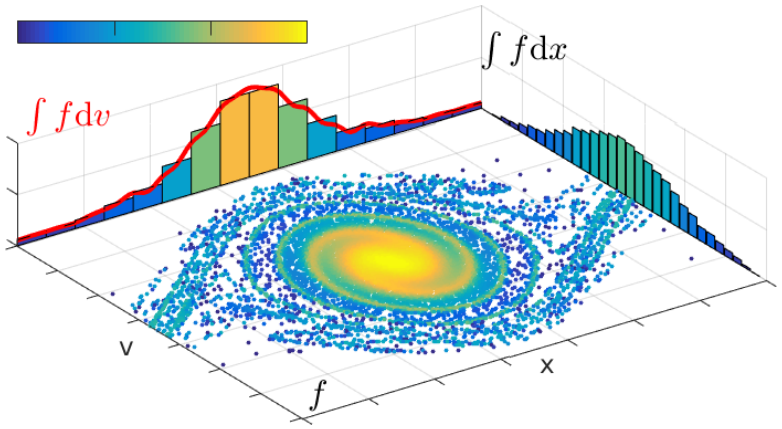
Jakob Ameres

Stochastic and Spectral Particle Methods for Plasma Physics



$$\delta w_n = \frac{f(x_n,v_n,0) - \alpha h(x_n,v_n)}{g(x_n,v_n,0)}$$

$$\hat{\rho}_k = \int \rho(x) \mathrm{e}^{-\mathrm{i}k\cdot x} \mathrm{d}x = \mathbb{E}\left[\mathrm{e}^{-\mathrm{i}k\cdot X}\right]$$

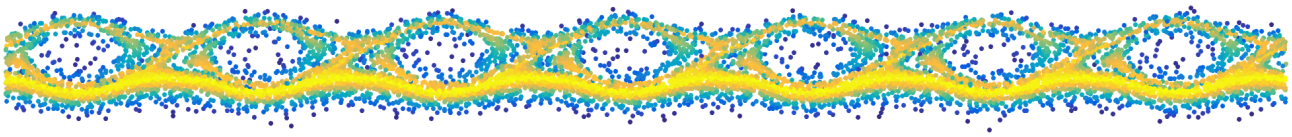
$$\mathcal{H} = \mathcal{H}_E + \mathcal{H}_B + \mathcal{H}_p$$

$$\mathbb{V}[X] = \mathbb{E}\left[\mathbb{V}[X|\mathcal{H}]\right] + \mathbb{V}\left[\mathbb{E}[X|\mathcal{H}]\right]$$

$$\mathbb{V}\left[\mathrm{e}^{-\mathrm{i}k\cdot X}\right]$$

$$\frac{1}{2}\int f(x,v,t) v^2 \, \mathrm{d}x\mathrm{d}v = \mathbb{E}\left[\frac{1}{2}V^2\right]$$

$$\mathcal{P}\left(\lim_{N_p\to\infty} \frac{1}{N_p}\sum_{n=1}^{N_p} X_n = \mathbb{E}[X]\right) = 1$$



Technische Universität München
Zentrum Mathematik
Numerische Methoden der Plasmaphysik

Stochastic and Spectral Particle Methods for Plasma Physics

Jakob Ameres

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Oliver Junge
Prüfer der Dissertation:	1.	Prof. Dr. Eric Sonnendrücker
	2.	Prof. Dr. Caroline Lasser
	3.	Prof. Jan S. Hesthaven, Ph.D. (schriftliche Beurteilung)

Die Dissertation wurde am 19.10.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 14.02.2018 angenommen.

Abstract

Particle methods are very popular for the discretization of kinetic equations like the electrostatic Vlasov–Poisson or the electromagnetic Vlasov–Maxwell system. They are easy to implement and embarrassingly parallel. In plasma physics the high dimensionality (6D) of the problems raises the costs of grid based codes, favoring the mesh free transport with particles and its inherent adaptivity by following characteristics. The Particle-in-Cell (PIC) scheme is a Monte Carlo method that couples the particle density to a grid based field solver. This introduces an error that is comprised of three components: the time discretization error, the field discretization error (bias) and the particle noise, given as the variance of a Monte Carlo estimator.

This work discusses the application of stochastic methods providing a setting in which the random particle noise is quantified and reduced, which includes measures of entropy and variance propagation for the field solver also in unstructured grids. For variance reduction control variates based on parametric shape functions and spectral expansions yield a significant noise reduction. Solvers of kinetic models are improved using reduced fluid descriptions by a purely particle based control variate scheme called Multilevel Monte Carlo. Conditional Monte-Carlo is used to justify variance reducing coarse graining techniques for Fokker–Planck collisions discretized as Ornstein Uhlenbeck process, and to improve control variates by stratification. Such combinations between Monte Carlo and other quadrature rules improve the particle gyroaverage and also the noise in linearized systems.

For physics governed by a small number of Fourier modes the mesh free Particle-in-Fourier (PIF) method is presented, which conserves energy and momentum. It has a more favorable bias residing in Fourier space and exhibits different computational demands since every particle contributes to every Fourier mode. The superb conservation and stability properties of PIF are demonstrated for electrostatic Vlasov–Poisson and fully electromagnetic multispecies Vlasov–Maxwell in multiple dimensions. Given the simplicity of PIF, benchmarks on CPU and GPU using different interpreters (fortran, julia, python, MATLAB) are presented. The relation between variance reducing Fourier filtered PIC, which is subject to aliasing and the aliasing free PIF is discussed. Also PIC and PIF can be mixed in order to efficiently describe the mode structure of physical instabilities in the curved geometry of toroidal fusion devices in arbitrary curvilinear coordinates. With the Legendre and Chebyshev polynomials the concept of spectral particle methods is fully generalized.

PIF and Eulerian solvers based on Fourier-spectral phase space share the same field discretization thus being suitable for a comparison between Lagrangian and Eulerian methods. For verification of the corresponding PIF implementation, a new fully Fourier-spectral Vlasov–Maxwell solver based on a Hamiltonian splitting scheme is presented.

Zusammenfassung

Für die Diskretisierung kinetischer Gleichungen, wie die des elektrostatischen Vlasov–Poisson oder des elektromagnetischen Vlasov–Maxwell Systems sind Teilchenmethoden nach wie vor die erste Wahl. Sie sind einfach zu implementieren und intrinsisch parallel. In der Plasmaphysik sind gitterbasierte Methoden aufgrund der hohen Dimensionalität der betrachteten Probleme mit deutlich höheren Kosten verbunden. Dies begünstigt den gitterlosen Transport mit Teilchen entlang der Charakteristiken, welcher dadurch zugleich inhärent adapativ ist. Das Particle-in-Cell (PIC) Verfahren ist eine Monte Carlo Methode, welche die Teilchendichte an einen gitterbasierten Feldlöser koppelt. Diese Kopplung ist eine Fehlerquelle die aus drei Komponenten besteht: dem Zeitdiskretisierungsfehler, dem Felddiskretisierungsfehler (auch systematischer Fehler) und dem Teilchenrauschen, welches durch die Varianz des Monte Carlo Schätzers gegeben ist.
Diese Arbeit diskutiert die Anwendung von stochastischen Methoden, mit denen das zufällige Teilchenrauschen quantifiziert und reduziert werden kann. Diese Methodik beinhaltet Entropieschätzer als auch Varianzpropagationsverfahren für Feldlöser, welche auch auf unstrukturierte Gitter anwendbar sind. Zur Varianzreduktion werden Control Variates verwendet, welche auf sowohl auf parametrisierten Funktionen als auch spektralen Entwicklungen basieren, und das Teilchenrauschen signifikant vermindern. Ein ausschließlich auf Teilchen basierendes Control Variate Verfahren, genannt Multilelevel Monte Carlo, kann Fluidmodelle an kinetische Modelle koppeln und somit das Lösen letzterer erleichtern. Mit bedingtem Monte Carlo können varianzreduzierende coarse graining Verfahren für als Ornstein Uhlenbeck Prozess diskretisierte Fokker–Planck Kollisionen gerechtfertigt werden. Dazu gehört auch die Verbesserung von Control Variates durch Stratifikation. Derartige Kombinationen zwischen Monte Carlo und anderen Quadraturregeln verbessern auch die Gyromittelung über Teilchen und vermindern das Rauschen in linearisierten Systemen.
Für Physik, die sich durch eine geringe Anzahl von Fouriermoden charakterisieren lässt, wird das gitterfreie Particle-in-Fourier (PIF) eingeführt, welches zugleich Energie und Impuls erhält. Sein systematischer Fehler liegt im Fourierraum und es stellt eine andere rechnerische Herausforderung als PIC dar, da jedes Teilchen zu jeder Fouriermode beiträgt. Die herausragenden Erhaltungs- und Stabilitätseigenschaften von PIF werden anhand des elektrostatischen Vlasov–Poisson und elektromagnetischen Vlasov–Maxwell mit Ionen und Elektronen in verschiedenen Dimensionen nachgewiesen. Aufgrund der Schlichtheit von PIF werden Vergleichstests auf CPU und GPU unter verschiedenen Interpretern (fortran, julia, python, MATLAB) gezeigt. Auch wird der Unterschied zwischen Fourier gefiltertem und damit varianzreduziertem PIC, welches noch Aliasing aufweist, und PIF diskutiert. Um die Modenstruktur von physikalischen Instabilitäten in toroidalen Fusionsgeräten in beliebigen krummlinigen Koordinaten effizient zu beschreiben wird PIC und PIF kombiniert. Mit den Legendre und Chebyshev Polynomen lässt sich dann das Konzept der spektralen Teilchenmethoden verallgemeinern. PIF und eulersche Löser, welche auf einer Fourier-spektralen Diskretisierung des Phasenraums basieren haben diesselbe Felddiskretisierung, was einen Vergleich zwischen Lagrange und Euler Methoden ermöglicht. Zur Verifizierung der Ergebnisse durch PIF wird mit einem hamiltonischen Splittingverfahren ein neuer voll Fourier-spektraler Vlasov–Maxwell Löser vorgestellt.

