of plasma dynamics, which involves space-time scales over many orders of magnitude and requires high resolution.

Our main topic concerns the so-called "gyrokinetic equations", which require that the characteristic frequency to be small relative to the ion gyrofrequency and that the average spatial scale of the fluctuations perpendicular to the magnetic field. These fluctuations have frequencies much lower than the ion gyrocyclic frequency and constitute the main formalism often used in analytical and numerical descriptions of low-frequency micro turbulence. We will use this guiding center motion in gyrokinetic theory, which is the effect of an external force on the rotational motion of a particle. We will apply this idea to motion in magnetic fields. For this purpose, we assume that the real particle trajectory can be decomposed into a circular orbit around a local guiding and the drift motion of the guiding center. Net force for the drift motion, which is averaged over a gyroperiod. This net force is then converted into an equivalent electric field – as in the case of gravitational driving – and the drift velocity. The implication is that the the magnetic field variation along a gyro-radius should be small compared to the magnitude of the magnetic field at the guiding center. [11]

After discretization and finding the characteristics of the motion, another step is to implement that theory into an application using the Python package Struphy (structure-preserving hybrid codes). It is developed at the Max Planck Institute for Plasma Physics to find solutions for plasma physics PDEs. It has a wide range of models, not only kinetic but also fluid, hybrid, and toy. The first model introduced is added to the toy models, and the second one is added to the kinetic models. In Struphy, we use geometric finite elements (FEEC) and particle-in-cell method (PIC) frameworks for the PDE discretizations.

2 Preliminaries

2.1 Finite Element Exterior Calculus

The theory of Finite Element Exterior Calculus includes utilizing the Finite Element method to approximate solutions to Partial Differential Equations. Firstly, the PDE undergoes a variational reformulation by rewriting it as a minimization problem. It actually means multiplying it by a test function and integrating it by parts to have a well-defined solution within a chosen function space. Following this, a finite-dimensional approximation of the function space is established, often employing simple functions like piecewise polynomials. These approximations replace the functions within the infinite-dimensional space in the variational formulation. In Finite Element methods, the fundamental approach to approximating a function $u \in V$ in a space $\mathbf{V}$ is

$$u \approx u_h = \sum_{i=1}^{M} u_i \Lambda_i \quad \in V_h \ . \tag{2.1}$$

Here, $\Lambda_i \in V$ represent linearly independent basis functions, $(u_i)_i \in \mathbb{R}^M$ denote coefficients and $M \in \mathbb{N}$ indicates the dimension of the subspace $V_h \subset V$ spanned by the basis functions. An advantageous feature of the Finite Element method is that $u_h \in V$, facilitates nice analysis of the method. To ensure the method is stable and well-defined, selecting compatible Finite Element spaces is crucial. This selection involves choosing Finite Element spaces that define a discrete de Rham complex, where each space is a subset of its corresponding continuous space. In conventional vector calculus terminology, this requirement translates to selecting Finite Element spaces that form a discrete analogue of the continuous setting.

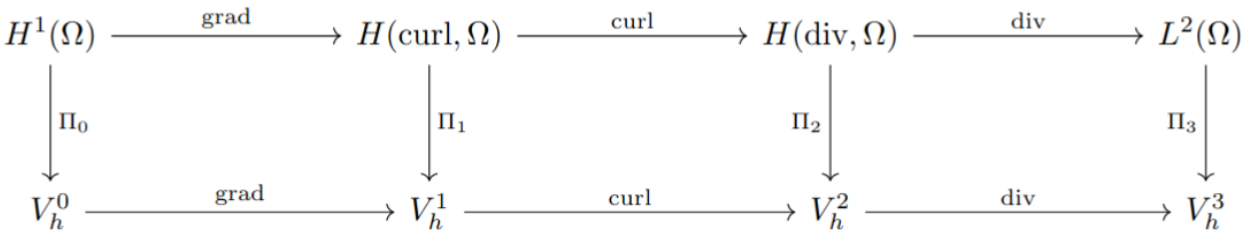

Figure 2.1 3D Derham Diagram

In the FEEC theory, the commuting diagram serves as an important element for assuring the stability and convergence of the Finite Element approximation. At each node of this diagram, the kernel of the succeeding operator corresponds to the image of the preceding one:

$$Im(grad) = Ker(curl), \;\; Im(curl) = Ker(div).$$

The lower row of the diagram establishes an exact sequence of finite-dimensional function spaces. Additionally, the diagram commutes at every level for any function $\phi \in H^1(\Omega)$,

$$\Pi_1(grad\phi) = grad(\Pi_0\phi). \tag{2.2}$$

Finite Element method is typically constructed element by element, which are commonly triangles or quadrilaterals in 2D and tetrahedra or cubes in 3D. A Finite Element approximation is characterized by a finite-dimensional function space with dimension N and N degrees of freedom, which uniquely describes a function within the finite-dimensional function space. [7]

The next step is to choose the basis function for finite element approximations. These functions are usually piece-wise polynomials such as Legendre or Chebyshev however, they have global support, which causes computational challenges since they lead to full matrices in the linear system. In order to solve this, problem B-splines basis functions are commonly used in finite element approximations and in Struphy.

To define B-splines main need is a non-decreasing sequence of one-dimensional knots $T = (\xi_1, \xi_2, .., \xi_{n+p+1})$ where n is the number of B spline functions and p is the order of the polynomial. Then B-splines are given by the formula as stated in [9] :

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi).$$

B-splines have compact support, and if there is more than one knot at the same point then that knot has multiplicity m, number of knots at that point. The derivatives of the functions are represented in

terms of B-spline lower-order bases. For a given polynomial order p and knot vector T, the derivative of the nth B-spline basis function is given by

$$D_{i,p} = \frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi).$$

B-spline basis functions can be used in higher dimensions basis construction with tensor products.

$$\Lambda_h^0(\Omega) = \text{span}\{N_i^p(\xi_1) N_j^p(\xi_2) N_k^p(\xi_3) \mid 1 \le i \le N_1, 1 \le j \le N_2, 1 \le k \le N_3\} \tag{2.3}$$

$$\Lambda_h^1(\Omega) = \text{span}\left\{ \begin{pmatrix} D_i^p(\xi_1) N_j^p(\xi_2) N_k^p(\xi_3) \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ N_i^p(\xi_1) D_j^p(\xi_2) N_k^p(\xi_3) \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ N_i^p(\xi_1) N_j^p(\xi_2) D_k^p(\xi_3) \end{pmatrix} \right\} \tag{2.4}$$

$$\Lambda_h^2(\Omega) = \text{span}\left\{ \begin{pmatrix} N_i^p(\xi_1) D_j^p(\xi_2) D_k^p(\xi_3) \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ D_i^p(\xi_1) N_j^p(\xi_2) D_k^p(\xi_3) \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ D_i^p(\xi_1) D_j^p(\xi_2) N_k^p(\xi_3) \end{pmatrix} \right\} \tag{2.5}$$

$$\Lambda_h^3(\Omega) = \text{span}\{D_i^p(x_1) D_j^p(x_2) D_k^p(x_3) \mid 1 \le i \le N_1, 1 \le j \le N_2, 1 \le k \le N_3\} \tag{2.6}$$

This construction of taking the derivative of the respective element in each basis element gives us linearly independent and computationally cheaper functions, which is also used in Struphy implementations. Moreover, computations of B-spline basis functions are done in the unit cube in Struphy. To define how one can solve PDEs in a unit cube we first should focus a bit on differential forms since they play a crucial role both in the theory and application.

A differential p-form is what can be integrated over a p dimensional submanifold of $\mathbf{R}^n$. For instance, 1-forms are objects that can be integrated over lines and curves. The coordinate representation or components of a p-form are defined such that the integral appears independent of the metric. In our case, differential forms are used to transform the physical domain to the logical domain in order to have computational advantages. The logical domain used in Struphy is $(0,1)^3$. Let $\hat{\Omega} = (0,1)^3$. [4] [2] This transformation is done by a diffeomorphism from the actual mapping to the logical cube as follows :

$$F : \hat{\Omega} \to \Omega \qquad \boldsymbol{\eta} \mapsto F(\boldsymbol{\eta}) = \mathbf{x} \tag{2.7}$$

where Jacobian matrix is given by

$$DF : \hat{\Omega} \to \mathbb{R}^{3\times 3}$$
$$DF_{i,j} = \frac{\partial F_i}{\partial \eta_j}$$

also metric tensor $G : \hat{\Omega} \to \mathbb{R}^{3\times 3}$ and volume element $\sqrt{g} : \hat{\Omega} \to \mathbb{R}$ are given by

$$G = DF^{\top} DF \tag{2.8}$$
$$\sqrt{g} = |\det(DF)| \tag{2.9}$$

Then transformation rules of differential 0-1-2 and 3-forms are

$$\hat{\phi}(\boldsymbol{\eta}) := \phi(F(\boldsymbol{\eta})) \tag{2.10}$$
$$\hat{\mathbf{E}}^1(\boldsymbol{\eta}) := DF^{\top}(\boldsymbol{\eta})\, \mathbf{E}(F(\boldsymbol{\eta})) \tag{2.11}$$
$$\hat{\mathbf{E}}^2(\boldsymbol{\eta}) := \sqrt{g}(\boldsymbol{\eta})\, DF^{-1}(\boldsymbol{\eta})\, \mathbf{E}(F(\boldsymbol{\eta})) \tag{2.12}$$
$$\hat{\phi}^3(\boldsymbol{\eta}) := \sqrt{g}(\boldsymbol{\eta})\, \phi(F(\boldsymbol{\eta})) \tag{2.13}$$

where 0-forms and 3-forms are scalar valued 1 and 2-forms are vector valued functions. With the above definition, we can conclude that elements of H^1 are 0-form scalars and taking the gradient takes them to the 1-form because they are in $H(curl)$. Through this procedure one can define the needed finite element spaces and pull-back equations to the logical domain.

2.2 Hamilton's principle

After pulling the equations to the logical domain, the next job to do is to find equations of motion with the help of the Hamiltonian structure. Hamiltonian is defined by the Lagrangian function. Lagrangian is the difference of the systems' kinetic and potential energy in general. To understand Hamilton's structure and least action principle we first introduce action functional which is defined by

$$S(\mathbf{u}) = \int_{\Omega} L(\mathbf{x}, \mathbf{u}^{(n)}(\mathbf{x})) \tag{2.14}$$

where $L(\mathbf{x}, \mathbf{u}^{(n)}(\mathbf{x}))$ is the Lagrangian, $\Omega \in \mathbf{R}^p$ open subset of p-dimensional Euclidean space, $\mathbf{x} \in \Omega$ and $\mathbf{u} : \Omega \to \mathbf{R}^p$. Moreover, the derivative which is $\mathbf{u}^{(n)}$ is defined by

$$\mathbf{u}^{(n)} = \frac{\partial^n \mathbf{u}}{\partial x_1 ... \partial x_n}.$$

Definition 2.2.1 *Let* $\mathbf{u}$ *be some function* $\Omega \to \mathbf{R}^q$ *. Suppose* $\tilde{u}(x, \epsilon)$ *is a smooth function* $\Omega \times \mathbf{R} \to \mathbf{R}^q$ *such that* $\tilde{u}(x, 0) = \mathbf{u}(x)$ *. The variation of a function* $\mathbf{u}$ *defined as 2.9 is*

$$\delta\mathbf{u}(x) := \frac{d}{d\epsilon}\Big|_{\epsilon=0} \tilde{u}.$$

Definition 2.2.2 *Let* $\mathbf{u}$ *and* $\tilde{u}$ *as in Definition 2.3.1. The variation of a functional S as*

$$\delta S(\mathbf{u}) := \frac{d}{d\epsilon}\Big|_{\epsilon=0} S(\tilde{u}(x, \epsilon)). \tag{2.15}$$

Definition 2.2.3 *The variational derivative* $\frac{\delta S}{\delta u}$ *of a functional S as given in 2.9 is defined via the relation*

$$\delta S(u) = \int_\Omega \frac{\delta S}{\delta \mathbf{u}} \cdot \delta\mathbf{u} dx.$$

After learning about the variation derivative one can state Hamilton's least action principle. It is shown that correct motion $\mathbf{u}$ of a dynamical system satisfies

$$\delta S(\mathbf{u}) = 0,$$

where S is the action functional defined in 2.9. By this principle, one can derive Euler Lagrange equations which allows us to write true equations of motion. This derivation is done by calculating $\delta S(\mathbf{u}) = 0$ using the formula in 2.9 and 2.10. [10] Taking the derivative inside the integral and assuming variations vanish at the boundary, we can get

$$\frac{\delta S}{\delta \mathbf{u}} = \frac{\partial L}{\partial \mathbf{u}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{u}}}. \tag{2.16}$$

2.3 Particle-In-Cell Method

The Particle-In-Cell (PIC) simulation operates on a straightforward principle: it tracks the motion of each plasma particle and derives macro-quantities such as density and current density from their positions and velocities. Using field equations, the forces acting on the particles are determined. The term "Particle-in-Cell" refers to how these quantities are assigned to the simulation particles. In general, any numerical simulation model that simultaneously solves the equations of motion for N particles are derived from the particles' positions and velocities can be considered a PIC simulation. PIC simulations have diverse applications beyond plasma physics, including solid-state and quantum physics. However,

in plasma physics, PIC codes typically solve the particle equations of motion under the Newton–Lorentz force for the non-relativistic case:

$$\frac{d\mathbf{X_i}}{dt} = \mathbf{V_i} \qquad \frac{d\mathbf{V_i}}{dt} = \frac{q}{m}(\mathbf{E}(X_i) + \mathbf{v} \times \mathbf{B}(X_i)) \tag{2.17}$$

We will also introduce the general equations of motion to understand the importance of the PIC algorithm.

$$\dot{\mathbf{q}} = G(t, \mathbf{q}), \qquad \mathbf{q}(t_0) = \mathbf{q}_0 \tag{2.18}$$

where $t \in \mathbb{R}$ $\mathbf{q}(t) \in \Omega \subset \mathbb{R}^n$ and $G : \Omega \times \mathbb{R} \to \mathbb{R}^n$ denotes the vector field in the ODE describing single particle motion in phase space. Moreover, flow map Φ of the initial value theorem satisfies :

$$\dot{\Phi} = G(t, \Phi) \qquad \Phi(t_0; t_0, \mathbf{q}_0) = \mathbf{q}_0. \tag{2.19}$$

This mapping propagates the equation from initial time t_0 to any time t. The flow map satisfies the Liouville theorem which helps us to construct discretization for the particle in cell method.