Contents

Chapter 1.

Introduction

In a plasma, the fourth state of matter, atoms are decomposed into their nuclei and electrons by a violent environment typically at a very high temperature. Examples include the sun, the vastness of interstellar space, hot flames and fusion experiments here on earth. The motion of the charged particles in such an ionized gas is subject to the Lorentz force stemming from the surrounding magnetic $B(x)$ and electric field $E(x)$. The trajectory of a particle with velocity v, position x, charge q and mass m is described by the following system of ordinary differential equations

$$\dot{x} = v, \quad \dot{v} = \frac{q}{m}\left(E(x) + v \times B(x)\right).$$

When considering an abundance of particles (more than you can imagine) it is more convenient to describe the evolution of their respective density f by the Vlasov equation,

$$\partial_t f(x,v,t) + v \cdot \nabla_x f(x,v,t) + \frac{q}{m}\left[E(x,t) + v \times B(x,t)\right] \cdot \nabla_v f(x,v,t) = 0.$$

This plain advection seems to pose no greater mathematical difficulty if it would not be for the fact that the particles generate their own electric and magnetic field. We know from Gauss Law that any charged particle generates an electric field E depending on its position x. On the other hand Amperes Law states that any movement of a charge generates a magnetic field B depending on the velocity v and position x. This introduces a nonlinear coupling between the time evolution of the fields E and B and the density f, staging our mathematical challenge. A more sophisticated model, including electric and magnetic fields, couples the Vlasov equation to the electromagnetic Maxwell's equations. Neglecting the magnetic contribution yields the purely electrostatic Vlasov–Poisson system for electrons($\gamma = -1$) with the electrostatic potential Φ obtained by the Poisson equation.

$$\partial_t f + v\partial_x f + \partial_x \Phi \partial_v f = 0$$
$$\gamma \partial_{xx} \Phi = 1 - \int f \mathrm{dv}$$

Depending on the sign of γ the model changes from the small scale electron dynamics subject to the Coulomb force ($\gamma = -1$) to the large scale physics of interstellar gas clouds dominated by gravity ($\gamma = 1$). Thus, despite its simplicity the Vlasov–Poisson system is worthwhile investigating, since it exhibits unintuitive behavior like collision-less Landau damping [1][2].

1.1. Particle methods

For numerical resolution the phase space density f can be approximate on a grid yielding Eulerian [3],[4] methods. Approximation of f by a density of markers results in Lagrangian particle methods, see fig. 1.1. Three spatial and tree velocity components raise the dimensionality of the problems in computational plasma physics to six, rendering grid-based methods

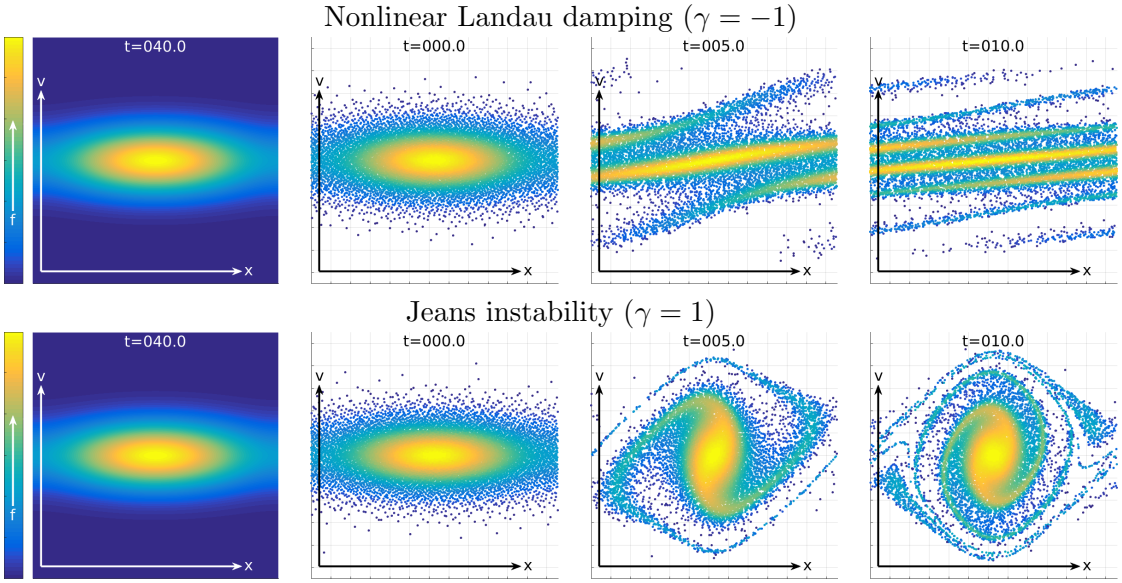


Figure 1.1.: Lagrangian particle simulation of the one-dimensional Vlasov–Poisson system (with PIC). The phase space markers transport the value of the initial condition $f(x, v, t = 0)$ along the trajectories given by the characteristics of the Vlasov equation. For identical initial conditions, a mere sign change in γ lifts the simulation from the microscopic scale of a damped electron Langmuir wave to the vast scale of interstellar gas clouds' gravitational collapse.

less attractive due to the curse of dimensionality. Independent of the dimension, Lagrangian particles provide diffusion free transport as they carry the values of the distribution f, such that the favorite method for kinetics of plasmas is Particle-In-Cell (PIC) [5],[6]. In general particle methods are very popular when it comes to the discretization of kinetic equations. They are easy to implement and embarrassingly parallel. Viewing the computational particles as macro-particles allows physicists an intuitive and descriptive modeling of physical processes although this approach is formally incorrect, since e.g. the Vlasov–Poisson system is merely a model for many particles represented by a density.[1] In PIC the density f is described by Monte Carlo markers called particles and coupled to a cell-based solver for the Poisson or Maxwell's equation, hence the name Particle **in** Cell. Every marker representing a volume of phase space projects its charge onto a spatial grid, which corresponds to a marginal density estimate of the density f. Once values on the grid are known, the desired field equations (Poisson or Maxwell) are typically solved with a finite difference approximation. The obtained fields are then used to advance the markers for a short time step by solving the respective equations of motion. Obtaining the spatially varying fields involves taking the Monte Carlo integral over the velocity space, see fig. 1.2. In general the Monte Carlo error — the variance — diminishes independently of the dimension with $\frac{1}{\sqrt{N_p}}$, where N_p is the number of particles. Yet the variance itself again depends on the dimension, which introduces the course of dimensionality also to PIC. The Monte Carlo error is of random nature such that it is often observed and simply referred to as noise.
Hybrid methods like the Semi-Lagrangian scheme (SL) combine interpolation with Lagrangian markers in order to benefit from the Lagrangian transport and the noiseless grid-based integration[7]. Lagrangian transport refers to the fact that every particle transports an initial value of f depicted as its color in figs. 1.1 and 1.2. Instead of interpolating this color onto a

[1] For problems where the number of particles remains small, e.g. the celestial bodies in the solar system but not in a plasma, particle-particle methods solve the actual N-body problem.

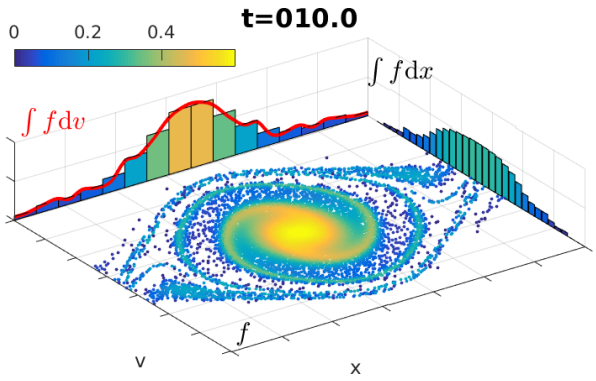


Figure 1.2.: The marginal density $\int f(x,v)\mathrm{d}v$ required for the right hand side of the Poisson equation is estimated from the particle density. Here, the Monte Carlo integral over the velocity space ($\mathrm{d}v$) can be taken by a histogram or by orthogonal series density estimation (OSDE) using finite elements based on cubic B-Splines. The latter are then perfectly suited to solve the Poisson equation. The histogram immediately raises the question for the suitable number of particles per cell to avoid over or under-smoothing which can be answered by elementary statistics.

grid at each time step like the SL method, this color can be also used to reduce the noise of a particle simulation.

1.2. Outline and contribution

The first part, chapter 2, of this thesis covers the stochastic aspects of particle simulations like PIC with the ultimate goal of establishing ties to the stochastic world. Since PIC is already a long established Monte Carlo method the very basics can be found in [8, 5, 9, 10, 11, 12]. In the setting of a stochastic process general aspects include the measure of error (variance), error propagation and entropy. After reviewing basic sampling steps we present new variance reduction techniques. The control variate scheme first applied to PIC by [13] is extensively discussed and new truly adaptive control variates are presented for non-equilibrium cases. So far conditional Monte Carlo has only made it to the PIC community by stratified sampling but it is actually the other pillar of variance reduction next to the control variates and allows even the heuristic coarse graining techniques a place in the stochastic world. Lagrangian methods are set in the most convenient coordinate system for variational integrators giving way to true long term stability, which is of the highest importance in multi-scale plasma physics, see [14]. Thus we review the problems arising from the combination of variance reduction techniques with geometric integration. Then the gyroaverage operator from gyrokinetic theory motivates us to mix Monte Carlo with deterministic quadrature rules resulting in new formulas for variance reduction.
Linear analysis of the Vlasov–Poisson system provides damping and growth-rates on a semi-analytical level, which are an essential verification step in numerical simulations. But linearization does not necessarily result in variance reduction. Nevertheless matrix pencil allow us to estimate entire dispersion relations from short time PIC simulations.
Collisions are introduced by the Fokker-Planck and transferred into the particle environment as the Ornstein Uhlenbeck process in [15]. The basic mechanism of applying a control variate in this setting is critically reviewed especially by numerical simulations where particles filters like sequential importance re-sampling (SIR) are applied. Recently multi-level Monte Carlo

is becoming quite popular [16], but instead of applying multi-level techniques on a time-step or grid basis like Ricketson [17, 18], the door to the combination of multiple levels is opened by the usage of different models such as fluid and kinetic descriptions. Although the concepts are introduced in two-dimensional models, everything is defined such that the presented algorithms can be implemented in almost every large Particle-In-Cell simulation.
The stochastic analysis has shown that increasing the degrees of freedom increases the background variance, such that we search for a rapidly converging representation of the fields in order to approximate also the true eigenvalues of the system very well. Furthermore anisotropies transported along the magnetic field lines can often be resolved with few Fourier modes. Thus, spectral methods suggest themselves, which are introduced in chapter 3 beginning with the review of Particle-in-Fourier (PIF) first formulated in a variational framework in [10]. Contrary to PIC where each particle contributes locally to its cell, in PIF every particle contributes to every Fourier mode. On the other hand for a global basis particles do not need to be sorted simplifying code optimization. A micro-benchmark for a one dimensional Vlasov–Poisson PIF skeleton code with various techniques implemented in MATLAB, Fortran, julia, python and OpenCL for GPU and CPU is presented.
Fourier modes form eigenfunctions of the Laplace operator such that the higher local costs of PIF can be seen as installing a pre-conditioner into the charge assignment. On a disc, such eigenfunctions are the Fourier-Bessel functions yielding the first extension of spectral particle methods onto non-periodic domains. But for arbitrary geometries eigenfunctions are not available resulting in dense matrices, which remain small with the number of physically relevant Fourier modes. The geometry of toroidally shaped devices is described with periodic toroidal and poloidal coordinates and a bounded radial direction. In a PIC-PIF hybrid the two periodic directions are discretized with PIF which is coupled to finite elements based on arbitrary degree B-Splines in the radial direction. Since the magnetic field follows these coordinates only the field aligned Harmonics are believed to be physically relevant. Established PIC codes of the ORB5 family filter the field aligned modes from the three dimensional finite element basis in order to reduce the presumably unphysical background noise. With the PIF-PIC hybrid this filtering step is directly incorporated into the choice of Fourier modes. This allows studying a (rather academic) drift kinetic ion temperature gradient instability (ITG) in different domains using free coordinate transformations on a normal laptop. Although the local B-Splines provide some sparsity, global but orthogonal polynomials (e.g. Chebyshev and Legendre) take the place of Fourier modes in non-periodic domains, such that the concept of spectral particle methods is fully generalized.
The setting in Fourier space adds some mathematical simplicity to PIF e.g. the absence of particle self force such that it conserves both energy and momentum. Since PIF discretizations are so straightforward a geometric PIF for the Vlasov-Maxwell system is derived based on the Hamiltonian splitting of [19]. Restricting a three dimensional domain onto to a line perpendicular to the magnetic field results in a three dimensional (1d2v) model suited for numerical studies of multi-species physics and implicit symplectic resolution of the $v \times B$ drift.
Complementary to the Lagrangian PIF, a pseudo-spectral discretization is the most natural choice. This alleviates comparisons and raises the confidence in obtained simulation results. For Vlasov–Poisson the Fourier-Fourier method of Joyce and Knorr is used [20] resulting in a geometric scheme, see [21]. Modifying the Hamiltonian splitting used for Vlasov–Maxwell system in PIF yields, contrary to [22], a spectral solver. This Eulerian approach to the solution of the Vlasov-Maxwell system is provided in chapter 4.
In the two fusion devices at the Max Planck Institute for Plasma physics, the Tokamak ASDEX Upgrade and the Stellarator Wendelstein 7-X, the hot plasma is confined by strong ex-

ternal magnetic fields. In this context the simple periodic electrostatic Vlasov–Poisson model is limited in its application to real physics, such that more involved systems along with a basic physics overview are introduced in the the appendix B. Adding self-consistent magnetic effects, modeling and the necessary normalizations with a multi-species Vlasov-Maxwell system are prepared for readers not familiar with plasma physics. Strong magnetization locks the particles onto the field lines yielding a gyrating motion at such a high frequency that the use of fully kinetic Vlasov models become unaffordable. Instead of solving asymptotic models like [23, 24] the most popular option is to make a high frequency approximation of the Vlasov–Maxwell system based on a coordinate transformation, which leads to a new model called gyrokinetic [25, 26]. It is merely a five dimensional system, but the biggest advantage is to remove the fast oscillations in the density, which allows for much larger time steps. Nevertheless, there is a whole zoo of gyrokinetic models, and their derivation is based upon many assumptions that are not always met in practice. Also the equations exhibit a much higher level of complexity and may be even harder to solve. Inheriting the important structure, the slightly reduced electrostatic four dimensional drift kinetic and two dimensional guiding center models are used.
Physics following the twisted field lines of magnetic equilibrium requires complex geometries, but coordinate systems are often hard-coded. Therefore, suitable coordinate systems are reviewed in an elementary description providing also test cases for flexible solvers. Then the usage of curvilinear coordinates for particle methods requires only the knowledge of an appropriate sampling technique and the corresponding transformations of the introduced systems.

Chapter 2.

Stochastic aspects of Particle-In-Cell

When performing a stochastic simulation of the Vlasov–Poisson systems or some of its relatives, we mix deterministic particle methods with a stochastic interpretation to create stochastic processes. Their convergence by the strong law of large numbers is only given for a very large particle limit, at high costs. We combine methods from the deterministic world with stochastic ensembles to reduce these costs. The two key ingredients for a stochastic particle simulation in plasma physics are the introduction of a control variate [13] and its combination with a collision operator to a time dependent stochastic process [15]. We give a full introduction on how to write a state of the art particle simulation for the Vlasov–Poisson problem. The main focus lies on the improvement of the particle mesh coupling. The particles describe random samples where Monte Carlo integration is used as the entry point [27] into the stochastic world. Thus further improvements can be made by using variance reduction techniques and low discrepancy sequences for sample generation [28]. We start with the definition of our problem in a stochastic setting and diagnose the sources of error, before attempting improvement.

2.1. Vlasov–Poisson

A common model for an electron plasma with a constant ion background is the Vlasov equation with a divergence free external magnetic field B, $\operatorname{div}(B) = 0$

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f - (E + v \times B) \cdot \nabla_v f = 0. \tag{2.1}$$

It can be coupled with the Poisson equation for the electric potential Φ. With the electron charge density $\rho = -\int f \, \mathrm{d}v$, the Poisson equation is defined as:

$$-\Delta\Phi = \rho_{\text{ion}} + \rho, \qquad E := -\nabla\Phi. \tag{2.2}$$

In most cases the ion charge density forms a uniform background which means $\rho_{\text{ion}} = 1$. Let the phase space be defined as $\Omega = \Omega_x \times \Omega_v = [0, L] \times \mathbb{R}$. The total energy of the system is the sum of the kinetic energy and the electrostatic energy:

$$\mathcal{H}(t) = \mathcal{H}_T(t) + \mathcal{H}_E(t) = \frac{1}{2} \iint f(x, v, t) v^2 \, \mathrm{d}x\mathrm{d}v + \frac{1}{2} \int_{\Omega_x} \|\nabla\Phi(x, t)\|^2 \, \mathrm{d}x. \tag{2.3}$$

2.1.1. Method of characteristics

Equation (2.1) describes a conservation law, which is solved by the methods of characteristics. We define the characteristics $(V(t), X(t))$, as functions of time such that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} f\left(X(t), V(t), t\right) = \frac{\mathrm{d}X(t)}{\mathrm{d}t} \partial_x f\left(X(t), V(t), t\right) \\ + \frac{\mathrm{d}V(t)}{\mathrm{d}t} \partial_v f\left(X(t), V(t), t\right) + \partial_t f\left(X(t), V(t), t\right) = 0.\end{aligned} \tag{2.4}$$

Inserting $\partial_t f$ from (2.1) into (2.4) yields the equations of motions for the characteristics of eqn. (2.1), which read

$$\frac{\mathrm{d}}{\mathrm{d}t}V(t) = -\left(E(t,X(t)) + V(t)\times B(t,X(t))\right) \text{ and } \frac{\mathrm{d}}{\mathrm{d}t}X(t) = V(t). \tag{2.5}$$