Theorem 2.3.1 *The flow of the differential equation (2.18) conserves volume. This implies if ρ is a solution of*

$$\frac{\partial \rho}{\partial t} + G.\nabla_q \rho = 0 \tag{2.20}$$

then

$$\int_{\Phi(\Omega)} \rho(t, \mathbf{q}) d\mathbf{q} = \int_{\Omega} \rho(t, \mathbf{y}) d\mathbf{y}. \tag{2.21}$$

The solution of the differential equation translates an initial condition $\mathbf{y} \in \mathbb{R}^n$to its value at time t. Then, with the help of the above theorem solution of the kinetic equation f can be expressed at phase space volume density $f^{\mathrm{vol}}(t, q)$ satisfies following transport equation

$$\partial_t f^{\mathrm{vol}} + \nabla \cdot (G(\mathbf{q}, t) f^{\mathrm{vol}}) = 0\,. \tag{2.22}$$

Then with the help of the PIC method, one can approximate the distribution function

$$f^{\mathrm{vol}} \approx f_h^{\mathrm{vol}}(t, q) = \frac{1}{N} \sum_{k=1}^{N} w_k\, \delta(q - q_k(t))\,, \tag{2.23}$$

where $w_k \in \mathbb{R}$ are particle weights.

By this approximation, PIC algorithm solves the characteristics of the kinetic equation according to (2.17) first discretizes and then computes, stores and updates weights for each time step and for each particle.

Since this algorithm is used for high number of particles, it has slow convergence to the exact solution. This problem is solved by noise reduction techniques, specifically with the control variate method which will be discussed in the implementation of the gyrokinetic model.

3 Poisson Model

The Poisson equation is a fundamental equation that is a guidance in various branches of physics, including gravitation, fluid models, and plasma physics. In plasma numerical simulations, solving the Poisson equation for the electric field is crucial therefore one can see the implementation of a Poisson solver in many models in Struphy. We'll now discretize and add a toy model Poisson to Struphy. The electric field serves as source terms coupled with the nonlinear flow equations, allowing accurate representation of the interaction with the density fields of charged particles. The Poisson equation connects the electromagnetic potential ϕ to the charge distribution ρ within this framework. [3]

3.1 Discretization

The discretization of the Poisson model serves as the initial step in introducing the concepts explored within this thesis. Let us consider strong Poisson equation :

$$-\Delta\Phi = \rho \tag{3.1}$$

$$-\nabla\Phi = \boldsymbol{E} \tag{3.2}$$

Where $\Phi \in H^1$, $E \in H(curl)$ and $\rho \in L^2$ is given. We can rewrite the Poisson equation with a $3x3$ matrix D which indicates the direction of the electric field. In the developed toy model perpendicular diffusion is used. We can rewrite (3.1) as:

$$-\nabla D\nabla\Phi = \rho$$

where $D \in R^{3x3}$ symmetric positive definite. We find the weak form of the Poisson equation by multiplying both sides with a test function which is also in H^1 and taking integral. Integral by parts gives the following:

$$\int_\Omega \nabla\Phi^T D\nabla\Psi d\eta = \int_\Omega \rho\Psi d\eta \qquad \forall\Psi \in H^1 \tag{3.3}$$

Where Ω is the physical domain. After having the weak form, we can discretize the equation using 3D Derham diagram. Since Φ and Ψ are in H^1 discrete Φ_h and Ψ_h are in V_h^0 by diagram. Then :

$$\Phi_h = \sum_{j=1}^{N_0} \Phi_j \Lambda_j^0 \tag{3.4}$$

$$\Psi_h = \sum_{i=1}^{N_0} \Psi_i \Lambda_i^0 \tag{3.5}$$

Where Φ_j and $\Psi_i \in \mathbb{R}$. Then adding discrete versions into the integral 2.3 gives

$$\sum_{i,j=1}^{N_0} \Phi_j \Psi_i \int_\Omega (\nabla \Lambda_j^0)^T D \nabla \Lambda_i^0 d\eta = \sum \Psi_i \int_\Omega \rho \Lambda_i^0 d\eta \tag{3.6}$$

Gradient of the discretized functions are:

$$\begin{aligned} \nabla \Phi_h &= \sum_{j=1}^{N_0} \Phi_j \nabla \Lambda_j^0 \\ &= \sum_{k=1}^{N_1} (G\vec{\Phi})_k \Lambda_k^1 \\ &= \sum_{k=1}^{N_1} \sum_{j=1}^{N_0} G_{k,j} \Phi_j \Lambda_k^1 \end{aligned}$$

$$\begin{aligned} \nabla \Psi_h &= \sum_{i=1}^{N_0} \Psi_j \nabla \Lambda_i^0 \\ &= \sum_{l=1}^{N_1} (G\vec{\Psi})_l \Lambda_l^1 \\ &= \sum_{l=1}^{N_1} \sum_{i=1}^{N_0} G_{l,j} \Psi_j \Lambda_l^1 \end{aligned}$$

Where $G \in \mathbb{R}^{N_1 x N_0}$.

Hence we combine (2.6) with gradients of discretized functions and get

$$\sum_{k,l=1}^{N_1} \sum_{i,j=1}^{N_0} G_{l,j} \Phi_j \int_\Omega \Lambda_l^1 D \Lambda_k^1 d\eta G_{k,i} \Psi_i = \sum G_{l,i} \Psi_i \int_\Omega \rho \Lambda_k^1 d\eta \tag{3.7}$$

Now, as described in the FEEC section we transform equation from physical domain to do logical domain by the transformation function $\mathbf{F}$ and get

$$\mathbb{M}_D^1 := \int_\Omega \Lambda_l^1 D \Lambda_k^1 d\eta = \int_{[0,1]^3} \Lambda_i^1(\eta) DF^{-1}(\eta) \hat{D}(\eta) DF^{-T} \Lambda_j^1(\eta) \sqrt{g} d^3x. \tag{3.8}$$

Therefore we get $\phi^T G^T \mathbb{M}_D^1 G \Psi = \rho \Psi$ and truncate it to get the final form :

$$\Phi^T G^T \mathbb{M}_D^1 G = \rho \tag{3.9}$$

where

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

3.2 Implementation and Convergence of the Model

Poisson model is added in toy models in Struphy which can be found here: `https://struphy.pages.mpcdf.de/struphy/sections/subsections/toy.html#struphy.models.toy.Poisson`. It is implemented with the help of the implicit diffusion propagator `https://struphy.pages.mpcdf.de/struphy/sections/subsections/propagators.html#struphy.propagators.propagators_fields.ImplicitDiffusion` which is designed to solve heat equations or specifically Poisson equation. This propagator is also changed and adapted later for the main gyrokinetic turbulance model for adiabatic electrons. Now this propagator solves $\phi \in H^1$

$$\int_\Omega \psi \, n_0(\mathbf{x}) \phi \, \mathrm{d}\mathbf{x} + \int_\Omega \nabla \psi^\top D_0(\mathbf{x}) \nabla \phi \, \mathrm{d}\mathbf{x} = \int_\Omega \psi \, \rho(t, \mathbf{x}) \, \mathrm{d}\mathbf{x} \qquad \forall \, \psi \in H^1 \,,$$

for our model where $n_0, \rho(t) : \Omega \to \mathbb{R}$ are real-valued functions $\rho(t)$ parametrized with time t and $D_0 : \Omega \to \mathbb{R}^{3\times3}$ is a positive diffusion matrix. Boundary terms from integration by parts are assumed to vanish.

Let's define convergence of the Poisson model. Let ϕ be the exact, ϕ_t be the numerical solutions, N be the number of elements in the grid and p is the spline degree. Then error of the numerical solution is defined as

$$||\phi - \phi_t|| \leq \frac{1}{N^p} \tag{3.10}$$

where p here is also the order of convergence. We expect the Poisson model to have an order of convergence p and this relation is implemented in Struphy. To assure that model works correctly different

tests have been done for various scenarios. One useful approach to assure the accuracy of a numerical method is to use a known analytical solution. This can be achieved by either applying a known solution to a specific scenario or by selecting a solution and constructing the problem around it. This technique is known as the method of manufactured solutions. For instance, with periodic boundary conditions, any periodic function can be chosen.We can determine the corresponding right-hand side by applying the operator, such as the Laplacian. The problem can be solved using this for various mesh sizes, allowing us to check the convergence order in a specified norm. For other boundary conditions, one can select a function that satisfies either homogeneous Dirichlet or Neumann boundary conditions. Alternatively, any smooth function can be chosen, and the boundary conditions can be determined according to the selected function. Regardless of the approach, it is crucial to remember to include the boundary conditions when defining the problem. In these tests of the Poisson model, convergence is tested firstly with $\mathbb{M}^1$ where

$$\mathbb{M}^1_{(\mu,ijk),(\nu,mno)} = \int \vec{\Lambda}^1_{\mu,ijk}\, G^{-1}\, \vec{\Lambda}^1_{\nu,mno} \sqrt{g}\, \mathrm{d}\boldsymbol{\eta}.$$

Moreover, they are tested in 1D and 2D with different boundary conditions such as Neumann Dirichlet and periodic. They are also tested with different mappings to see how accurate the solution is, for example with different sizes of Cuboid, Orthogonal and Collela mappings. The following figures are from the $\mathbb{M}^1$ test with orthogonal mapping and Neumann boundary condition. In our case, the cosine function is used. In order to make this test work, right hand side is firstly defined in physical domain

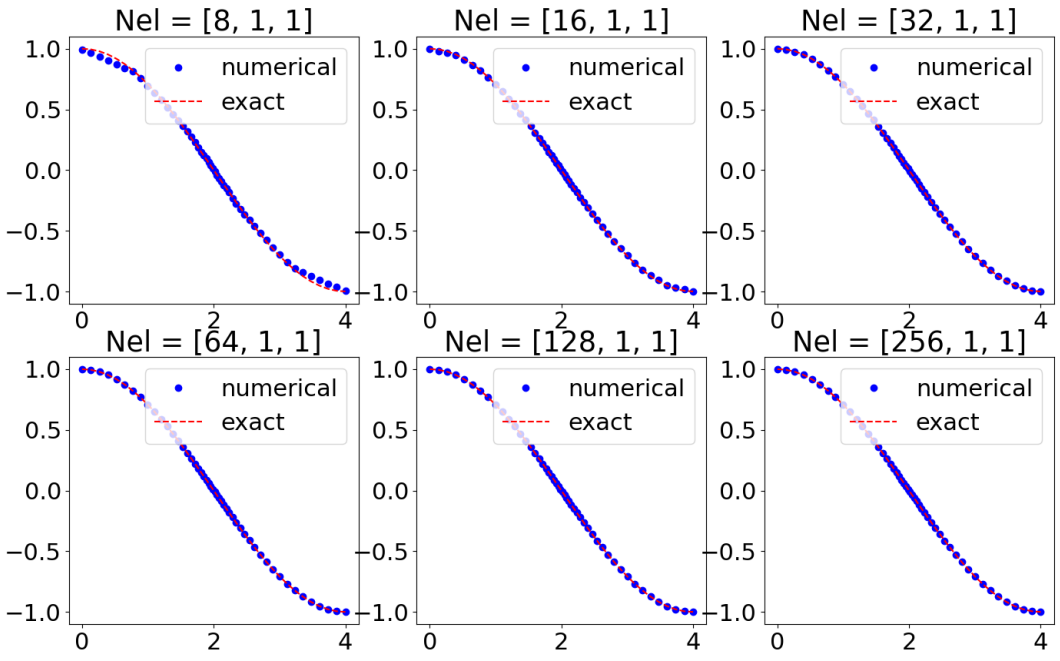

Figure 3.1 Cosine function with $\mathbb{M}^1$, $p = 1$.

and pulled back to the logical domain. In the above figure (3.1) one can see a cosine function in which how numerical and exact solutions behave when the number of elements increase. Increasing number of grid elements lead to a better approximation and if the approximation converges with a desired rate, it is so close to the exact function so that there is no change after some $N \gg 8$.

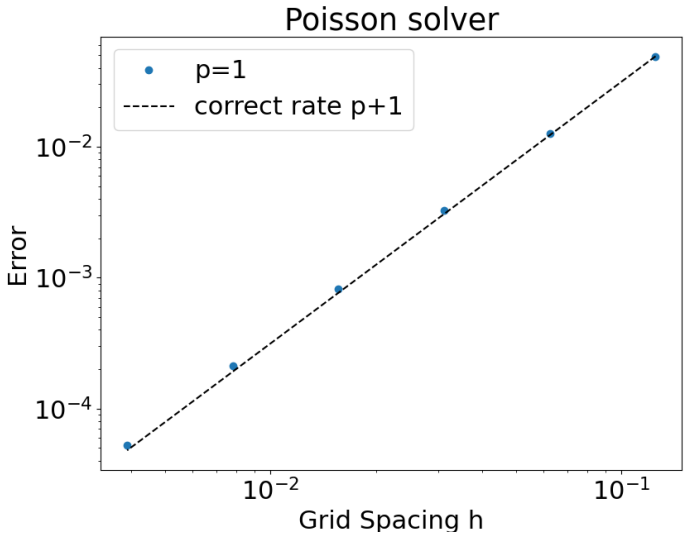

Figure 3.2 Convergence rate for spline degree 1 with $\mathbb{M}^1$.

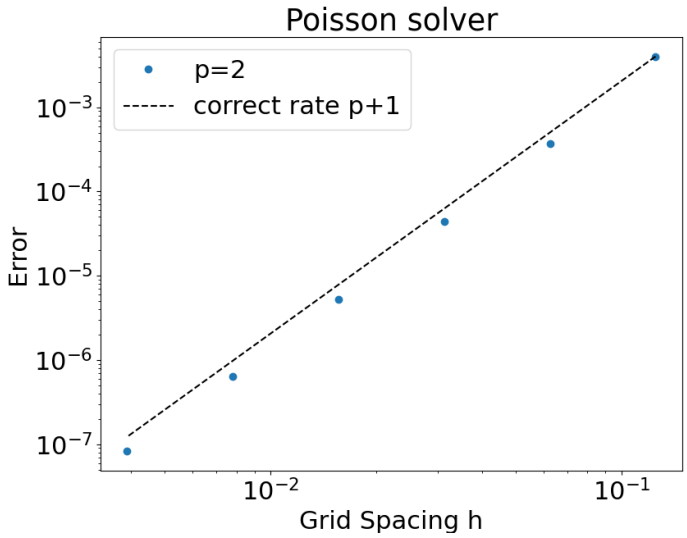

Figure 3.3 Convergence rate for spline degree 2 with $\mathbb{M}^1$.

Furthermore, test is done for spline degrees one two and three which can be seen from above and below figures. It can be deduced from the figures as well as from (3.10) that a more significant p implies higher convergence rate and more minor error.