Then f as solution of eqn. (2.1) is constant along the characteristics (2.5), which means for given initial position in phase space (X_0, V_0) we have

$$f(X(t=0), V(t=0), t=0) = f(X(t), V(t), t) \quad \forall t \geq 0. \tag{2.6}$$

In this way eqn. (2.1) can be solved with the method of characteristics. Given the fields B and E we can follow the characteristics by solving eqn. (2.5) with a standard ODE integrator. We can introduce a second density $g(x, v, t)$ which solves

$$\frac{\partial g}{\partial t} + v\cdot\nabla_x g - (E + v\times B)\cdot\nabla_v g = 0 \tag{2.7}$$

and call it the sampling density, prior or the law of (X, V). By imposing $\int_\Omega g(x, v, t = 0)\,\mathrm{d}x\mathrm{d}v = 1$, $g(x, v, t = 0) \geq 0$, $\forall(x, v)$ the initial sampling distribution $g(\cdot,\cdot, t = 0)$ becomes a probability density. Since g follows the same Vlasov equation (2.7) as f, see eqn. (2.1), it is constant along the same characteristics (2.6).
The Vlasov equation (2.7) conserves positivity and volume, therefore, g stays a probability density for all $t \geq 0$. In order to verify that $g(x, v, t)$ is the probability density of the characteristics $(X(t), V(t))$ we rewrite the characteristics as a mapping φ_t. Since f is constant along the characteristics, we can implicitly define a diffeomorphism $\varphi_t : (x_0, v_0) \mapsto (x, v)$ for every $t \geq 0$ such that

$$f(x, v, t) = f(\varphi_t(x_0, v_0), t) = f(x_0, v_0, 0). \tag{2.8}$$

The same property then also holds for g, namely $g(\varphi_t(x_0, v_0), t) = g(x_0, v_0, 0)$. We seek a change in variables $(x, v) := \varphi_t(x_0, v_0)$ and denote the Jacobi determinant of φ_t as J_{φ_t}. For any phase-space volume V equation (2.9) then holds under the change of variables; also for f.

$$\begin{aligned}\iint_{\varphi(V)} g(x, v, t)\,\mathrm{d}x\mathrm{d}v &= \iint_V g\left(\varphi_t(x_0, v_0), t\right) J_{\varphi_t}(x_0, v_0)\,\mathrm{d}x_0\mathrm{d}v_0 \\ &= \iint_V g\left(x_0, v_0, 0\right) J_{\varphi_t}(x_0, v_0)\,\mathrm{d}x_0\mathrm{d}v_0\end{aligned} \tag{2.9}$$

This means that at time t, $g\left(x_0, v_0, t = 0\right) J_{\varphi_t}(x_0, v_0)$ is the probability density for the random deviate $(X(t), V(t)) = \varphi(X_0, V_0)$ and the Jacobian has to be taken into account. For the Vlasov equation the Jacobi determinant is one $J_{\varphi_t}(x, v) = 1$. Therefore the characteristics transport the actual value of the probability density at every time t. This also holds true for a symmetric integrator, e.g. one time step of the symplectic Euler scheme given in equation (2.10).

$$\begin{aligned}\varphi_t(x, v) &= \left(x + tv,\ v + t\frac{q}{m}E(x + tv, t)\right), \\ \nabla\varphi_t(x, v) &= \begin{pmatrix} 1 & t \\ t\frac{q}{m}\partial_x E(x + tv, t) & 1 + t^2\frac{q}{m}\partial_x E(x + tv, t)\end{pmatrix}\end{aligned} \tag{2.10}$$

We then see that the semi-discrete flow also has the right Jacobi determinant:

$$\det(\nabla\varphi_t) = 1 + t^2\frac{q}{m}\partial_x E(x + tv, t) - t^2\frac{q}{m}\partial_x E(x + tv, t) = 1. \tag{2.11}$$

Yet when we consider the standard explicit Euler scheme and its Jacobi determinant given in eqn. (2.12) the determinant of the flow is not one.

$$\begin{aligned} \varphi_t(x,v) &= \left(x + tv,\ v + t\frac{q}{m}E(x,t)\right), \quad \nabla\varphi_t = \begin{pmatrix} 1 & t \\ t\frac{q}{m}\partial_x E(x,t) & 1 \end{pmatrix} \\ \det(\nabla\varphi_t) &= 1 - t^2\frac{q}{m}\partial_x E(x,t) \neq 1 \end{aligned} \tag{2.12}$$

Therefore the likelihood g has to be rescaled accordingly such that it continuously represent the distribution of the random deviate $(X(t), V(t))$. Technically f should still stay constant because of the method of characteristics, which leads ultimately to an inconsistency. Hence in this thesis volume preserving integrators are used whenever possible. The Vlasov–Poisson system is a Hamiltonian system in which our phase space coordinates (x, v) coincide with the Hamiltonian coordinates (q, p). Without magnetic field the system can be written as

$$(\dot{p}, \dot{q}) = J^{-1}\nabla_{(p,q)}H(p,q), \quad J = \begin{pmatrix} & -I \\ I & \end{pmatrix}. \tag{2.13}$$

For different systems we will obtain a different matrix J and the coordinates (p, q) cannot be identified as (x, v) much longer. An integrator is called symplectic if the mapping induced by φ_t is symplectic with respect to J, which is checked by

$$\nabla\varphi_t(p,q)^t J \nabla\varphi_t(p,q) = J. \tag{2.14}$$

See Hairer's lecture notes for a short introduction to Hamiltonian systems [29]. Such symplectic integrators always conserve phase space volume and can also conserve quantities like energy but not every phase space volume preserving integrator is symplectic, see also [30]. But conservation of phase space is such an important property that schemes like the Boris method perform so well although they cannot be symplectic for any system [31]. Many of these integrators along with detailed theory for plasma physics can already be found here [14]. Here the third and fourth order symplectic Runge Kutta schemes from [32] are denoted by $rk3s$ and $rk4s$ respectively. The standard second order scheme $rk2s$ corresponds to the well known leap frog, and the first order $rk1s$ is the symplectic Euler. We saw that the likelihoods f and g are propagated using the Jacobi determinant of the flow. This means nothing has to be done here, since the Jacobian is always one when a suitable integrator is used. But we cannot propagate other likelihoods, which are not constant such as the marginal densities, *sampled* charge density g_x and the *sampled* velocity density g_v.

$$g_x(x,t) = \int g(x,v,t)\ \mathrm{d}v, \quad g_v(v,t) = \int g(x,v,t)\ \mathrm{d}x \tag{2.15}$$

Also for integrators which do not preserve phase space volume but are dissipative such as asymptotically preserving schemes like [24, 23] the likelihoods have to be propagated accordingly.

2.1.2. Stochastic process

We will now slightly deviate in notation from the standard Particle-In-Cell (PIC) method [5]. The introduction of the probability density function g allows us to define a corresponding random variable $Z(t) = (X(t), V(t))$. As a time dependent variable $Z(t)$ is a stochastic process [33] describing the solution to eqn. (2.7). We also know that $Z(t)$ is constant along the characteristic (2.6). But we are interested in the solution of (2.7), subject to (2.2). And not necessarily the complete distribution f but at least certain moments of f such as kinetic

energy, momentum and electrostatic field energy emerging from (2.2). We suppose that the initial conditions $g(x, v, t = 0)$ and $f(x, v, t = 0)$ are known for all $(x, v) \in \Omega$, then we immediately have a solution by following the characteristics

$$\begin{aligned} f(X(t=0), V(t=0), t=0) &= f(X(t), V(t), t) \\ g(X(t=0), V(t=0), t=0) &= g(X(t), V(t), t) \quad \forall t \geq 0. \end{aligned} \tag{2.16}$$

Interesting moments of the plasma distribution are integrals of the form

$$\theta(t) := \int_\Omega \Theta(x, v) f(x, v, t) \, \mathrm{d}x\mathrm{d}v. \tag{2.17}$$

Equation (2.17) can be rewritten as the expected value of a stochastic process since g is the corresponding probability density for Z.

$$\begin{aligned} \theta(t) &:= \int_\Omega \Theta(x, v) f(x, v, t) \, \mathrm{d}x\mathrm{d}v \\ &= \int_\Omega \Theta(x, v) \frac{f(x, v, t)}{g(x, v, t)} \, g(x, v, t) \, \mathrm{d}x\mathrm{d}v \\ &= \mathbb{E}\left[\Theta(X(t), V(t)) \frac{f(X(t), V(t), t)}{g(X(t), V(t), t)}\right] \end{aligned} \tag{2.18}$$

Using eqn. (2.16) the *weight process* $W(t)$ is defined as

$$W(t) := \frac{f(X(t), V(t), t)}{g(X(t), V(t), t)} = \frac{f(X(0), V(0), 0)}{g(X(0), V(0), 0)} = W(0) \ \forall t \geq 0, \tag{2.19}$$

giving the relation between the sampling distribution g and the function f. This yields an simplification of eqn. (2.18):

$$\theta(t) := \mathbb{E}\left[\Theta(X(t), V(t)) W(t)\right]. \tag{2.20}$$

To make use of the standard Monte Carlo estimator we define N_p independently and identically distributed (i.i.d.) samples - called markers or particles - of the initial random deviate $Z(0)$ using the knowledge of the probability density $g(x, v, t = 0)$.

$$Z_k(0) = (X_k(0), V_k(0)) \ \text{ i.i.d.} \sim Z(0) \tag{2.21}$$

Every sample follows the same stochastic process, giving us the ability to calculate $Z(t)$ from $Z(0)$ for all $t \geq 0$ by following the characteristics (2.16). This corresponds to advancing the markers in time, by use of a standard integrator for ordinary differential equations [34, 35]. Here we tend to use more sophisticated methods [36, 37], which are now also a growing field in plasma physics [14]. Since the stochastic process $Z(t)$ is defined by the characteristics and every realization of $z(t) \in \mathbb{R}$ is a characteristic we call $Z(t)$ a characteristic as well. The i.i.d. duplicates $(Z_k)_{k=1,\ldots,N_p}$ of Z allow us to define a new random deviate $\hat{\theta}$, which we call the standard Monte Carlo estimator, see eqn. (2.22).

$$\theta(t) = \mathbb{E}\left[\Theta(X(t), V(t)) W(t)\right] = \mathbb{E}\Big[\underbrace{\frac{1}{N_p} \sum_{k=1}^{N_p} \Theta(X_k(t), V_k(t)) W_k(t)}_{:=\hat{\theta}} \Big] \tag{2.22}$$

The standard Monte Carlo estimators expectation coincides with $\theta = \mathbb{E}\left[\hat{\theta}\right]$ but its variance decreases with the number of particles N_p.

$$\mathbb{V}\left[\hat{\theta}\right] = \frac{\mathbb{V}\left[\Theta(X(t), V(t)) W(t)\right]}{N_p} \tag{2.23}$$

If actual $(z_k)_{k=1,\dots,N_p} \in \mathbb{R}$ samples as realizations of the random deviates $(Z_k)_{k=1,\dots,N_p}$ are drawn the *estimator* in eqn. (2.22) is turned into an *estimate* for θ.

$$\theta(t) = \mathbb{E}\left[\Theta(X(t), V(t))W(t)\right] \approx \frac{1}{N_p}\sum_{k=1}^{N_p} \Theta(x_k(t), v_k(t))w_k(t) \tag{2.24}$$

In the following expectations are constantly approximated by the standard Monte Carlo estimator such that we neglect the difference between estimator and estimate because of its unnecessary notational overhead. The capital notation $Z_k = (X_k, V_k)$ is used when the focus lies on the stochastic aspect, while the lower case notation $z_k = (x_k, v_k)$ is used when we focus on the actual numerics.

Kinetic energy

We are interested in moments of the Vlasov equation such as the kinetic energy. An estimate for the kinetic energy $\mathcal{H}_T(t)$ can be computed by setting $\Theta(x, v) = \frac{1}{2}v^2$ such that the standard Monte Carlo estimator $\hat{\mathcal{H}}_T(t)$ reads

$$\begin{aligned} \mathcal{H}_T(t) &= \frac{1}{2}\int_\Omega f(x, v, t)\ v^2\ \mathrm{d}x\mathrm{d}v \\ &= \frac{1}{2}\mathbb{E}[V(t)^2 W(t)] \\ &\approx \hat{\mathcal{H}}_T(t) := \frac{1}{N_p}\sum_{k=1}^{N_p} V_k(t)^2 W_k(t) \end{aligned} \tag{2.25}$$

The deterministic approach, yielding the same estimator starts with the discretization of the density g with Dirac masses as a Klimontovich density

$$g(x, v, t) \approx g_p(x, v, t) = \frac{1}{N_p}\sum_{k=1}^{N_p} \delta\left(x - X_k(t)\right)\delta\left(v - V_k(t)\right), \tag{2.26}$$

yielding also an approximation to f

$$f(x, v, t) \approx f_p(t, x, v) = \frac{1}{N_p}\sum_{k=1}^{N_p} \delta\left(x - X_k(t)\right)\delta\left(v - V_k(t)\right)W_k(t). \tag{2.27}$$

Inserting the discretization f_p yields the same standard Monte Carlo estimator $\hat{\theta}$ for θ:

$$\begin{aligned} \theta(t) &:= \int_\Omega \Theta(x, v) f(x, v, t)\ \mathrm{d}x\mathrm{d}v \\ &\approx \int_\Omega \Theta(x, v) f_p(x, v, t)\ \mathrm{d}x\mathrm{d}v \\ &= \int_\Omega \Theta(x, v)\frac{1}{N_p}\sum_{k=1}^{N_p} \delta\left(x - X_k(t)\right)\delta\left(v - V_k(t)\right)W_k(t)\ \mathrm{d}x\mathrm{d}v \\ &= \frac{1}{N_p}\sum_{k=1}^{N_p} \Theta(X_k(t), V_k(t))W_k(t) = \hat{\theta}(t). \end{aligned} \tag{2.28}$$

Depending on the variance reduction and particle filtering techniques, the weight process $W_k(t)$ is not anymore only defined by the ratio between f and g. Therefore, as a starting point we explicitly introduce the plasma probability f_k^t and the particle probability g_k^t:

$$f_k^t := f(X_k(t), V_k(t), t), \quad g_k^t := g(X_k(t), V_k(t), t), \tag{2.29}$$

such that the standard Monte Carlo estimator reads

$$\hat{\theta}(t) = \frac{1}{N_p} \sum_{k=1}^{N_p} \Theta(X_k(t), V_k(t)) \frac{f_k^t}{g_k^t}. \tag{2.30}$$

For a marker z_k^t the quantity g_k^t denotes the likelihood of finding a marker at the given phase space position due to the sampling density g. Since f is a probability density up to normalization f_k^t, is also a likelihood but for the plasma density. It can be interpreted as the likelihood of finding a plasma particle at z_k.

Mass and L^p-norm

The mass $\iint f \, \mathrm{d}x\mathrm{d}v$ and the L^p-norm of f for $p \in \mathbb{N}$ are two conserved quantities of the Vlasov–Poisson system, one encounters first and that are most easy to prove. They are also intrinsically conserved by the standard particle method and therefore we give their definition, also to be used with Fokker–Planck collisions. The mass reads

$$\iint_\Omega f(x,v,t) \, \mathrm{d}x\mathrm{d}v = \mathbb{E}\left[\frac{f(Z(t),t)}{g(Z(t),t)}\right] \approx \frac{1}{N_p} \sum_{k=1}^{N_p} \frac{f_k^t}{g_k^t} \tag{2.31}$$

and the L^p-norm is defined as

$$\iint_\Omega f(x,v,t)^p \, \mathrm{d}x\mathrm{d}v = \mathbb{E}\left[\frac{|f(Z(t),t)|^p}{g(Z(t),t)}\right] \approx \frac{1}{N_p} \sum_{k=1}^{N_p} \frac{\left|f_k^t\right|^p}{g_k^t}. \tag{2.32}$$

Entropy

The Vlasov–Poisson system conserves entropy, also known as the differential entropy,

$$\int_\Omega f \ln(f) \, \mathrm{d}x\mathrm{d}v = \mathbb{E}\left[\frac{f(Z(t),t)\ln\left(f(Z(t),t)\right)}{g(Z(t),t)}\right] \approx \frac{1}{N_p} \sum_{k=1}^{N_p} \frac{f_k^t \ln(f_k^t)}{g_k^t}. \tag{2.33}$$

The particle method will also conserve the discrete entropy, yet over time as the particles represent less and less the true solution f. Thus the discrete entropy will differ from the true solution. There are different ways to estimate the entropy from a sample, [38] gives an overview, while in [39] mesh based examples ready for implementation can be found. Here, in the spirit of grid less methods for particles, the author's choice is the nearest neighbor estimator for the Shannon entropy [40]. The nearest neighbor kernel density estimator [41][p.305] of the sampling distribution g is given in eqn. (2.34). It gives us a value of the sampling density g at every sample point

$$\hat{g}(x_k, v_k) = \frac{1}{N_p} \frac{\Gamma\left(\frac{d}{2}+1\right)}{\pi^{\frac{d}{2}} (R_k)^d}. \tag{2.34}$$

Rescaling the sampling density estimator in eqn. (2.34) by the weight, allows for estimation of the plasma density f at every sample point by using eqn. (2.35)

$$\hat{f}(x_k, v_k) = \frac{1}{N_p} \frac{\Gamma\left(\frac{d}{2}+1\right)}{\pi^{\frac{d}{2}} (R_k)^d} \frac{f_k}{g_k}. \tag{2.35}$$

Corrections in the asymptotic limit of $N_p \to \infty$ result in the Shannon entropy estimator from [41][p.304] for the sampling density, which reads

$$\hat{H}(g) = \left[\frac{d}{N_p} \sum_{k=1}^{N_p} \ln\left(R_k\right)\right] + \ln\left(\frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\right) + \gamma + \ln(N_p). \tag{2.36}$$

Here $d = 2$ denotes the dimension, $\gamma \approx 0.5772156649$ the Euler-Mascheroni constant and R_k the euclidean distance to the nearest neighbor of the kth particle:

$$R_k := \min_{l \neq k, l=1,\dots,N_p} \left\| \begin{pmatrix} x_k \\ v_k \end{pmatrix} - \begin{pmatrix} x_l \\ v_l \end{pmatrix} \right\|_2 . \tag{2.37}$$

We want to obtain the Shannon entropy for the plasma density f given an arbitrary sampling density g. Let $w = \frac{f}{g}$ denote the weight function, we obtain

$$-\iint \ln(f) f \, \mathrm{d}x\mathrm{d}v = -\iint \ln(gw) gw \, \mathrm{d}x\mathrm{d}v = -\iint w \, \ln(g) g \, \mathrm{d}x\mathrm{d}v - \iint \ln(w) w \, g \, \mathrm{d}x\mathrm{d}v. \tag{2.38}$$