Here we can see that the 1D Poisson model in Struphy with $\mathbb{M}^1$ converges with a rate $p + 1$ and this is

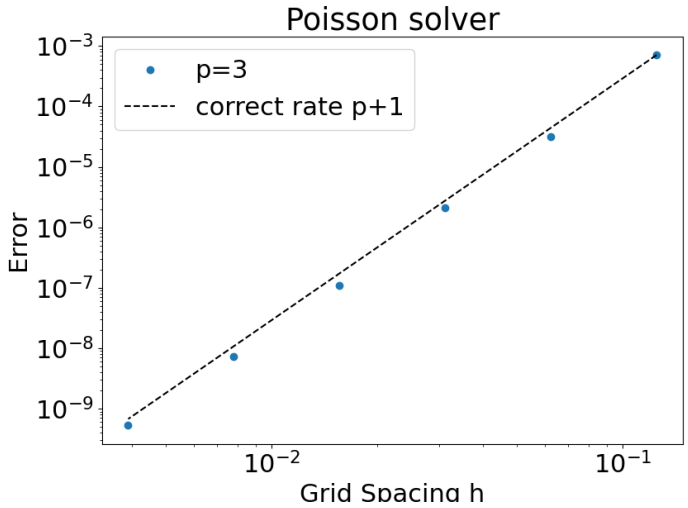

Figure 3.4 Convergence rate for spline degree 3 with $\mathbb{M}^1$.

higher than expected.

Secondly we pay attention to convergence with the use of $\mathbb{M}^1_D$ where

$$\mathbb{M}^1_{D(\mu,ijk),(\nu,mno)} = \int \vec{\Lambda}^1_{\mu,ijk}\, DF^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} DF^{-\top} \vec{\Lambda}^1_{\nu,mno} \sqrt{g}\, \mathrm{d}\boldsymbol{\eta}.$$

Differently, the periodic boundary condition is used this time and one can see similar behavior as $\mathbb{M}^1$ and same convergence rate $p+1$. Furthermore, in the results of this model clamped splines are used as a boundary condition which is are either 0 or 1 on the boundary. Clamped splines has N+p dimension whereas periodic splines has N where N is the number of elements. At the end 2D density plot of the numerical and exact solutions are presented.

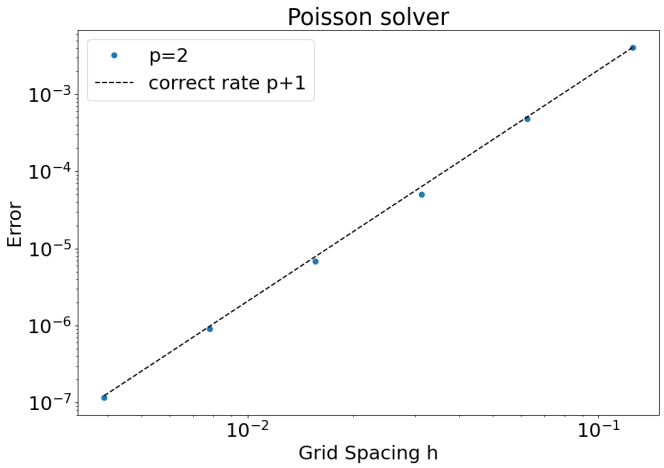

Figure 3.5 Convergence rate for spline degree 3 with $\mathbb{M}_D^1$

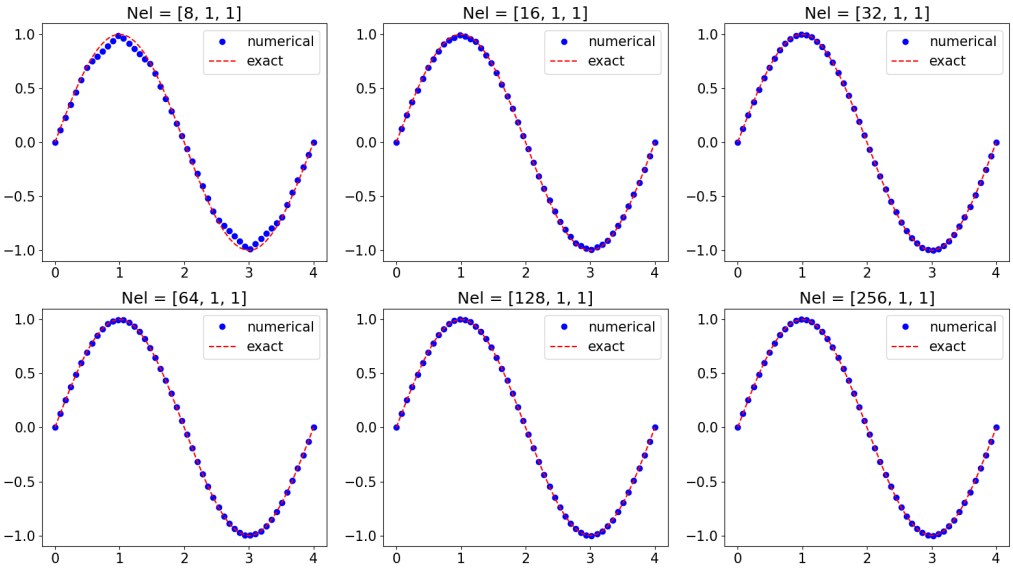

Figure 3.6 Cosine function with $\mathbb{M}_D^1$, $p = 1$.

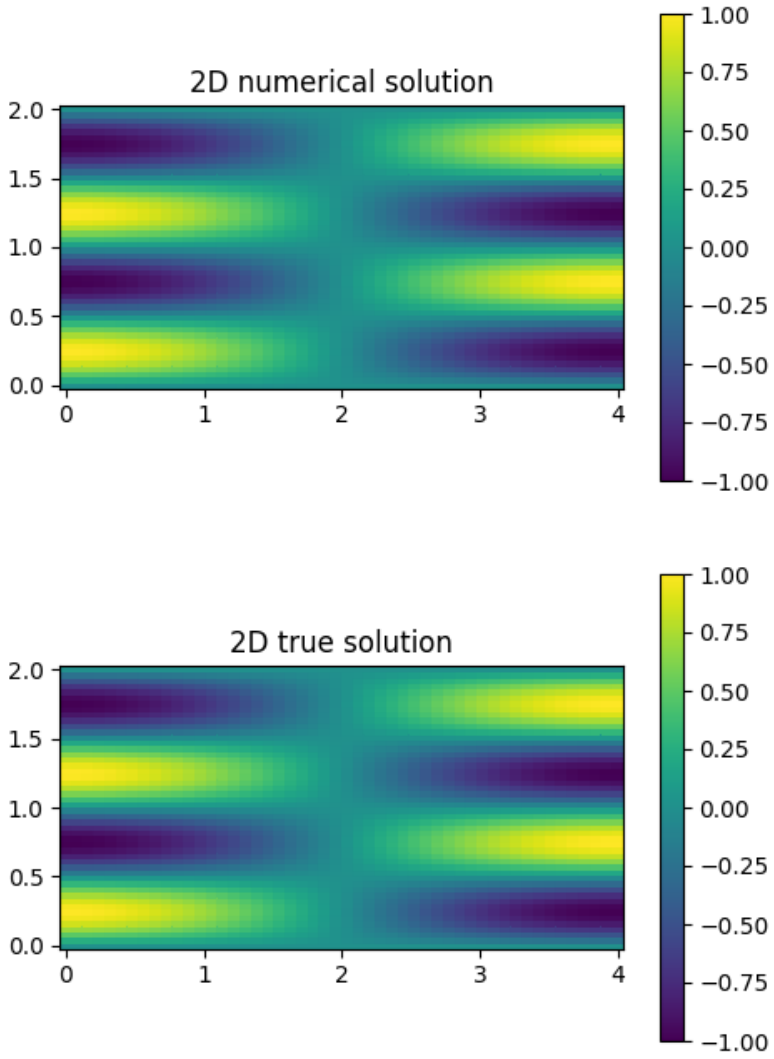

Figure 3.7 $\mathbb{M}_D^1$ in 2D.

4 Adding a Gyrokinetic Turbulence Model with Adiabatic Electrons in Struphy

Following the implementation of the toy Poisson model in Struphy, the main goal of this thesis is to add a gyrokinetic model. Firstly, the model needs to be analysed and discretized. We will consider the following gyrokinetic model:

$$\frac{\partial f}{\partial t} + \left[v_\parallel \frac{\mathbf{B}^*}{B^*_\parallel} + \frac{\mathbf{E}^* \times \mathbf{b}_0}{B^*_\parallel} \right] \cdot \nabla f + \frac{q}{m} \left[\frac{\mathbf{B}^*}{B^*_\parallel} \cdot \mathbf{E}^* \right] \cdot \frac{\partial f}{\partial v_\parallel} = 0 \,, \tag{4.1}$$

$$-\nabla_\perp \cdot \left(\frac{m n_0}{|B_0|^2} \right) \nabla_\perp \phi + e n_{e0} \left(1 + \frac{e}{k_B T_{e0}} \phi \right) = q \int f B^*_\parallel \, \mathrm{d}v_\parallel \mathrm{d}\mu \,, \tag{4.2}$$

where $f(\mathbf{x}, v_\parallel, \mu, t)$ is the guiding center distribution, $\phi(\mathbf{x})$ denotes the electrostatic potential, n_0 and T_{e0} are electron equilibrium density and temperature, respectively, and

$$\mathbf{E}^* = -\nabla\phi - \frac{\mu}{q} \nabla |B_0| \,, \qquad \mathbf{B}^* = \mathbf{B}_0 + \frac{m}{q} v_\parallel \nabla \times \mathbf{b}_0 \,, \qquad B^*_\parallel = \mathbf{B}^* \cdot \mathbf{b}_0 \,.$$

Here, $\mathbf{B}_0$ with direction $\mathbf{b}_0 = \mathbf{B}_0/|B_0|$ stands for a static equilibrium magnetic field.

In this scenario the magnetic field strength $\mathbf{B}_0$ is significant, and the swift gyrations induced by the $\mathbf{v} \times \mathbf{b}_0$ term coupled with the six-dimensional nature of equation 4.1. The context of turbulence timescales and the resultant high-frequency collective oscillations, present challenges for investigating low-frequency nonlinear physics. Therefore, researchers made an averaging procedure aimed at decreasing the effects of rapid gyration[8]. So they transformed the six-dimensional kinetic equation into a more manageable five-dimensional one. This approach, complemented by a specialized closure of Poisson's equation tailored to low-frequency dynamics, establishes the framework of low-frequency gyrokinetics.

There are three important properties of the gyrokinetic turbulance model represented above:

- **Adiabatic Electrons:** Adiabatic electrons are used to write the gyrokinetic Poisson equation. Let's first define what adiabatic invariant is.

Definition 4.0.1 (Adiabatic Invariant) *A quantity that is conserved through all orders under slow variations.[8]*

When fluctuations are one dimensional in plasma, normally they would be two in two component plasma, rather than writing full equations of motion to see the behavior of electrons one can write collective behavior of electrons through an equation of state. BERNSTEIN and TREHAN (1960) [1] hypothesize the existence of thermal equilibrium among the electrons, thereby we can write Taylor expansion of behavior of electrons through an equation of state. Therefore n_e is as follows [6]:

$$n_e = n_{e,0} \exp(\frac{e\phi}{k_B T_{e,0}}) = n_{e,0}(1 + \frac{e}{k_B T_{e0}} + ..)$$

which allows us to write second term in the left hand side of the gyrokinetic poisson in (4.2).

- **Polarization:** Position of the particle in gyrokinetic theory is divided into two parts : gyrocenter position and gyration around the center. Since there is a gyromotion around magnetic field gyrocenter represents averaged position. By this, we get a polarization charge where the total charge is split into free charge and polarization charge, the latter of which is described by a polarization vector. This polarization affects the continuity equation for the polarization charge and modifies the Poisson equation in the gyrokinetic framework.

- **Quasi-neutrality:** Plasma consists of positively charged ions and negatively charged electrons and quasi-neutrality means density of ions are approximately equal to density of electrons, i.e.

$$n_i(x) = n_e(x)$$

and incorporates the gyrocenter polarization.

4.1 Normalization

Normalization of the units is a key step for us at the beginning of the model discretization. Goal here is to find suitable units for the model quantities. A quantity in Struphy is represented as :

$$\mathbf{X} = X' \hat{X}$$

where $\hat{X}$ is the unit and X' is the numerical value of the quantity. In Struphy there are three basic units: the unit of length $\hat{x}$, expressed in meter, the unit of the magnetic field strength $\hat{B}$, expressed in Tesla, the unit of number density, expressed in $10^{20} m^{-3}$. These three units can be defined by user. Moreover,

there are some other units which are fixed such as: $\hat{p}, \hat{\rho}, \hat{j}, \hat{t}$. We use basic and fixed units to derive others and find the simplest equation form. In the gyrokinetic model following derivations are going to help us. Let us first introduce the unit of the electron cyclotron frequency and its product with the time unit,

$$\hat{\Omega}_c := \frac{q}{m}\hat{B} \qquad \varepsilon := \frac{1}{\hat{\Omega}_c \hat{t}}\,.$$

In our model, it makes sense to set the unit of the $E \times B$-velocity to $\hat{v}$,

$$\frac{\hat{E}}{\hat{B}} = \hat{v} = \frac{\hat{x}}{\hat{t}}\,.$$

This determines the unit $\hat{E}$ of the electric field and renders Faraday's law scale invariant. It also sets the unit for the electric potential,

$$\hat{\phi} = \hat{E}\hat{x} = \hat{v}\hat{B}\hat{x}\,.$$

Here is the representation of the equations of gyrokinetic model with units and numerical values :

$$\frac{\hat{f}}{\hat{t}}\frac{\partial f'}{\partial t'} + \frac{\hat{f}}{\hat{x}}\left[\hat{v}v'_\parallel \frac{\mathbf{B}^{*\prime}}{B^{*\prime}_\parallel} + \frac{\hat{E}}{\hat{B}}\frac{\mathbf{E}^{*\prime}\times \mathbf{b}_0}{B^{*\prime}_\parallel}\right]\cdot\nabla f' + \frac{q\hat{E}\hat{f}}{m\hat{v}_\parallel}\left[\frac{\mathbf{B}^{*\prime}}{B^{*\prime}_\parallel}\cdot\mathbf{E}^{*\prime}\right]\cdot\frac{\partial f'}{\partial v'_\parallel} = 0\,, \tag{4.3}$$

$$-\left(\frac{m\hat{n}_0\hat{\phi}}{\hat{B}^2\hat{x}^2}\right)\nabla_\perp\cdot\left(\frac{n'_0}{|B'_0|^2}\right)\nabla_\perp\phi' + en'_{e0}\left(1+\frac{e}{k_B T_{e0}}\hat{\phi}\phi'\right) = q\hat{n}\int f' B^{*\prime}_\parallel \,\mathrm{d}v'_\parallel \mathrm{d}\mu' \tag{4.4}$$