By using the entropy estimator (2.36) for the $\ln(g)g$ term in eqn. (2.38) we obtain a nearest neighbor entropy estimator for the plasma density f in eqn. (2.39).

$$\hat{H}(f) = \frac{1}{N_p} \sum_{k=1}^{N_p} \left[d \ln(R_k) \; + \ln\left(\frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} \right) + \gamma + \ln(N_p) \right] \frac{f_k}{g_k} - \frac{1}{N_p} \sum_{k=1}^{N_p} \log\left(\frac{f_k}{g_k} \right) \frac{f_k}{g_k}. \tag{2.39}$$

With above entropy estimates, the discretization error is already included.
In gyrokinetic theory, a different estimate for the entropy has been established. Since some physicists [42], [43], [44] relate the entropy to the δf method we give an analytical treatment. At first one is interested in the difference in Shannon's entropy

$$\mathcal{F}(f) := \iint_\Omega f \ln(f) - f_0 \ln(f_0) \, \mathrm{d}x\mathrm{d}v, \tag{2.40}$$

since it accounts for the change in entropy relative to an initial condition or an equilibrium. We use a quadratic Taylor expansion in f around f_0, the difference $\delta f := f - f_0$ and keep only the leading order in δf:

$$\begin{aligned} f \ln(f) - f_0 \ln(f_0) &\simeq \frac{(f-f_0)^2}{2f_0} + (\ln(f_0)+1)(f-f_0) \\ &= \frac{(\delta f)^2}{2f_0} + \delta f \, (\ln(f_0)+1) \;\; \sim \;\; \frac{(\delta f)^2}{2f_0}. \end{aligned} \tag{2.41}$$

Inserting the particle discretization this yields:

$$\mathcal{F}(f) = \int_\Omega \frac{(\delta f)^2}{2f_0} \, \mathrm{d}x\mathrm{d}v \approx \frac{1}{2}\frac{1}{N_p} \sum_{k=1}^{N_p} \frac{\left(f(t, x_k(t), v_k(t)) - f_0(x_k(t), v_k(t))\right)^2}{g(t, x_k(t), v_k(t)) f_0(x_k(t), v_k(t))}. \tag{2.42}$$

In the literature [43], [44] the δ-weights are defined as

$$\delta w_k := \frac{f(t, x_k(t), v_k(t)) - f_0(x_k(t), v_k(t))}{g(t, x_k(t), v_k(t))} \tag{2.43}$$

including the normalization by the sampling density g, yielding an additional layer of misunderstanding when estimating with

$$\frac{1}{2}\frac{1}{N_p} \sum_{k=1}^{N_p} \frac{\delta w_k^2}{f_0(x_k(t), v_k(t))}. \tag{2.44}$$

This is often mixed with the methods of control variates and linearization, but we will not investigate this further.

We turn to a better estimate for an entropy difference, namely the Kullback-Leibler divergence, or relative entropy [45]. It can be defined with respect to an equilibrium $f_{eq}(x,v)$, similar to (2.40) as

$$\int_\Omega f \ln\left(\frac{f}{f_{eq}}\right)\,\mathrm{d}x\mathrm{d}v = \mathbb{E}\left[\frac{f(Z(t),t)}{g(Z(t),t)}\ln\left(\frac{f(Z(t),t)}{f_{eq}(Z(t))}\right)\right] \approx \frac{1}{N_p}\sum_{k=1}^{N_p}\frac{f_k^t}{g_k^t}\ln\left(\frac{f_k^t}{f_{eq}(x_k^t,v_k^t)}\right). \tag{2.45}$$

Here a quadratic Taylor expansion in f around f_0 yields

$$f\ln\left(\frac{f}{f_0}\right) \simeq \frac{(f^2-f_0^2)}{(2f_0)}, \tag{2.46}$$

which lacks the $f_0 f$ term.
Entropy estimates are widely used as a measure of error [46] in particle simulations and are often related to the particle noise. This makes it hard to distinguish between discretization errors and new physics and therefore, we employ stochastic error estimates.

2.1.3. Measure of error (MSE)

We can quantify the error of the estimator $\hat{\theta}$ by the definition of the mean squared error (MSE) which is the expectation of the ℓ^2 error

$$\begin{aligned}\mathrm{MSE}\left[\hat{\theta}\right] =& \mathbb{E}\left[\left(\hat{\theta}(t)-\theta(t)\right)^2\right] = \mathbb{V}\left[\hat{\theta}(t)\right] + \left(\mathbb{E}\left[\hat{\theta}(t)\right]-\theta(t)\right)^2 \\ =& \frac{\mathbb{V}\left[\Theta(X(t),V(t))\right]}{N_p} + \left(\mathbb{E}\left[\hat{\theta}(t)\right]-\theta(t)\right)^2.\end{aligned} \tag{2.47}$$

The MSE consists of the variance and the square bias which varies for different estimators. In the case of the kinetic energy (2.25), the bias vanishes as $\mathbb{E}\left[V(t)^2W(t)\right] = \mathcal{H}_T(t)$.

$$\mathrm{MSE}\left[\hat{\mathcal{H}}_T(t)\right] = \frac{\mathbb{V}\left[V(t)^2W(t)\right]}{N_p} + \underbrace{\left(\mathbb{E}\left[V(t)^2W(t)\right]-\mathcal{H}_T(t)\right)^2}_{=0} \tag{2.48}$$

This means $\hat{\mathcal{H}}_T(t)$ is an unbiased estimator for the kinetic energy and will converge by the strong law of large numbers for $N_p \to \infty$ *almost surely* to $\mathcal{H}_T(t)$.

So far we can estimate any moment Θ of the phase space density f by following the characteristics with the randomly drawn markers $Z_k^0,\ k=1,\dots,N_p$ forming samples Z_k^t of the stochastic process $Z(t)$ by time integration. But this requires the knowledge of the electric field $E(x,t)$ stemming from the Poisson equation, which we have to solve given the samples of $Z(t)$.

2.1.4. Finite elements for the Poisson equation

For PIC codes, it is very common to use finite elements to solve for the fields [15, 47, 19, 48]. Therefore, we provide a brief example for the Poisson equation. The same Ansatz and test functions $\psi_n \in H_1([0,L],\mathbb{R}) = V,\quad n=1,\dots,N_h$ are chosen for the variational formulation. In many cases these are B-splines because of their partition of unity and good approximation properties, see [49, 50]. Nevertheless any other set of function like Fourier modes or orthogonal polynomials suitable for a Galerkin discretization can be chosen. Define mass $M_{n_1,n_2} := \langle\psi_{n_1},\psi_{n_2}\rangle_{L^2([0,L])}$ and stiffness matrix $K_{n_1,n_2} := \langle\nabla\psi_{n_1},\nabla\psi_{n_2}\rangle_{L^2([0,L])}$ for

$n_1, n_2 = 1, \ldots, N_h$. The corresponding finite element or discretized Galerkin subspace is denoted as $V_h = \operatorname{span}\left(\{\psi_1, \ldots, \psi_{N_p}\}\right) \subset V$. The weak form of the Poisson equation (2.2) without ion contribution using only the electron charge density $\rho(x,t) = \int f(x,v,t)\ \mathrm{d}v$ reads

$$\langle \nabla\Phi(t), \nabla\psi\rangle_{L^2([0,L])} = \langle \rho(t), \psi\rangle_{L^2([0,L])} \quad \forall \psi \in V. \tag{2.49}$$

But before we solve the weak Poisson equation (2.49), let us start by the L^2 projection of $\rho(x,t)$ into the discrete space V_h. In general ρ is not contained in V_h thus one searches for the L^2 projection $\rho_h \in V_h$ of ρ onto the space V_h given by

$$\langle \rho_h(t), \psi\rangle_{L^2([0,L])} = \langle \rho(t), \psi\rangle_{L^2([0,L])} \quad \forall \psi \in V_h. \tag{2.50}$$

In the particle environment ρ is not available but a particle discretization, given by the stochastic process $X(t)$ or the Klimontovich density

$$\rho_p(x,t) = \frac{1}{N_p} \sum_{k=1}^{N_p} \delta\left(x - X_k(t)\right) W_k(t). \tag{2.51}$$

In the following, let $\psi := (\psi_1, \ldots, \psi_{N_h})^t$ be the vector valued function containing all Ansatz functions, then the discretized right hand side $b(t)$ with its estimator $\hat{b}(t)$ is defined as

$$\begin{aligned} b(t) &:= \langle \rho(x,t), \psi\rangle_{L^2([0,L])} \\ &= \int \psi(x)\rho(x,t)\ \mathrm{d}x = \mathbb{E}\Big[\underbrace{\psi\left(X(t)\right) W(t)}_{B(t):=}\Big] = \mathbb{E}\left[B(t)\right] \\ &\approx \int \psi(x)\rho_p(x,t)\ \mathrm{d}x = \frac{1}{N_p}\sum_{k=1}^{N_p} W_k(t)\psi(X_k(t)) =: \hat{b}(t). \end{aligned} \tag{2.52}$$

Using the mass matrix M we denote the L^2 projection as a linear operator $\mathcal{M} : b \mapsto M^{-1}b$, which allows us to use the linearity of the expectation later. Then the L^2 projection of $\rho(x,t)$ using the finite element space yields a discrete approximation $\rho_h(x,t)$, see eqn. (2.53) depending on the right hand side vector $b(t) \in \mathbb{R}^{N_h}$.