By multiplying each unit in (4.3) with $\frac{\hat{t}}{\hat{f}}$ and using conventions in Struphy $\frac{\hat{x}}{\hat{t}} = \hat{v}$ and $\frac{\hat{E}}{\hat{B}} = \hat{v}$ we get

$$\frac{\partial f'}{\partial t'} + \left[v'_\parallel \frac{\mathbf{B}^{*\prime}}{B^{*\prime}_\parallel} + \frac{\mathbf{E}^{*\prime}\times \mathbf{b}_0}{B^{*\prime}_\parallel}\right]\cdot\nabla f' + \frac{q\hat{E}\hat{t}}{m\hat{v}_\parallel}\left[\frac{\mathbf{B}^{*\prime}}{B^{*\prime}_\parallel}\cdot\mathbf{E}^{*\prime}\right]\cdot\frac{\partial f'}{\partial v'_\parallel} = 0.$$

Furthermore, using $\Omega_c = \frac{\hat{q}\hat{B}}{m}$ and $\epsilon = \frac{1}{\Omega_c \hat{t}}$:

$$\begin{aligned}\frac{q\hat{E}\hat{t}\hat{B}}{m\hat{v}\hat{B}} &= \frac{q\hat{B}\hat{t}}{m}\\ &= \Omega_c\hat{t}\\ &= \frac{1}{\epsilon}.\end{aligned}$$

Then to simply (4.2) use $\hat{\phi} = \hat{E}\hat{x} = \hat{v}\hat{B}\hat{x}$ and $q = Ze$ divide everything with $q\hat{n}$. Furthermore, by quasi-neutrality $n'_{e0} = Zn_0$:

$$-\left(\frac{\hat{m}}{\hat{B}\hat{t}q}\right)\nabla_\perp \cdot \left(\frac{n'_0}{|B'_0|^2}\right)\nabla_\perp \phi' + \left(1 + \frac{e\hat{B}\hat{v}\hat{x}}{k_B T_{e0}}\phi'\right) = \int f' B^{*\prime}_{\parallel} dv'_{\parallel} d\mu'.$$

As above $\epsilon = \dfrac{m}{\hat{q}\hat{B}\hat{t}}$ and $q = Ze$

$$\begin{aligned} e\hat{B}\hat{v}\hat{x} &= \frac{e\hat{B}}{m}\hat{t}m\hat{v}\frac{\hat{x}}{\hat{t}} \\ &= \frac{1}{Z}\Omega_c \hat{t} m\hat{v}^2 \\ &= \frac{1}{Z\epsilon} m\hat{v}^2. \end{aligned}$$

Since velocity scale is chosen to be thermal, $m\hat{v}^2$ and $k_B T_{e0}$ are both energies they have same unit. Thermal velocity is given by $\hat{v} = \hat{v}_{th} = \sqrt{\frac{k_B T}{m}}$. So that we can use it as

$$e\hat{B}\hat{v}\hat{x} = \frac{1}{Z\epsilon} k_B T_{e0}.$$

Finally, normalized versions of (4.1) and (4.2) are given by :

$$\frac{\partial f}{\partial t} + \left[v_\parallel \frac{\mathbf{B}^*}{B^*_\parallel} + \frac{\mathbf{E}^* \times \mathbf{b}_0}{B^*_\parallel}\right] \cdot \nabla f + \frac{1}{\epsilon}\left[\frac{\mathbf{B}^*}{B^*_\parallel} \cdot \mathbf{E}^*\right] \cdot \frac{\partial f}{\partial v_\parallel} = 0 \tag{4.5}$$

$$-\epsilon \nabla_\perp \cdot \left(\frac{n_0}{|B_0|^2}\right)\nabla_\perp \phi + n_0\left(1 + \frac{1}{Z\epsilon}\phi\right) = \int f B^*_\parallel \, \mathrm{d}v_\parallel \mathrm{d}\mu \tag{4.6}$$

where

$$\mathbf{E}^* = -\nabla\phi - \epsilon\mu\nabla|B_0|\,, \qquad \mathbf{B}^* = \mathbf{B}_0 + \epsilon v_\parallel \nabla \times \mathbf{b}_0.$$

From now on every equation of the model is normalized as above. In the normalization steps the same assumptions are used.

4.2 Lagrangian

The Lagrangian of the above system reads

$$L((\mathbf{u}_{gc}, a_\parallel), f, \phi) = \int_\Omega \left[(v_\parallel \mathbf{b}_0 + \frac{1}{\epsilon}\mathbf{A}_0) \cdot \mathbf{u}_{gc} - \frac{v_\parallel^2}{2} - \mu|B_0| - \frac{1}{\epsilon}\phi\right] f B^*_\parallel \, \mathrm{d}v_\parallel \mathrm{d}\mu \, d\mathbf{x} \tag{4.7}$$

$$+\int_\Omega \frac{1}{2|B_0|^2}|\nabla_\perp\phi|^2 f_0 B_\parallel^* \,\mathrm{d}v_\parallel \mathrm{d}\mu \, d\mathbf{x} + \int_\Omega \frac{1}{\epsilon}\phi n_0 \left(1+\frac{1}{2Z\epsilon}\phi\right) d\mathbf{x} \tag{4.8}$$

where $\nabla \times \mathbf{A}_0 = \mathbf{B}_0$ and $(\mathbf{u}_{\mathrm{gc}}, a_\parallel)$ denote the Eulerian phase space velocity, i.e. the vector field in the ODE

$$\dot{\mathbf{x}}(t) = \mathbf{u}_{\mathrm{gc}}(t, \mathbf{x}(t), v_\parallel(t))\,, \tag{4.9}$$

$$\dot{v_\parallel}(t) = a_\parallel(t, \mathbf{x}(t), v_\parallel(t))\,. \tag{4.10}$$

The first term in the line of (4.8),

$$H_2 := -\int_\Omega \frac{1}{2|B_0|^2}|\nabla_\perp\phi|^2 f_0 B_\parallel^* \,\mathrm{d}v_\parallel \mathrm{d}\mu \,\mathrm{d}\mathbf{x}$$

is a second-order correction coming from the gyrokinetic coordinate transformation. The function $f^{\mathrm{vol}} := f B_\parallel^*$ is assumed a conserved volume-form transported with the flow of (4.9)-(4.10), i.e.

$$\partial_t f^{\mathrm{vol}} + \nabla\cdot(\mathbf{u}_{\mathrm{gc}} f^{\mathrm{vol}}) + \frac{\partial}{\partial v_\parallel}(a_\parallel f^{\mathrm{vol}}) = 0\,. \tag{4.11}$$

Moreover, the Jacobian $B_\parallel^*$ of the gyrokinetic coordinate transformation satisfies the Lioville theorem,

$$\partial_t B_\parallel^* + \nabla\cdot(\mathbf{u}_{\mathrm{gc}} B_\parallel^*) + \frac{\partial}{\partial v_\parallel}(a_\parallel B_\parallel^*) = 0\,, \tag{4.12}$$

which leads to a transport equation for f:

$$\partial_t f + \mathbf{u}_{\mathrm{gc}}\cdot\nabla f + a_\parallel \frac{\partial f}{\partial v_\parallel} = 0\,. \tag{4.13}$$

Note that the Lagrangian L does not depend on $a_\parallel$. The action is as usual given by

$$I((\mathbf{u}_{\mathrm{gc}}, a_\parallel), f, \phi) = \int_{t_0}^{t_1} L((\mathbf{u}_{\mathrm{gc}}, a_\parallel), f, \phi)\,\mathrm{d}t$$

for arbitrary time intervals. The gyrokinetic Poisson equation follows from the Euler-Lagrange equation

$$\frac{\delta I}{\delta \phi} = 0\,.$$

The expressions for $(\mathbf{u}_{\mathrm{gc}}, a_{\parallel})$ could be derived from constrained variations as described in section (2.2). Here, however, we shall take a different approach and derive the equations of motion in Lagrangian coordinates. These are defined as $(\mathbf{x}_0, v_{\parallel,0})$ in

$$(\mathbf{x}, v_{\parallel}) = \Phi_t(\mathbf{x}_0, v_{\parallel,0})\,,$$

where Φ_t is the flow of (4.9)-(4.10), i.e. the solution of the ODE system with dependence on the initial condition (which are the Lagrangian coordinates).

Definition 4.2.1 (Right Invariant) *A Lagrange density $\ell_0 : \mathbf{T}\Omega \times \Omega \to \mathbb{R}$ is called right invariant if, under the change of coordinates $\mathbf{q}_0 = \mathbf{\Phi}_t^{-1}(\mathbf{q})$, $\Omega \to \Omega$ it satisfies*

$$\int_\Omega \ell_0(\mathbf{\Phi}_t(\mathbf{q_0}), \dot{\mathbf{\Phi}}_t(\mathbf{q}_0), q_0) d\mathbf{q}_0 = \int_\Omega \ell(t, \mathbf{q}, u(t, \mathbf{q})) d\mathbf{q}.$$

We can now use that the Lagrange density in (4.7) is right-invariant and substitute the change of coordinates $(\mathbf{x}, v_{\parallel}) = (\mathcal{X}_t, \mathcal{V}_{\parallel,t})$, where $(\mathcal{X}_t, \mathcal{V}_{\parallel,t})$ denotes the space- resp. velocity-components of the flow Φ_t. Noting that the flow satisfies (4.9)-(4.10) we have

$$\dot{\mathcal{X}}_t = \mathbf{u}_{\mathrm{gc}}(t, \mathcal{X}_t, \mathcal{V}_{\parallel,t})\,, \tag{4.14}$$

$$\dot{\mathcal{V}}_{\parallel,t} = a_{\parallel}(t, \mathcal{X}_t, \mathcal{V}_{\parallel,t})\,, \tag{4.15}$$

such that the Lagrangian becomes

$$\begin{aligned} L(\Phi_t, \dot{\Phi}_t, \phi) = \int_\Omega [(\mathcal{V}_{\parallel,t}\mathbf{b}_0(\mathcal{X}_t) + \frac{1}{\epsilon}\mathbf{A}_0(\mathcal{X}_t)) \cdot \dot{\mathcal{X}}_t - \frac{\mathcal{V}_{\parallel,t}^2}{2} - \mu|B_0|(\mathcal{X}_t) - \frac{1}{\epsilon}\phi(\mathcal{X}_t)] f_0 B^*_{\parallel,0}\, \mathrm{d}v_{\parallel,0} d\mu_0 d\mathbf{x_0} \\ + \int_\Omega \frac{1}{2|B_0|^2}|\nabla_\perp \phi|^2 f_0 B^*_{\parallel}\, dv_{\parallel} d\mu\, d\mathbf{x} + \int_\Omega \frac{1}{\epsilon}\phi n_0 (1 + \frac{1}{Z\epsilon}\phi) d\mathbf{x}. \end{aligned} \tag{4.16}$$

Unconstrained variations with respect to Φ_t now yield the classical Euler-Lagrange equations for ordinary differential equations,

$$\frac{\partial L_{\mathrm{gc}}}{\partial \Phi_t} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L_{\mathrm{gc}}}{\partial \dot{\Phi}_t} = 0\,,$$

where L_{gc} is the expression in brackets in (4.7), which is the sole part of L that depends on Φ_t,

$$L_{\mathrm{gc}} = (\mathcal{V}_{\parallel,t}\mathbf{b}_0(\mathcal{X}_t) + \frac{1}{\epsilon}\mathbf{A}_0(\mathcal{X}_t)) \cdot \dot{\mathcal{X}}_t - \frac{\mathcal{V}_{\parallel,t}^2}{2} - \mu|B_0|(\mathcal{X}_t) - \frac{1}{\epsilon}\phi(\mathcal{X}_t)\,.$$

We compute the derivatives of the single particle Lagrangian,

$$\frac{\partial L_{\mathrm{gc}}}{\partial \mathcal{X}_t} = (\mathcal{V}_{\parallel,t}\nabla \mathbf{b}_0 + \frac{1}{\epsilon}\nabla \mathbf{A}_0)\cdot \dot{\mathcal{X}}_t - \mu\nabla|B_0| - \frac{1}{\epsilon}\nabla\phi\,, \tag{4.17}$$

$$\frac{\partial L_{\mathrm{gc}}}{\partial \dot{\mathcal{X}}_t} = \mathcal{V}_{\parallel,t}\mathbf{b}_0(\mathcal{X}_t) + \frac{1}{\epsilon}\mathbf{A}_0(\mathcal{X}_t)\,, \tag{4.18}$$

$$\frac{\partial L_{\mathrm{gc}}}{\partial \mathcal{V}_{\parallel,t}} = \mathbf{b}_0\cdot\dot{\mathcal{X}}_t - \mathcal{V}_{\parallel,t}\,, \tag{4.19}$$

$$\frac{\partial L_{\mathrm{gc}}}{\partial \dot{\mathcal{V}}_{\parallel,t}} = 0\,. \tag{4.20}$$

From the last two equations we obtain

$$\mathbf{b}_0\cdot\dot{\mathcal{X}}_t = \mathcal{V}_{\parallel,t}\,. \tag{4.21}$$

Moreover, taking the time derivative of (4.18) yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L_{\mathrm{gc}}}{\partial \dot{\mathcal{X}}_t} = \dot{\mathcal{V}}_{\parallel,t}\mathbf{b}_0 + \mathcal{V}_{\parallel,t}\dot{\mathcal{X}}_t\cdot\nabla\mathbf{b}_0 + \frac{1}{\epsilon}\dot{\mathcal{X}}_t\cdot\nabla\mathbf{A}_0\,.$$

By taking the difference of this expression with (4.17) and using the formula $\nabla\mathbf{A}\cdot\mathbf{b} - \mathbf{b}\cdot\nabla\mathbf{A} = \mathbf{b}\times(\nabla\times\mathbf{A})$ we obtain

$$\dot{\mathcal{X}}_t\times(\mathcal{V}_{\parallel,t}\nabla\times\mathbf{b_0} + \frac{1}{\epsilon}\mathbf{B}_0) - \mu\nabla|B_0| - \frac{1}{\epsilon}\nabla\phi = \dot{\mathcal{V}}_{\parallel,t}\mathbf{b}_0\,.$$

Hence we substitute the definitions of $\mathbf{E}^*$ and $\mathbf{B}^*$ form above to write this as

$$\epsilon\dot{\mathcal{X}}_t\times\mathbf{B}^* + \epsilon\mathbf{E}^* = m\dot{\mathcal{V}}_{\parallel,t}\mathbf{b}_0\,. \tag{4.22}$$

We can take the scalar product of this equation with $\mathbf{B}^*$ to obtain the first equation of motion,

$$\dot{\mathcal{V}}_{\parallel,t} = \frac{1}{\epsilon}\frac{\mathbf{B}^*}{B^*_\parallel}\cdot\mathbf{E}^*\,.$$

Finally, in (4.22) we take the cross product with $\mathbf{b}_0$ form the right,

$$\epsilon(\dot{\mathcal{X}}_t\times\mathbf{B}^*)\times\mathbf{b}_0 + \epsilon\mathbf{E}^*\times\mathbf{b}_0 = 0\,,$$

then use the formula $(\mathbf{a}\times\mathbf{b})\times\mathbf{c} = (\mathbf{c}\cdot\mathbf{a})\mathbf{b} - (\mathbf{c}\cdot\mathbf{b})\mathbf{a}$ and simplify to get rid of ϵ, one can obtain

$$(b_0 \cdot \dot{\mathcal{X}}_t)\mathbf{B}^* - (\mathbf{b}_0 \cdot \mathbf{B}^*)\dot{\mathcal{X}}_t + \mathbf{E}^* \times \mathbf{b}_0 = 0\,.$$

Using now (4.21) yields the other equations of motion,

$$\dot{\mathcal{X}}_t = \mathcal{V}_{\parallel,t}\frac{\mathbf{B}^*}{B_\parallel^*} + \frac{\mathbf{E}^* \times \mathbf{b}_0}{B_\parallel^*}\,.$$

4.3 Noether's theorem

Let's first state Noether's theorem and adapt it to our case to ensure energy conservation.

Theorem 4.3.1 *A symmetry on a Lagrangian corresponds to a quantity Q conserved in time. [5]*

For the detailed proof see [13].

The symmetry of L with respect to time transformations is due to the fact that L has no explicit time dependence, i.e. time only appears through the time-dependence of the unknowns. Hence, the following energy is conserved.

$$W_{\mathrm{tot}} = \int \mathbf{u}_{\mathrm{gc}} \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{u}_{\mathrm{gc}}}\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x} - L\,,$$

where $\mathcal{L}$ is the Lagrange density such that

$$L = \int \mathcal{L}\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{x}\,.$$

For the Lagrangian (4.7) this leads to

$$W_{\mathrm{tot}} = \int_\Omega \left(\frac{v_\parallel^2}{2} + \mu|B_0| + \frac{1}{\epsilon}\phi\right) f B_\parallel^*\,\mathrm{d}v_\parallel \mathrm{d}\mu\,\mathrm{d}\mathbf{x} - \int_\Omega \frac{1}{2|B_0|^2}|\nabla_\perp \phi|^2 f_0 B_\parallel^*\,\mathrm{d}v_\parallel \mathrm{d}\mu\,\mathrm{d}\mathbf{x} - \int_\Omega \frac{1}{\epsilon}\phi n_0\left(1 + \frac{1}{2Z\epsilon}\phi\right)\,\mathrm{d}\mathbf{x}\,.$$

We can use

$$n_0 = \int_\Omega f_0 B_\parallel^*\,\mathrm{d}v_\parallel \mathrm{d}\mu\,,$$

Likewise, we write the gyrokinetic Poisson equation as

$$-\nabla_\perp \cdot \left(\frac{n_0}{|B_0|^2}\right)\nabla_\perp \phi + \frac{n_0}{Z\epsilon^2}\phi = \frac{1}{\epsilon}\int f B_\parallel^*\,\mathrm{d}v_\parallel \mathrm{d}\mu - \frac{n_0}{\epsilon},$$

integrate by parts and substitute it to obtain

$$W_{\text{tot}} = \int_\Omega \left(\frac{v_\parallel^2}{2} + \mu |B_0| \right) f B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} + \int_\Omega \frac{n_0}{2|B_0|^2} |\nabla_\perp \phi|^2 \, \mathrm{d}\mathbf{x} + \int_\Omega \frac{n_0}{2Z\epsilon^2} \phi^2 \, \mathrm{d}\mathbf{x} \,.$$

Let us verify that this energy is indeed conserved:

$$\begin{aligned}
\dot{W}_{\text{tot}} &= \int_\Omega \left(\frac{v_\parallel^2}{2} + \mu |B_0| \right) (\partial_t f B_\parallel^*) \, \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} - \int_\Omega \phi \partial_t \, \nabla_\perp \cdot \frac{n_0}{|B_0|^2} \nabla_\perp \phi \, \mathrm{d}\mathbf{x} + \int_\Omega \frac{1}{Z\epsilon^2} \phi \, \partial_t \phi \, \mathrm{d}\mathbf{x} \\
&= \int_\Omega \left(\frac{v_\parallel^2}{2} + \mu |B_0| \right) (\partial_t f B_\parallel^*) \, \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} + \int_\Omega \frac{1}{\epsilon} \phi \partial_t \left(\int f B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu - n_0 \right) \mathrm{d}\mathbf{x} \\
&= \int_\Omega \left(\frac{v_\parallel^2}{2} + \mu |B_0| + \frac{1}{\epsilon} \phi \right) (\partial_t f B_\parallel^*) \, \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} \,.
\end{aligned}$$

We can now substitute equation (4.11) and integrate by parts to obtain

$$\begin{aligned}
\dot{W}_{\text{tot}} &= - \int_\Omega \left(\frac{v_\parallel^2}{2} + \mu |B_0| + \frac{1}{\epsilon} \phi \right) \left[\nabla \cdot \left(v_\parallel \mathbf{B}^* f + \mathbf{E}^* \times \mathbf{b}_0 f \right) + \frac{1}{\epsilon} \partial_{v_\parallel} (\mathbf{E}^* \cdot \mathbf{B}^* f) \right] \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} \\
&= \int_\Omega \frac{1}{\epsilon} \left[-\mathbf{E}^* \cdot v_\parallel \mathbf{B}^* f + v_\parallel \mathbf{E}^* \cdot \mathbf{B}^* f \right] \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} \\
&= 0 \,.
\end{aligned}$$

4.4 Energy-conserving semi-discretization of gyrokinetic Vlasov-Poisson

First, let us rewrite the gyrokinetic poisson equation

$$-\nabla_\perp \cdot \left(\frac{n_0}{|B_0|^2} \right) \nabla_\perp \phi + \frac{n_0}{Z\epsilon^2} \phi = \frac{1}{\epsilon} \int f B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu - \frac{n_0}{\epsilon}. \tag{4.23}$$

Then the weak form of the gyrokinetic Poisson equation (L^2-scalar product with test function $\psi \in H^1$ and integration by parts on the left):

$$\int \frac{n_0}{|B_0|^2} \nabla_\perp \phi \cdot \nabla_\perp \psi \, \mathrm{d}\mathbf{x} + \int \frac{n_0(\mathbf{x})}{Z\epsilon^2} \phi \, \psi \, \mathrm{d}\mathbf{x} = \frac{1}{\epsilon} \int \int \psi f B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu \, \mathrm{d}\mathbf{x} - \int \frac{n_0(\mathbf{x})}{\epsilon} \psi \, \mathrm{d}\mathbf{x} \,, \qquad \forall \, \psi \in H^1 \,,$$

Next, we have to use the mapping $F : \boldsymbol{\eta} \mapsto \mathbf{x}$ and the transformation rule of the gradient,

$$\nabla_x \phi(\mathbf{x}) = DF^{-\top} \hat{\nabla}_\eta \hat{\phi}(\boldsymbol{\eta}) \,,$$

where $DF : \boldsymbol{\eta} \mapsto \mathbb{R}^{3\times 3}$ denotes the Jacobian matrix of the mapping and $\hat{\phi}(\boldsymbol{\eta}) := \phi(F(\boldsymbol{\eta}))$ is the transformation of 0-forms. The perpendicular gradient is approximated on the logical unit cube to consist only of the first two components of the full gradient,

$$\nabla_\perp \phi \approx DF^{-\top} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \hat{\nabla}_\eta \hat{\phi} \,.$$

This is possible because in Struphy, the poloidal plane is always spanned by the first two coordinates (η_1, η_2), and to a first approximation the magnetic background field can be assumed perpendicular to the poloidal plane. We can thus pull back the weak Poisson equation to the logical unit cube:

$$\int \frac{n_0}{|B_0|^2} \hat{\nabla}\hat{\phi}^\top G_\perp^{-1} \hat{\nabla}\hat{\psi} \sqrt{g} \,\mathrm{d}\boldsymbol{\eta} + \int \frac{1}{Z\epsilon^2} \hat{\phi}\hat{\psi} \sqrt{g} \,\mathrm{d}\boldsymbol{\eta} = \frac{1}{\epsilon} \int \int \hat{\psi} \hat{f}^{\mathrm{vol}} \,\mathrm{d}v_\parallel \mathrm{d}\mu \,\mathrm{d}\boldsymbol{\eta} - \int \frac{1}{\epsilon} \hat{n}_0(\boldsymbol{\eta}) \hat{\psi} \sqrt{g} \,\mathrm{d}\boldsymbol{\eta} \,, \qquad \forall\, \hat{\psi} \in H^1 \,. \tag{4.24}$$

Here,

$$G_\perp^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} G^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \qquad G = DF^\top DF \,,$$

denotes the "perpendicular components" of the inverse metric tensor G^{-1}, $\sqrt{g} = |\det DF|$ stands for the Jacobian determinant and $\hat{f}^{\mathrm{vol}} = \hat{f}\hat{B}^*_\parallel \sqrt{g}$ is the volume form corresponding to f. The electrostatic potential is discretized as a 0-form:

$$\hat{\phi} \approx \hat{\phi}_h(t, \boldsymbol{\eta}) = \sum_{i=0}^{N_0 - 1} \phi_i(t) \, \Lambda_i^0(\boldsymbol{\eta}) = \boldsymbol{\phi}^\top \boldsymbol{\Lambda}^0(\boldsymbol{\eta}) \quad \in V_h^0 \subset H^1 \,.$$

Its discrete gradient lives in $V_h^1 \subset H(\mathrm{curl})$ and can be written as

$$\hat{\nabla}\hat{\phi}_h(t, \boldsymbol{\eta}) = \sum_{i=0}^{N_1 - 1} (\mathbb{G}\boldsymbol{\phi}(t))_i \, \Lambda_i^1(\boldsymbol{\eta}) \quad \in V_h^1 \subset H(\mathrm{curl}) \,, \tag{4.25}$$

where $\mathbb{G} \in \mathbb{R}^{N_1 \times N_0}$ is the gradient matrix. We can thus already write all but one term in (4.23) in discrete form:

$$\boldsymbol{\psi}^\top \mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}} \mathbb{G}\, \boldsymbol{\phi} + \frac{1}{Z\epsilon^2} \boldsymbol{\psi}^\top \mathbb{M}^0_{\mathrm{ad}}\, \boldsymbol{\phi} = \frac{1}{\epsilon} \boldsymbol{\psi}^\top \int \int \boldsymbol{\Lambda}^0 \hat{f}^{\mathrm{vol}}\, \mathrm{d}v_\parallel \mathrm{d}\mu\, \mathrm{d}\boldsymbol{\eta} - \boldsymbol{\psi}^\top \int \frac{1}{\epsilon} \hat{n}_0(\boldsymbol{\eta}) \boldsymbol{\Lambda}^0 \sqrt{g}\, \mathrm{d}\boldsymbol{\eta}\,, \qquad \forall\, \boldsymbol{\psi} \in \mathbb{R}^{N_0}\,. \tag{4.26}$$

Here, bold symbols such as $\boldsymbol{\phi} = (\phi_i)_{i=0}^{N_0-1} \in \mathbb{R}^{N_0}$ indicate vectors, with the Euclidean scalar product denoted by $(\boldsymbol{\psi}, \boldsymbol{\phi}) = \boldsymbol{\psi}^\top \boldsymbol{\phi}$ and we introduced the two weighted mass matrices

$$\mathbb{M}^1_{\mathrm{gyro}} = \int \frac{n_0}{|B_0|^2} \boldsymbol{\Lambda}^1 G_\perp^{-1} (\boldsymbol{\Lambda}^1)^\top \sqrt{g}\, \mathrm{d}\boldsymbol{\eta} \qquad \in \mathbb{R}^{N_1 \times N_1}\,, \tag{4.27}$$

$$\mathbb{M}^0_{\mathrm{ad}} = \int n_0 \boldsymbol{\Lambda}^0 (\boldsymbol{\Lambda}^0)^\top \sqrt{g}\, \mathrm{d}\boldsymbol{\eta} \qquad \in \mathbb{R}^{N_0 \times N_0}\,. \tag{4.28}$$

The guiding centers are discretized with N particles:

$$\hat{f}^{\mathrm{vol}} \approx \hat{f}^{\mathrm{vol}}_h(t, \boldsymbol{\eta}, v_\parallel) = \frac{1}{N} \sum_{k=0}^{N-1} w_k\, \delta(\boldsymbol{\eta} - \boldsymbol{\eta}_k(t))\, \delta(v_\parallel - v_{\parallel,k}(t))\,,$$

where $(\boldsymbol{\eta}_k(t), v_{\parallel,k}(t))$ satisfy the ODEs (4.8)-(4.9) on the logical domain:

$$\dot{\boldsymbol{\eta}}_k = DF^{-1} \left(v_{\parallel,k} \frac{DF \hat{\mathbf{B}}^{*2}}{\sqrt{g} \hat{B}^*_\parallel} + \frac{DF^{-\top} \hat{\mathbf{E}}^{*1} \times DF^{-\top} \hat{\mathbf{b}}^1_0}{\hat{B}^*_\parallel} \right)\,, \tag{4.29}$$

$$\dot{v}_{\parallel,k} = \frac{1}{\epsilon} \frac{DF^{-\top} \hat{\mathbf{E}}^{*1} \cdot DF \hat{\mathbf{B}}^{*2}}{\sqrt{g} \hat{B}^*_\parallel}\,, \tag{4.30}$$

where we used the following pull-back formulas:

$$\mathbf{B}^* = \frac{DF}{\sqrt{g}} \hat{\mathbf{B}}^{*2} \qquad (2\text{-form pullback})\,,$$

$$\mathbf{E}^* = DF^{-\top} \hat{\mathbf{E}}^{*1} \qquad (1\text{-form pullback})\,,$$

$$\mathbf{b}_0 = DF^{-\top} \hat{\mathbf{b}}^{*1}_0 \qquad (1\text{-form pullback})\,,$$

$$B^*_\parallel(F(\boldsymbol{\eta})) = \hat{B}^*_\parallel(\boldsymbol{\eta}) \qquad (0\text{-form pullback})\,.$$