$$\rho(x,t) \approx \rho_h(x,t) = \left(\mathcal{M}b(t)\right)^t \cdot \psi(x) \tag{2.53}$$

$$\approx \hat{\rho}_h(x,t) = \left(\mathcal{M}\hat{b}(t)\right)^t \cdot \psi(x) \tag{2.54}$$

This vector is also defined by the expectation $b(t) = \mathbb{E}\left[B(t)\right]$ and therefore, can be estimated from samples of $Z(t)$ by the estimator $\hat{b}(t)$. Introducing this estimator into the purely mesh based description of ρ yields a stochastic estimator $\hat{\rho}_h$, see eqn. (2.54). The introduction of the multivariate random deviate $b(t)$ is precisely the point, where the particles are coupled to the mesh. Hence equations (2.53) and (2.54) describe the particle mesh coupling. The first approximation is made by the basis functions in V_h and the second one by the Monte Carlo estimator. Staying in the Galerkin framework we can also solve the Poisson equation in the same manner by use of the stiffness matrix K and a linear operator $\mathcal{K}y := K^{-1}(y)$. The approximations on the field yield then

$$\begin{aligned} \Phi(x,t) &\approx \Phi_h(x,t) = \left(\mathcal{K}b(t)\right)^t \cdot \psi(x) \\ &\approx \hat{\Phi}_h(x,t) = \left(\mathcal{K}\hat{b}(t)\right)^t \cdot \psi(x). \end{aligned} \tag{2.55}$$

Reintroducing the ion contribution is notationally cumbersome, thus we define

$$\rho_{ion,h} = \int_0^L \rho_{\mathrm{ion}}(x)\psi(x)\ \mathrm{d}x = \int_0^L \psi(x)\ \mathrm{d}x \tag{2.56}$$

In the following the affine linear operator

$$\mathcal{K}x := K^{-1}(\rho_{ion,h} + x) \tag{2.57}$$

shall incorporate the complete field solve including the boundary conditions. Because these are periodic the operator $\mathcal{K}$ is supposed to deal with the singular Poisson solve also on a numerical level. Incorporating the ion background $\rho_{\text{ion}} = 1$ in all the following theory is more confusing than helpful such that, without loss of generality, we chose to ignore it in the following discussions. The electric field $E(x,t)$ at a given position $x \in [0, L]$ is approximated by its estimator $\hat{E}_h(x,t)$:

$$\begin{aligned} E(x,t) = -\nabla\Phi(x,t) &\approx E_h(x,t) = -\nabla\Phi_h(x,t) \\ &\approx \hat{E}_h(x,t) = -\nabla\hat{\Phi}(x,t) = \left(\mathcal{K}\hat{b}(t)\right)^t (-\nabla\psi(x)). \end{aligned} \tag{2.58}$$

2.1.5. Particle mesh coupling by KDE

Any basic statistics course will cover kernel density estimation (KDE), which uses a smoothing kernel in order to construct a continuous function from a marker density [51, 52, 53]. This is most certainly the reason why the earliest PIC codes used KDE for the particle mesh coupling [5]. A smoothing kernel K is a symmetric and mostly hat shaped function satisfying at least the following constraints

$$K(x) \geq 0 \ \forall x, \quad \int xK(x)\ dx = 0, \qquad \int x^2 K(x)\ dx \neq 0. \tag{2.59}$$

A mollified version ρ_h of the charge density, now subject to a discretization error is obtained by convolution with the kernel

$$\rho_h(x,t) = \int_0^L f(y,v,t) K\left(\frac{x-y}{h}\right) \frac{1}{h}\ \mathrm{d}y\mathrm{d}v. \tag{2.60}$$

Here h denotes the width of the smoothing kernel. With increasing h the small oscillations are lost, which is mostly desirable. Inserting the Klimontovich density f_p into eqn. (2.60) yields the KDE $\hat{\rho}$ for ρ.

$$\begin{aligned} \hat{\rho}(x,t) &= \int_0^L \frac{1}{N_p} \sum_{n=1}^{N_p} \delta\left(y - x_n(t)\right) \delta\left(v - v_n(t)\right) w_n K\left(\frac{x-y}{h}\right) \frac{1}{h}\ \mathrm{d}y\mathrm{d}v \\ &= \frac{1}{N_p} \sum_{n=1}^{N_p} K\left(\frac{x - x_n(t)}{h}\right) \frac{1}{h}\ w_n \end{aligned} \tag{2.61}$$

In the Klimontovich density each particle is represented by a δ function, but after the convolution with the smoothing kernel K it appears as if every particle has the physical shape $S(x) = K\left(\frac{x}{h}\right)\frac{1}{h}$. Thus it is very common in the community to denote $\hat{\rho}$ by

$$\hat{\rho}(x,t) = \frac{1}{N_p} \sum_{n=1}^{N_p} S(x - x_n(t)) w_n. \tag{2.62}$$

Now (2.62) can be evaluated on any grid consisting of arbitrary grid points $(\bar{x}_m)_{m=1,\ldots,M} \in [0, L]$, yielding

$$\bar{\rho}_m := \hat{\rho}(\bar{x}_m, t). \tag{2.63}$$

The grid does not have to be regular. It is only important that we are able to solve the Poisson equation with a collocation or Galerkin based method on this grid. This includes the discrete Fourier transformation (DFT), Chebyshev transform (see [54]) and finite differences as the standard choice. It is very important to note that although the smoothing window width h disappeared somewhere around eqn. (2.62) into the shape function S it is still a free parameter not linked in any way to the cell size of the used grid. But if B-splines, see fig. 2.1, are used as a smoothing kernel K it is numerically more efficient and also much easier to implement to chose the same h for the cell size and the smoothing window width. An optimal smoothing window width h depends of course on the number of samples [51], but since it is now also linked to the cell size the well known *particle per cell* criterion is obtained.
The Poisson equation can be solved on the grid in Fourier space using the DFT $\mathcal{F}$, inverse DFT $\mathcal{F}^{-1}$ and the discrete wave vector $\bar{k}$ yielding

$$\bar{\Phi} = \mathcal{F}^{-1} \frac{1}{(\mathrm{i}\bar{k})^2} \mathcal{F}\bar{\rho}. \tag{2.64}$$

Note that the ion background can be subtracted by manually setting the zeroth Fourier mode to zero. The historically most common choice is a second order finite difference approximation, where the potential is obtained by solving

$$\frac{\bar{\Phi}_{m-1} - 2\bar{\Phi}_m + \bar{\Phi}_{m+1}}{h^2} = \bar{\rho}_m, \tag{2.65}$$

for $\bar{\Phi}$. However the potential $\Phi(\bar{x}_m) = \bar{\Phi}$ is obtained at the grid points it can be evaluated again at the particle positions by interpolation using the original shape function S and a corresponding mass matrix $\mathcal{M}$ for the interpolation

$$\hat{\Phi}(x_n(t), t) = \sum_m \left(\mathcal{M}^{-1}\bar{\Phi}\right)_m S(\bar{x}_m - x_n(t)). \tag{2.66}$$

The electric field can then be obtained by the derivative

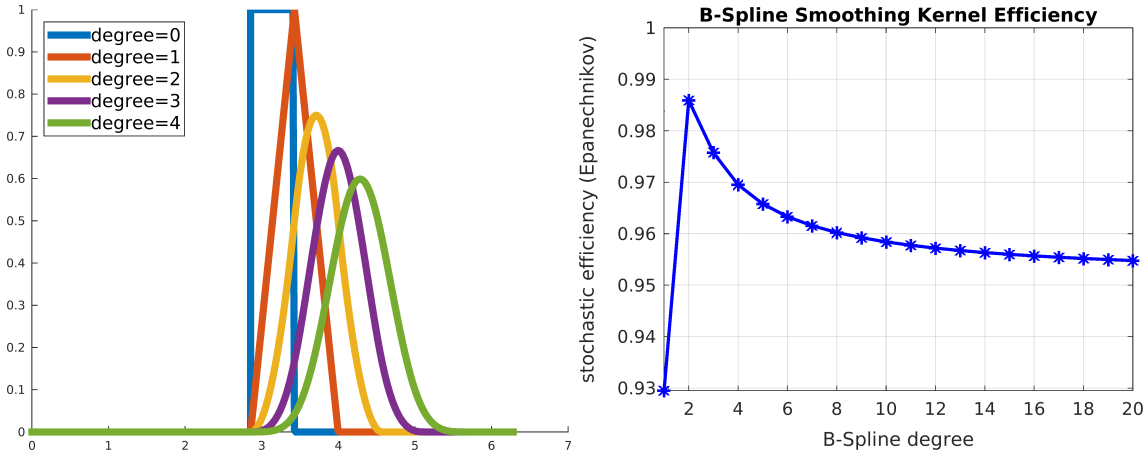
$$\hat{E}(x_n(t), t) = -\sum_m \left(\mathcal{M}^{-1}\bar{\Phi}\right)_m S'(\bar{x}_m - x_n(t)). \tag{2.67}$$

Many popular PIC codes based in some form on Birdsall's ES-PIC [5] employ an additional discretization for obtaining the electric field from the potential at the grid points by e.g.

$$\hat{E}(x_n(t), t) = -\sum_m \bar{E}_m S(\bar{x}_m - x_n(t)) \text{ with } \bar{E}_m = -\frac{\bar{\Phi}_{m+1} + \bar{\Phi}_{m-1}}{2h}. \tag{2.68}$$

An analog projection can also be made on the level of the discrete Fourier transform. Such ad-hoc discretizations do mostly not fit in a variational framework thus giving rise to unnatural long term effects, such as e.g. the finite grid instability[55]. However it is shown in [10], that a discretization with linear shape functions and linear finite elements can coincide with the second finite difference approximation. It can be summarized that finite elements are closer to orthogonal series density estimation (OSDE) and collocation methods use kernel density estimation (KDE).
In [51] estimates for variance and bias of the KDE are obtained by Taylor expansion and an optimal smoothing window width, balancing variance an bias, is found as

$$h^* = \left[\frac{\int K(x)^2 \,\mathrm{d}x}{\left(\int K(x)x\right)^4 \int \rho''(x) \,\mathrm{d}x}\right]^{\frac{1}{5}} N_p^{-\frac{1}{5}}. \tag{2.69}$$



(a) Arbitrary degree B-splines (b) Epanechnikov efficiency

Figure 2.1.: Arbitrary degree B-splines as smoothing kernels and their stochastic efficiency relative to the Epanechnikov kernel with efficiency one. Note that the difference in efficiency is marginal.