Moreover, by using the formula $M\mathbf{a} \times M\mathbf{b} = \det(M) M^{-\top} (\mathbf{a} \times \mathbf{b})$ and by substituting the 3-form representation $\hat{B}^{*3}_\parallel = \sqrt{g} \hat{B}^*_\parallel$, we obtain the final version

$$\dot{\boldsymbol{\eta}}_k = v_{\parallel,k} \frac{\hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}} + \frac{\hat{\mathbf{E}}^{*1} \times \hat{\mathbf{b}}^1_0}{\hat{B}^{*3}_{\parallel}}\,, \tag{4.31}$$

$$\dot{v}_{\parallel,k} = \frac{1}{\epsilon} \frac{\hat{\mathbf{E}}^{*1} \cdot \hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}}\,. \tag{4.32}$$

The coupling term in the discrete Poisson equation (4.24) becomes

$$\boldsymbol{\psi}^\top \int \int \boldsymbol{\Lambda}^0 \hat{f}^{\mathrm{vol}} \,\mathrm{d}v_\parallel \mathrm{d}\mu \,\mathrm{d}\boldsymbol{\eta} \approx \boldsymbol{\psi}^\top \frac{1}{N\epsilon} \sum_{k=0}^{N-1} w_k \boldsymbol{\Lambda}^0(\boldsymbol{\eta}_k(t))\,.$$

Hence, the discrete Poisson equation reads

$$\mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}} \mathbb{G}\, \boldsymbol{\phi} + \frac{1}{Z\epsilon^2} \mathbb{M}^0_{\mathrm{ad}}\, \boldsymbol{\phi} = \frac{1}{N\epsilon} \sum_{k=0}^{N-1} w_k \boldsymbol{\Lambda}^0(\boldsymbol{\eta}_k(t)) - \int \frac{1}{\epsilon} \hat{n}_0(\boldsymbol{\eta}) \boldsymbol{\Lambda}^0 \sqrt{g}\,\mathrm{d}\boldsymbol{\eta}\,. \tag{4.33}$$

The discrete version of the energy reads

$$W_{\mathrm{tot}}(t) = \frac{1}{N} \sum_{k=1}^{N-1} w_k \left(\frac{v_{\parallel,k}(t)^2}{2} + \mu_k |\hat{B}_0|(\boldsymbol{\eta}_k(t)) \right) + \boldsymbol{\phi}^\top(t) \mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}} \mathbb{G}\, \boldsymbol{\phi}(t) + \frac{1}{2Z\epsilon^2} \boldsymbol{\phi}^\top(t) \mathbb{M}^0_{\mathrm{ad}}\, \boldsymbol{\phi}(t)\,. \tag{4.34}$$

Let us verify that this semi-discrete energy is indeed conserved:

$$\begin{aligned}
\dot{W}_{\mathrm{tot},h} &= \frac{1}{N} \sum_{k=1}^{N-1} w_k \left(v_{\parallel,k} \dot{v}_{\parallel,k} + \mu_k \dot{\boldsymbol{\eta}}_k \cdot \nabla |\hat{B}_0| \right) + \boldsymbol{\phi}^\top \mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}} \mathbb{G}\, \dot{\boldsymbol{\phi}} + \frac{1}{Z\epsilon^2} \boldsymbol{\phi}^\top(t) \mathbb{M}^0_{\mathrm{ad}}\, \dot{\boldsymbol{\phi}} \\
&= \frac{1}{N} \sum_{k=1}^{N-1} w_k \left(v_{\parallel,k} \frac{1}{\epsilon} \frac{\hat{\mathbf{E}}^{*1} \cdot \hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}} + \mu_k \dot{\boldsymbol{\eta}}_k \cdot \nabla |\hat{B}_0| \right) + \frac{1}{\epsilon} \boldsymbol{\phi}^\top \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{1}{N} \sum_{k=0}^{N-1} w_k \boldsymbol{\Lambda}^0(\boldsymbol{\eta}_k(t)) - \int \hat{n}_0(\boldsymbol{\eta}) \boldsymbol{\Lambda}^0 \sqrt{g}\,\mathrm{d}\boldsymbol{\eta} \right) \\
&= \frac{1}{N} \sum_{k=1}^{N-1} w_k \left(v_{\parallel,k} \frac{1}{\epsilon} \frac{\hat{\mathbf{E}}^{*1} \cdot \hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}} + \mu_k \dot{\boldsymbol{\eta}}_k \cdot \nabla |\hat{B}_0| \right) + \boldsymbol{\phi}^\top \frac{1}{N\epsilon} \sum_{k=0}^{N-1} w_k \dot{\boldsymbol{\eta}}_k \cdot \nabla \boldsymbol{\Lambda}^0 \\
&= \frac{1}{N} \sum_{k=1}^{N-1} w_k \left(v_{\parallel,k} \frac{1}{\epsilon} \frac{\hat{\mathbf{E}}^{*1} \cdot \hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}} + \dot{\boldsymbol{\eta}}_k \cdot \mu_k \nabla |\hat{B}_0| + \frac{1}{\epsilon} \dot{\boldsymbol{\eta}}_k \cdot \nabla \hat{\phi} \right) \\
&= \frac{1}{N\epsilon} \sum_{k=1}^{N-1} w_k \left(v_{\parallel,k} \frac{\hat{\mathbf{E}}^{*1} \cdot \hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}} - \left(v_{\parallel,k} \frac{\hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}} + \frac{\hat{\mathbf{E}}^{*1} \times \hat{\mathbf{b}}^1_0}{\hat{B}^{*3}_{\parallel}} \right) \cdot \hat{\mathbf{E}}^{*1} \right) \\
&= 0\,.
\end{aligned}$$

4.5 Hamiltonian structure

It is straightforward to write the gyrokinetic Vlasov-Poisson system as a non-canonical Hamiltonian system in the variables $\mathbf{X} = (\boldsymbol{\eta}_k)_{k=0}^{N-1} \in \mathbb{R}^{3N}$ and $\mathbf{V}_\parallel = (v_{\parallel,k})_{k=0}^{N-1} \in \mathbb{R}^N$, where the electric potential $\phi(\mathbf{X})$ is viewed as a complicated function of $\mathbf{X}$ related to the inverse gyrokinetic Poisson equation. The discrete energy (4.33) is the Hamiltonian, written as

$$H(\mathbf{X}, \mathbf{V}_\parallel) = \frac{1}{2}\mathbf{V}_\parallel^\top \mathbb{W}\mathbf{V}_\parallel + \mathbf{w}^\top \mathbf{M}(\mathbf{X}) + \frac{1}{2}\boldsymbol{\phi}^\top(\mathbf{X})\mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}}\mathbb{G}\,\boldsymbol{\phi}(\mathbf{X}) + \frac{1}{2Z\epsilon^2}\boldsymbol{\phi}^\top(\mathbf{X})\mathbb{M}^0_{\mathrm{ad}}\,\boldsymbol{\phi}(\mathbf{X})\,, \tag{4.35}$$

where

$$\mathbb{W} = \mathrm{diag}(\mathbf{w}) \in \mathbb{R}^{N\times N}\,, \qquad \mathbf{w} = \left(\frac{w_k}{N}\right)_{k=0}^{N-1} \in \mathbb{R}^N \qquad \mathbf{M}(\mathbf{X}) = \left(\mu_k|\hat{B}_0|(\boldsymbol{\eta}_k)\right)_{k=0}^{N-1} \in \mathbb{R}^N\,.$$

Moreover, the discrete Poisson equation is written as

$$\mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}}\mathbb{G}\,\boldsymbol{\phi} + \frac{1}{Z\epsilon^2}\mathbb{M}^0_{\mathrm{ad}}\,\boldsymbol{\phi} = \frac{1}{\epsilon}\mathbb{L}^0(\mathbf{X})\mathbf{w} - \frac{\mathbf{n}_0}{\epsilon}\,. \tag{4.36}$$

with

$$\mathbf{n}_0 = \int \hat{n}_0(\boldsymbol{\eta})\boldsymbol{\Lambda}^0\sqrt{g}\,\mathrm{d}\boldsymbol{\eta} \in \mathbb{R}^{N_0}\,, \qquad \mathbb{L}^0(\mathbf{X}) \in \mathbb{R}^{N_0\times N}\,, \qquad \mathbb{L}^0_{ik}(\mathbf{X}) = \Lambda^0_i(\boldsymbol{\eta}_k) \in \mathbb{R}\,.$$

The gradient of the Hamiltonian (4.34) with respect to $\mathbf{X}$ can be computed with the help of the discrete Poisson equation (4.35),

$$\begin{aligned}\frac{\partial}{\partial \mathbf{X}}H(\mathbf{X}, \mathbf{V}_\parallel) &= \mathbb{N}\frac{\partial}{\partial \mathbf{X}}\mathbf{M}(\mathbf{X}) + \boldsymbol{\phi}^\top(\mathbf{X})\frac{\partial}{\partial \mathbf{X}}\left(\mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}}\mathbb{G}\,\boldsymbol{\phi} + \frac{1}{Z\epsilon^2}\mathbb{M}^0_{\mathrm{ad}}\,\boldsymbol{\phi}\right)\\ &= \mathbb{N}\frac{\partial}{\partial \mathbf{X}}\mathbf{M}(\mathbf{X}) + \frac{1}{\epsilon}\boldsymbol{\phi}^\top(\mathbf{X})\frac{\partial}{\partial \mathbf{X}}\mathbb{L}^0(\mathbf{X})\mathbf{w}\,.\end{aligned}$$

where $\mathbb{N} = \mathrm{diag}(\mathbf{w}) \otimes \mathbb{I}_{3\times 3} \in \mathbb{R}^{3N\times 3N}$ is a convenient way of expressing the multiplication with the weights. In order to compute the last term, let us view $\boldsymbol{\phi}^\top(\mathbf{X}) = \boldsymbol{\phi}^\top$ as independent of $\mathbf{X}$ to see the index contraction, then

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\eta}_i}(\frac{1}{\epsilon}\boldsymbol{\phi}^\top \mathbb{L}^0(\mathbf{X})\mathbf{w}) &= \frac{\partial}{\partial \boldsymbol{\eta}_i}\sum_{j,k}\phi_j \Lambda_j^0(\boldsymbol{\eta}_k) w_k \\ &= \frac{1}{\epsilon}\frac{\partial}{\partial \boldsymbol{\eta}_i}\sum_k \hat{\phi}(\boldsymbol{\eta}_k) w_k \\ &= \frac{1}{\epsilon}\sum_k \delta_{i,k}\nabla\hat{\phi}(\boldsymbol{\eta}_k) w_k \\ &= \frac{1}{\epsilon}\nabla\hat{\phi}(\boldsymbol{\eta}_i) w_i \,.\end{aligned}$$

Hence we arrive at

$$\frac{\partial}{\partial \mathbf{X}} H(\mathbf{X}, \mathbf{V}_\parallel) = \mathbb{N}\frac{\partial}{\partial \mathbf{X}}\mathbf{M}(\mathbf{X}) + \frac{1}{\epsilon}(\mathbb{G}\boldsymbol{\phi}(\mathbf{X}))^\top \mathbb{L}^1(\mathbf{X})\,\mathbb{N}$$

where

$$\mathbb{L}^1(\mathbf{X}) \in \mathbb{R}^{N_1 \times N \times 3}\,, \qquad \mathbb{L}^1_{ik} = \Lambda^1_i(\boldsymbol{\eta}_k) \in \mathbb{R}^3 \text{ (vector-valued basis functions)}\,.$$

With this we can compute the gradient of the Hamiltonian w.r.t $(\mathbf{X}, \mathbf{V}_\parallel)$,

$$\nabla H(\mathbf{X}, \mathbf{V}_\parallel) = \begin{pmatrix} \mathbb{N}\frac{\partial}{\partial \mathbf{X}}\mathbf{M}(\mathbf{X}) + \frac{1}{\epsilon}(\mathbb{G}\boldsymbol{\phi}(\mathbf{X}))^\top \mathbb{L}^1(\mathbf{X})\,\mathbb{N} \\ \mathbb{W}\mathbf{V}_\parallel \end{pmatrix}.$$

Hence, we are able to write the equations of motion (4.30)-(4.31) as a non-canonical Hamiltonian system:

$$\begin{pmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{V}}_\parallel \end{pmatrix} = \begin{pmatrix} \epsilon b_{0\times}(\mathbf{X}, \mathbf{V}_\parallel)\mathbb{N}^{-1} & \frac{\hat{\mathbf{B}}^{*2}(\mathbf{X}, \mathbf{V}_\parallel)}{\hat{B}^{*3}_\parallel(\mathbf{X}, \mathbf{V}_\parallel)}\,\mathbb{W}^{-1} \\ -\frac{\hat{\mathbf{B}}^{*2\top}(\mathbf{X}, \mathbf{V}_\parallel)}{\hat{B}^{*3}_\parallel(\mathbf{X}, \mathbf{V}_\parallel)}\,\mathbb{W}^{-1} & 0 \end{pmatrix} \nabla H(\mathbf{X}, \mathbf{V}_\parallel)\,, \tag{4.37}$$

where the diagonal block is the skew-symmetric rotation matrix

$$b_{0\times}(\mathbf{X}) = \mathrm{diag}\left(\frac{1}{\hat{B}^{*3}_\parallel(\boldsymbol{\eta}_k, v_{\parallel,k})} \begin{pmatrix} 0 & -b^1_{0,3}(\boldsymbol{\eta}_k) & b^1_{0,2}(\boldsymbol{\eta}_k) \\ b^1_{0,3}(\boldsymbol{\eta}_k) & 0 & -b^1_{0,1}(\boldsymbol{\eta}_k) \\ -b^1_{0,2}(\boldsymbol{\eta}_k) & b^1_{0,1}(\boldsymbol{\eta}_k) & 0 \end{pmatrix} \right) \in \mathbb{R}^{3N\times 3N}\,.$$

4.6 Time discretization

The insight that the gyrokinetic Vlasov-Poisson system has the Hamiltonian structure (4.35) can help with the time discretization. We could leverage many techniques for discretizing non-canonical Hamilto-

nian ODEs (also called "Poisson systems") that lead to structure-preserving algorithms. Unfortunately, our system has the most general form of a Poisson system; denoting $\mathbf{Z} = (\mathbf{X}, \mathbf{V}_\parallel)$ we have

$$\dot{\mathbf{Z}} = \mathbb{J}(\mathbf{Z}) \nabla H(\mathbf{Z}) \,,$$

where the Poisson matrix $\mathbb{J}(\mathbf{Z})$ depends non-linearly on $\mathbf{Z}$. A Poisson integrator (i.e. a Poisson map) that respects all Casimirs of $\mathbb{J}(\mathbf{Z})$ will thus be very hard to obtain, and we don't know how exactly to do that (it is not the purpose of this thesis). However, we can still go for exact energy conservation with the discrete gradient method. For this, our first strategy will be to split the Poisson matrix as

$$\mathbb{J}(\mathbf{Z}) = \mathbb{J}_1(\mathbf{Z}) + \mathbb{J}_2(\mathbf{Z})$$

such that

$$\mathbb{J}_1(\mathbf{Z}) = \begin{pmatrix} \epsilon b_{0\times}(\mathbf{Z}) \mathbb{N}^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbb{J}_2(\mathbf{Z}) = \begin{pmatrix} 0 & \frac{\hat{\mathbf{B}}^{*2}(\mathbf{Z})}{\hat{B}^{*3}_\parallel(\mathbf{Z})} \mathbb{W}^{-1} \\ -\frac{\hat{\mathbf{B}}^{*2\top}(\mathbf{Z})}{\hat{B}^{*3}_\parallel(\mathbf{Z})} \mathbb{W}^{-1} & 0 \end{pmatrix}.$$

The first splitting step of our scheme is thus a discrete version of

$$\begin{pmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{V}}_\parallel \end{pmatrix} = \begin{pmatrix} \epsilon b_{0\times}(\mathbf{Z}) \mathbb{N}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \nabla H(\mathbf{Z}) \,, \tag{4.38}$$

which corresponds to

$$\dot{\boldsymbol{\eta}}_k = \frac{\hat{\mathbf{E}}^{*1}(\boldsymbol{\eta}_k) \times \hat{\mathbf{b}}^1_0(\boldsymbol{\eta}_k)}{\hat{B}^{*3}_\parallel(\boldsymbol{\eta}_k, v_{\parallel,k})} \,, \qquad \hat{\mathbf{E}}^{*1} = -\nabla \hat{\phi} - \epsilon \mu_k \nabla |\hat{B}_0| \,.$$

The second splitting step is a discrete version of

$$\begin{pmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{V}}_\parallel \end{pmatrix} = \begin{pmatrix} 0 & \frac{\hat{\mathbf{B}}^{*2}(\mathbf{Z})}{\hat{B}^{*3}_\parallel(\mathbf{Z})} \mathbb{W}^{-1} \\ -\frac{\hat{\mathbf{B}}^{*2\top}(\mathbf{Z})}{\hat{B}^{*3}_\parallel(\mathbf{Z})} \mathbb{W}^{-1} & 0 \end{pmatrix} \nabla H(\mathbf{Z}) \,, \tag{4.39}$$

which corresponds to

$$\dot{\boldsymbol{\eta}}_k = v_{\parallel,k}\frac{\hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}}\,, \qquad \hat{\mathbf{B}}^{*2} = \hat{\mathbf{B}}^2_0 + \epsilon v_{\parallel,k}\nabla\times\hat{\mathbf{b}}^1_0\,, \qquad \hat{B}^{*3}_{\parallel} = |\hat{B}^3_0| + \sqrt{g}\epsilon v_{\parallel,k}\hat{\mathbf{b}}^1_0\cdot(\nabla\times\hat{\mathbf{b}}_0)_{vec} \tag{4.40}$$

$$\dot{v}_{\parallel,k} = \frac{1}{\epsilon}\frac{\hat{\mathbf{E}}^{*1}\cdot\hat{\mathbf{B}}^{*2}}{\hat{B}^{*3}_{\parallel}}\,, \qquad \hat{\mathbf{E}}^{*1} = -\nabla\hat{\phi} - \epsilon\mu_k\nabla|\hat{B}_0|. \tag{4.41}$$

4.7 Discrete Gradient Method

The discrete gradient method features exact energy conservation. We shall implement the Gonzalez discrete gradient method:

$$\frac{\mathbf{Z}^{n+1}-\mathbf{Z}^n}{\Delta t} = \mathbb{J}(\mathbf{Z}^n)\overline{\nabla}H(\mathbf{Z}^{n+1},\mathbf{Z}^n)\,, \tag{4.42}$$

where

$$\overline{\nabla}H(\mathbf{Z}^{n+1},\mathbf{Z}^n) = \nabla H(\mathbf{Z}^{n+1/2}) + (\mathbf{Z}^{n+1}-\mathbf{Z}^n)\frac{H(\mathbf{Z}^{n+1}) - H(\mathbf{Z}^n) - (\mathbf{Z}^{n+1}-\mathbf{Z}^{n+1})^\top\nabla H(\mathbf{Z}^{n+1/2})}{||\mathbf{Z}^{n+1}-\mathbf{Z}^n||}\,, \tag{4.43}$$

and $\mathbf{Z}^{n+1/2} = (\mathbf{Z}^{n+1}+\mathbf{Z}^n)/2$. Like all discrete gradients, but here particularly easy to see, the Gonzalez discrete gradient satisfies

$$(\mathbf{Z}^{n+1}-\mathbf{Z}^{n+1})\overline{\nabla}H(\mathbf{Z}^{n+1},\mathbf{Z}^n) = H(\mathbf{Z}^{n+1}) - H(\mathbf{Z}^n)\,,$$

which immediatly implies exact conservation of H through the skew symmetry of $\mathbb{J}$.

5 Implementation of the Gyrokinetic Model in Struphy

After discretizing the model, the next step is implementing the PDE model in Struphy. One should follow Struphy's framework in order to find the desired solutions. Here, we will briefly discuss the steps of implementation.

The first step is to choose particle species, bulk species, and velocity scale. Particle species are the unknowns of the model, and Struphy has three choices: electromagnetic fields, fluid species, and kinetic species. In the DriftKineticElectrostaticAdiabatic model, there are two unknowns which are ϕ and particles. As defined in the discretization $\phi \in H^1$ and in the electromagnetic field, and with the help of the averaging step, the particles are in the kinetic field and 5-dimensional. Then, one should choose bulk species. In our case, the bulk species must return the name of one of the model species, which is ions for particles. Next, it is necessary to choose the velocity scale. This selection corresponds to setting the velocity unit of normalization $\hat{v}$. This then sets the time unit $\hat{t} = \frac{\hat{x}}{\hat{v}}$ based on $\hat{x}$, where the length unit is determined via the parameter file. A new velocity scale, "thermal" is added to Struphy for DriftKineticElectrostaticAdiabatic model. It is defined as

$$\hat{v}_{th} = \sqrt{\frac{k_B T}{m}}$$

where $k_B T$ is the unit for internal energy.

Furthermore, the next step of the implementation is to add propagators. Propagators are the main building blocks of the models in Struphy. Time stepping and calculations for each time step is done by the propagator. In the gyrokinetic model four propagators are used.

The first propagator solves gyrokinetic Poisson equation and it is in propagator fields class of Struphy. Implicit diffusion propagator is used as we used it in a similar purpose in Poisson Model. It is adapted to use in gyrokinetic model because at first in the left hand side of the propagator there were only one term to solve poisson or heat equations but in the gyrokinetic Poisson we have both terms containing both

M^0_{ad} and M^1_{gyro}. Therefore implicit diffusion operator is changed as below so that it can be adapted for more general cases:

$$\Big(\frac{\sigma_1}{\Delta t}\mathbb{M}^0_{n_0} + \mathbb{G}^\top \mathbb{M}^1_{D_0}\mathbb{G}\Big)\boldsymbol{\phi}^{n+1} = \frac{\sigma_2}{\Delta t}\mathbb{M}^0_{n_0}\boldsymbol{\phi}^n + \frac{\sigma_3}{\Delta t}\sum_i (\Lambda^0, \rho_i)_{L^2}$$

where $M^0_{n_0}$ and $M^1_{D_0}$ are in class WeightedMassOperators and $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{R}$ are artificial parameters that can be tuned to change the model. Here are three common usage options :

1. $\sigma_1 = \sigma_2 = 0$ and $\sigma_3 = \Delta t$ is Poisson solver with a given charge density $\sum_i \rho_i$.

2. $\sigma_2 = 0$ and $\sigma_1 = \sigma_3 = \Delta t$ is Poisson solver with adiabatic electrons.

3. $\sigma_1 = \sigma_2 = 1$ $\sigma_3 = 0$ and is Implicit heat equation solver.

Second option is used in gyrokinetic poisson equation.
Afterwards PushDriftKineticbxEstar is added to the propagator markers class for pushing step of the guiding center drift part $\mathbf{b}_\times \mathbf{E}^*$ where equations of motion are :

$$\dot{\mathbf{X}} = \frac{1}{B^*_\parallel}\mathbf{E}^* \times \mathbf{b}_0 \tag{5.1}$$

$$\dot{v}_\parallel = 0. \tag{5.2}$$

Here are the implemented marker updates in the logical domain for each marker position p:

$$\dot{\boldsymbol{\eta}}_p = \frac{1}{\sqrt{g}\,\hat{B}^*_\parallel(\boldsymbol{\eta}_p, v_{\parallel,p})}\hat{\mathbf{E}}^{*1}(\boldsymbol{\eta}_p, \mu_p) \times \hat{\mathbf{b}}^1_0(\boldsymbol{\eta}_p)\,, \tag{5.3}$$

$$\dot{v}_{\parallel,p} = 0\,. \tag{5.4}$$

Then PushDriftKineticParallel is added for pushing step of the guiding center drift part $\mathbf{B}^*$ where equations of motion are :

$$\dot{\mathbf{X}} = v_\parallel \frac{\mathbf{B}^*}{B^*_\parallel}\,, \tag{5.5}$$

$$\dot{v}_\parallel = \frac{1}{\epsilon}\frac{\mathbf{B}^*}{B^*_\parallel} \cdot \mathbf{E}^*\,. \tag{5.6}$$

Here are the implemented marker updates in the logical domain for each marker position p:

$$\dot{\boldsymbol{\eta}}_p = \frac{1}{\sqrt{g}\,\hat{B}^*_{\parallel}(\boldsymbol{\eta}_p, v_{\parallel,p})}\hat{\mathbf{B}}^{*2}(\boldsymbol{\eta}_p, v_{\parallel,p})\, v_{\parallel,p}\,, \tag{5.7}$$

$$\dot{v}_{\parallel,p} = \frac{1}{\sqrt{g}\,\hat{B}^*_{\parallel}(\boldsymbol{\eta}_p, v_{\parallel,p})}\hat{\mathbf{E}}^{*1}(\boldsymbol{\eta}_p, v_{\parallel,p}) \cdot \hat{\mathbf{B}}^{*2}(\boldsymbol{\eta}_p, v_{\parallel,p})\,. \tag{5.8}$$

There were similar propagators which are pushing $\mathbf{b}_\times\mathbf{E}^*$ and $\mathbf{B}^*$ however neither of them contains $\nabla\phi$ in the electric field computation which is the most important difference of the model. It takes ϕ from the solution of the implicit diffusion propagator then PushDriftKineticParallel and PushDriftKineticbxEstar use this ϕ in the calculations. Propagators in Struphy help us to choose which scheme to use. In the gyrokinetic models explicit schemes are implemented such as Runge-Kutta and Euler.

Then, desired calculations of the fields are done in the kernels of Struphy where these kernels are linked in the propagators. $\mathbf{B}^*$, $\mathbf{B}^*_{\parallel}$ and $\mathbf{E}^*$ are calculated as well as they are updated for each step in the kernels.

Fourth propagator is a variation for implicit diffusion propagator to have another choice namely only with the adiabatic electrons and without the gyration. It is implemented with minor changes on implicit diffusion which solves the following eqaution :

$$\int_\Omega \psi\, n_0(\mathbf{x})\phi\, \mathrm{d}\mathbf{x} = \sum_i \int_\Omega \psi\,(\rho_i(\mathbf{x}) - \rho_0(\mathbf{x}))\,\mathrm{d}\mathbf{x} \qquad \forall\, \psi \in H^1\,,$$

This operator in general can be used for the following type of equations

$$\sigma_1 \mathbb{M}^0_{n/T}\boldsymbol{\phi} = (\Lambda^0, n_e - n_{e0})_{L^2}.$$

In order to use these propagators and define $\hat{B}^*_{\parallel}$, $\hat{\mathbf{B}}^{*2}$, $\hat{\mathbf{E}}^{*1}$, $\hat{\mathbf{b}}^1_0$ MHD or Braginskii equilibrium is used. We took projectors of the each required element in the required form. They are pre-defined in Struphy `https://struphy.pages.mpcdf.de/struphy/sections/subsections/mhd_equils.html#mhd-equilibria`, `https://struphy.pages.mpcdf.de/struphy/sections/subsections/braginskii_equils.html`.

Moreover, an important step is to add control variate method which can be selected by user in the pa-

rameter file. This is a noise reduction technique that is used to have numerically cheaper calculations of the models. Here we rewrite the discrete gyrokinetic poisson :

$$-\nabla_\perp \cdot \left(\frac{n_0}{|B_0|^2}\right)\nabla_\perp \phi + \frac{n_0}{Z\epsilon^2}\phi = \frac{1}{\epsilon}\int f B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu - \frac{n_0}{\epsilon}. \tag{5.9}$$

Control variate method gets rid of the last term $\frac{n_0}{\epsilon}$ and calculates an approximation for f by writing $f = f - f_0 + f_0$ and using quasi-neutrality such that

$$n_0 = Z\int f_0 B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu.$$

Then we can get

$$\begin{aligned}
-\nabla_\perp \cdot \left(\frac{n_0}{|B_0|^2}\right)\nabla_\perp \phi + \frac{n_0}{Z\epsilon^2}\phi &= \frac{1}{\epsilon}\left(\int f B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu - \int f_0 B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu\right) \\
&= \frac{1}{\epsilon}\left(\int (f - f_0 + f_0) B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu - \int f_0 B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu\right) \\
&= \frac{1}{\epsilon}\int (f - f_0) B_\parallel^* \, \mathrm{d}v_\parallel \mathrm{d}\mu.
\end{aligned}$$

It is users' choice in gyrokinetic model to use either full f or control variate method but control variate is numerically easier to get as shown above. This method is pre-defined in Struphy and can be used by update weight method of particles `https://struphy.pages.mpcdf.de/struphy/sections/subsections/pic_classes.html#struphy.pic.base.Particles.update_weights`.

After the propagators, scalar variables are added to the model to see the changes in energy. We have energy conservation only in semi discrete form or with the use of discrete gradient method. Therefore the total energy

$$H(\mathbf{X}, \mathbf{V}_\parallel) = \frac{1}{2}\mathbf{V}_\parallel^\top \mathbb{W}\mathbf{V}_\parallel + \mathbf{w}^\top \mathbf{M}(\mathbf{X}) + \frac{1}{2}\boldsymbol{\phi}^\top(\mathbf{X})\mathbb{G}^\top \mathbb{M}^1_{\mathrm{gyro}}\mathbb{G}\,\boldsymbol{\phi}(\mathbf{X}) + \frac{1}{2Z\epsilon^2}\boldsymbol{\phi}^\top(\mathbf{X})\mathbb{M}^0_{\mathrm{ad}}\,\boldsymbol{\phi}(\mathbf{X})$$

is calculated for each step to see energy changes in time.

6 Tests of the Gyrokinetic Model

Different scenarios for the gyrokinetic model have been tested and results are presented below. The first one is the landau damping test which is the different kind of fluctuations of different particles in plasma. Some of the particles in the plasma move slightly faster than the wave, and the interaction slows down these particles, but in the opposite case, particles moving slower than the wave gain energy from the wave and accelerate. Due to the asymmetry in the velocity distribution of the particles, more energy is transferred from the wave to the particles, leading to a net damping effect [12]. In the graph below we can see the change in amplitude in time and our model having the same frequency as exact solution.

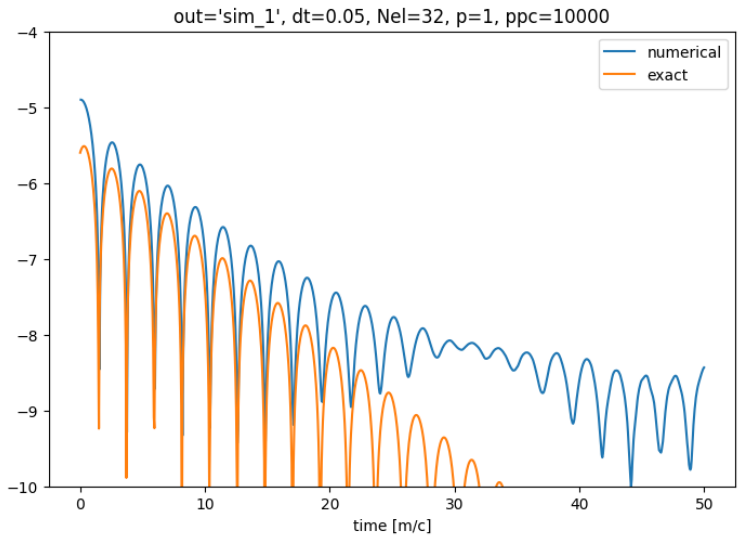

Figure 6.1 Landau damping.

Another result below shows the figures both full f and control variate methods. Cosine perturbation is selected and we can see the cosine function with minor errors.

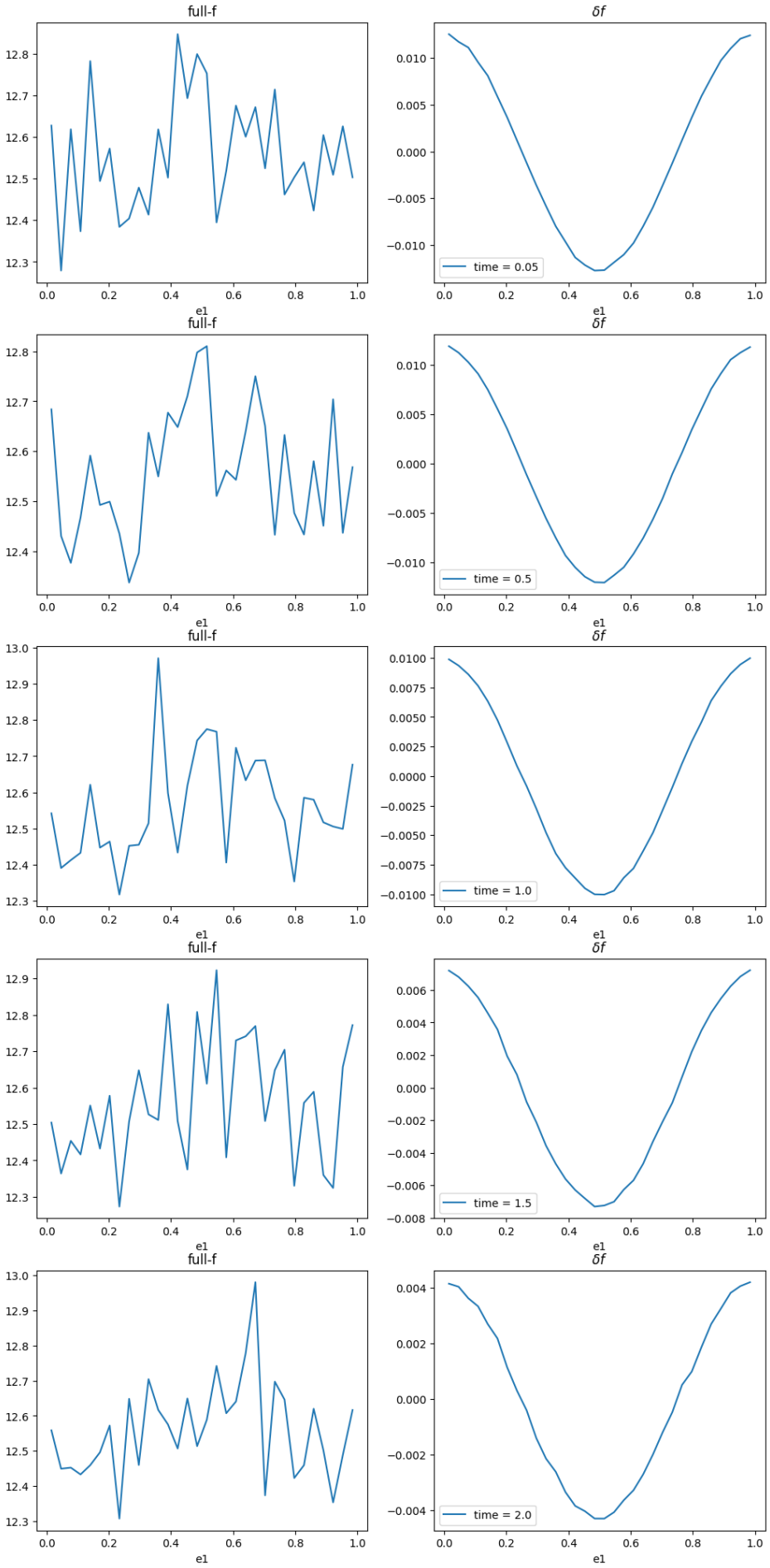

Figure 6.2 full-f vs δf.

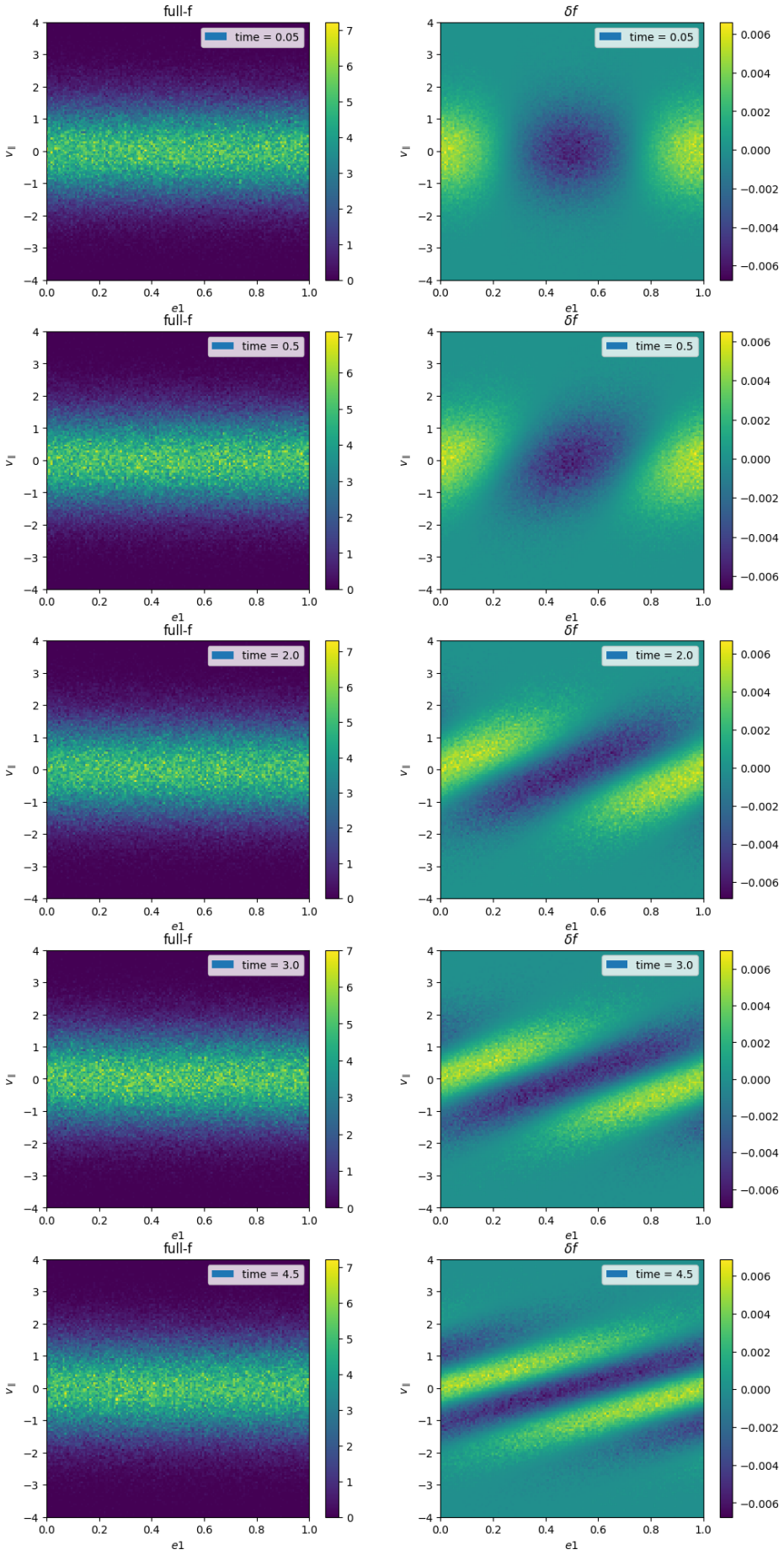

Figure 6.3 $v_{\parallel}$ vs e_1.

7 Conclusion

This thesis is focused numerical analysis and implementation of Poisson model and gyrokinetic turbulance model with adiabatic electrons. The successful discretization and implementation of the Poisson and the gyrokinetic model highlighted several key achievements.

Firstly, Poisson equation is a fundamental PDE in mathematical physics and the initial part of the thesis concentrated on the Poisson equation and we aimed to achieve a high-order accurate solution. Using B-splines, we ensured that the order of convergence is degree plus one. This approach has demonstrated good accuracy and computational efficiency compared to lower-order methods. The implementation details, numerical experiments, and convergence analysis confirmed the robustness of the spline-based discretization technique.

Secondly, the analysis and implementation of the gyrokinetic model is the main topic in this thesis. Model equations are normalized according to the structure in Struphy and then analysed using the FEEC and PIC frameworks. The equations of motion were derived from the discretized gyrokinetic model with the help of the Lagrangian and energy of the model is provided by the Hamiltonian. Struphy provided a framework to simulate the dynamics of plasma particles. The numerical implementation was carried out with attention to maintaining the accuracy and stability of the solution. The successfull implementation of the model also shows us Struphy is an accurate and efficient tool and all the code can be found in `https://gitlab.mpcdf.mpg.de/struphy/struphy`. Due to limited time discrete gradient method couldn't implemented. This implicit scheme reaches full energy conservation so will be useful in further research.

In conclusion, this thesis has successfully addressed the challenges of discretizing and implementing the Poisson equation and a gyrokinetic model. The methods developed and implemented provide a framework for future studies in gyrokinetic and related fields.

List of Figures

Bibliography

[1] I.B. Bernstein and S.K. Trehan. “Plasma oscillations (I)”. In: *Nuclear Fusion* 1.1 (Sept. 1960), p. 3. DOI: 10.1088/0029-5515/1/1/002. URL: https://dx.doi.org/10.1088/0029-5515/1/1/002.

[2] A. Buffa et al. “Isogeometric Discrete Differential Forms in Three Dimensions”. In: *SIAM Journal on Numerical Analysis* 49.2 (2011), pp. 818–844. DOI: 10.1137/100786708. eprint: https://doi.org/10.1137/100786708. URL: https://doi.org/10.1137/100786708.

[3] Lionel Cheng et al. “Using neural networks to solve the 2D Poisson equation for electric field computation in plasma fluid simulations”. In: *arXiv preprint arXiv:2109.13076* (2021).

[4] Theodore Frankel. “Contents”. In: *The Geometry of Physics: An Introduction*. Cambridge University Press, 2011, pp. vii–xviii.

[5] Tristan Johnson. “Noether’s Theorem: Symmetry and Conservation”. In: (2016).

[6] G Knorr and J Nuehrenberg. “The adiabatic electron plasma and its equation of state”. In: *Plasma Physics* 12.12 (Dec. 1970), p. 927. DOI: 10.1088/0032-1028/12/12/003. URL: https://dx.doi.org/10.1088/0032-1028/12/12/003.

[7] Michael Kraus et al. “GEMPIC: geometric electromagnetic particle-in-cell methods”. In: *Journal of Plasma Physics* 83.4 (2017), p. 905830401. DOI: 10.1017/S002237781700040X.

[8] John A Krommes. “The gyrokinetic description of microturbulence in magnetized plasmas”. In: *Annual review of fluid mechanics* 44 (2012), pp. 175–201.

[9] Nozomi Magome et al. “Higher-continuity s-version of finite element method with B-spline functions”. In: *Journal of Computational Physics* 497 (2024), p. 112593. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2023.112593. URL: https://www.sciencedirect.com/science/article/pii/S0021999123006885.

[10] William A Newcomb. *Lagrangian and Hamiltonian methods in magnetohydrodynamics*. Tech. rep. California Univ., Livermore (USA). Lawrence Livermore Lab., 1961.

[11] Alexander Piel. “An introduction to laboratory, space, and fusion plasmas”. In: *Plasma Physics* (2010).

[12] P. Stubbe and A. I. Sukhorukov. “On the physics of Landau damping”. In: *Physics of Plasmas* 6.8 (Aug. 1999), pp. 2976–2988. ISSN: 1070-664X. DOI: 10.1063/1.873584. eprint: https://pubs.aip.org/aip/pop/article-pdf/6/8/2976/19038911/2976_1_online.pdf. URL: https://doi.org/10.1063/1.873584.

[13] Hideo Sugama. “Gyrokinetic field theory”. In: *Physics of Plasmas* 7.2 (2000), pp. 466–480.