Therefore, h should be chosen proportional to $N_p^{-\frac{1}{5}}$, which is a very slow convergence compared to grid based methods. In PIC the indefinite integral of a KDE is used, which can be compared with estimating the cumulative distribution function (CDF), thus alleviating this problem, see [56]. Different smoothing kernels can be discussed and the Epanechnikov kernel, given in eqn. (2.70) is found to be the $MISE$ optimal one [51] with respect to the quadratic expansion

$$K_e(x) := \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{1}{5}x^2\right) & -\sqrt{5} \leq x \leq \sqrt{5} \\ 0 & \text{otherwise} \end{cases} \tag{2.70}$$

Thus, we can compare compare any smoothing kernel K satisfying eqn. (2.59) to the Epanechnikov kernel. This is done by defining the efficiency of a smoothing Kernel relative to the Epanechnikov kernel, which has by definition efficiency one according to eqn. (2.71).

$$\text{eff}(K) := \frac{3}{5\sqrt{5}} \frac{\sqrt{\int x^2 K(x) \ \mathrm{d}x}}{\int K(x)^2 \ \mathrm{d}x} \tag{2.71}$$

Because arbitrary degree B-splines are so popular in PIC codes the Epanechnikov efficiency was calculated in fig. 2.1. Although the most efficient B-spline appears to be the quadratic the difference to the ones of higher order is rather small.

2.1.6. Stochastic errors in the particle mesh coupling

Note that in both cases (spectral and finite elements) there are two kinds of discretization errors. The first is, of course, the plain discretization error of the finite element space V_h and the finite number of Fourier modes. The second is the statistical error when estimating the coefficients $b(t)$ by the estimator $\hat{b}(t)$, also referred to as the particle noise.

Mean squared error (MSE)

For a fixed coordinate $x_0 \in \Omega_x$ the squared error for the Galerkin approximation ρ_h of ρ reads

$$\left(\rho_h(x_0, t) - \rho(x_0, t)\right)^2 . \tag{2.72}$$

The expectation of the estimator of the Galerkin approximation is the Galerkin approximation, since the expectation is linear

$$\mathbb{E}\left[\hat{\rho}_h(x,t)\right] = \mathbb{E}\left[\left(\mathcal{M}\hat{b}(t)\right)^t \cdot \psi(x)\right] = \left(\mathcal{M}\mathbb{E}\left[\hat{b}(t)\right]\right)^t \cdot \psi(x) = \left(\mathcal{M}b(t)\right)^t \cdot \psi(x) = \rho_h(x,t). \tag{2.73}$$

This describes the fact that $\hat{b}$ is an unbiased estimator for b. Comparing the particle mesh approximation $\hat{\rho}_h$ with the actual function ρ yields the squared error

$$\left(\hat{\rho}_h(x_0,t) - \rho(x_0,t)\right)^2. \tag{2.74}$$

This describes how well the combination of the mesh based approximation and the Monte Carlo estimation approximates the density ρ. It is also possible to measure, how well the Monte Carlo estimator $\hat{\rho}_h$ approximates the Galerkin discretization ρ_h by taking another squared error

$$\left(\hat{\rho}_h(x_0,t) - \rho_h(x_0,t)\right)^2. \tag{2.75}$$

In the stochastic theory there is a fundamental difference between the errors (2.74) and (2.75) such that we make a naming distinction here. Another estimator is denoted as $\hat{\rho}(x,t) := \hat{\rho}_h(x,t)$. In the following we call $\hat{\rho}(x,t)$ the estimator of the charge density ρ, and $\hat{\rho}_h$ the estimator of the Galerkin approximation of the charge density ρ. Thus, we denote the error of the particle mesh coupling as

$$\left(\hat{\rho}(x_0,t) - \rho(x_0,t)\right)^2, \tag{2.76}$$

clarifying our incentive to compare to the actual density ρ. Since $\hat{\rho}(x_0)$ is a random deviate, we take the expectation of (2.76) and define the mean squared error (MSE) as the expectation of (2.76) by

$$\begin{aligned}
\mathrm{MSE}\left[\hat{\rho}(x_0,t)\right] &:= \mathbb{E}\left[\left(\hat{\rho}(x_0,t) - \rho(x_0,t)\right)^2\right] \\
&= \mathbb{E}\left[\left(\hat{\rho}(x_0,t) - \mathbb{E}\left[\hat{\rho}(x_0,t)\right]\right)^2\right] \;+\; \left(\mathbb{E}\left[\hat{\rho}(x_0,t)\right] - \rho(x_0,t)\right)^2 \\
&= \mathbb{V}\left[\hat{\rho}(x_0,t)\right] \;+\; \left(\mathbb{E}\left[\hat{\rho}(x_0,t)\right] - \rho(x_0,t)\right)^2 \\
&= \underbrace{\mathbb{V}\left[\hat{\rho}_h(x_0,t)\right]}_{Variance} \;+\; \underbrace{\left(\rho_h(x,t) - \rho(x_0,t)\right)^2}_{Bias^2}.
\end{aligned} \tag{2.77}$$

It measures how well $\hat{\rho} = \hat{\rho}_h$ is expected to approximate ρ. The stochastic component of the MSE is the variance, which decreases with the number of particles N_p. The bias is the plain error of the Galerkin approximation at x_0 without any stochastic contribution. Increasing the number or order of finite elements or the number of Fourier modes reduces the bias, but potentially changes the variance depending on h.
This variance-bias-trade-off for kernel density estimation is extensively discussed in [51] when it comes to finding optimal smoothing parameters. There is also theory available which focuses on the expectation of the L^1 error, which is discussed in [52][pp. 40-48].
Calculating the MSE of $\hat{\rho}_h$ implies by the chosen notation that one should take the expectation of eqn. (2.75). We find it to be unbiased because of eqn. (2.73):

$$\begin{aligned}
\mathrm{MSE}\left[\hat{\rho}_h(x_0,t)\right] &:= \mathbb{E}\left[\left(\hat{\rho}_h(x_0,t) - \rho_h(x_0,t)\right)^2\right] \\
&= \mathbb{V}\left[\hat{\rho}_h(x_0,t)\right] + \left(\rho_h(x_0,t) - \rho_h(x_0,t)\right)^2 = \mathbb{V}\left[\hat{\rho}_h(x_0,t)\right].
\end{aligned} \tag{2.78}$$

To summarize, the estimator $\hat{\rho}_h$ is an unbiased estimator for ρ_h but a biased estimator for ρ. For systems that rely on the Galerkin discretization we still converge with a large number

of particles. For PIC based on a finite element subspace $V_h \subset V$ the orthogonal space $V_h^\perp$ is rather hard to imagine. When following the characteristics of the Vlasov equation (2.5) not the charge density ρ, but the electric field $E = -\nabla\Phi$ is essential. Using the estimator $\hat{\rho}$ in the Poisson equation yields in the same way estimators $\hat{\Phi}$ and $\hat{E} = -\nabla\hat{\Phi}$ for the potential Φ and the electric field E. We provide the definition of the other estimators using the same notational convention:

$$\begin{aligned} \mathrm{MSE}\left[\hat{\Phi}(x,t)\right] &:= \mathbb{E}\left[\left(\hat{\Phi}(x,t) - \Phi(x,t)\right)^2\right] \\ &= \mathbb{V}\left[\hat{\Phi}_h(x,t)\right] + \left(\Phi_h(x,t) - \Phi(x,t)\right)^2, \end{aligned} \tag{2.79}$$

$$\begin{aligned} \mathrm{MSE}\left[\hat{E}(x,t)\right] &:= \mathbb{E}\left[\left(\hat{E}(x,t) - E(x,t)\right)^2\right] \\ &= \mathbb{V}\left[\hat{E}_h(x,t)\right] + \left(E_h(x,t) - E(x,t)\right)^2. \end{aligned} \tag{2.80}$$

The bias depends entirely on the Galerkin discretization, therefore, a lot of theory is available for its estimation. For example a-posteriori estimates of the bias can be obtained from h or p refinement. Here we focus on the stochastic part such that the unknown variances $\mathbb{V}\left[\hat{\rho}(x,t)\right]$, $\mathbb{V}\left[\hat{\Phi}_h(x,t)\right]$ and $\mathbb{V}\left[\hat{E}_h(x,t)\right]$ have to be examined.

Variance-, covariance-estimation and propagation

With the PIC estimate the uncertainty lies within the determination of the right hand side $b(t) = \mathbb{E}\left[B(t)\right]$ by the Monte Carlo estimator $\hat{b}(t)$. This uncertainty also propagates, thus, making the solution vector $a(t) = \mathcal{K}b(t)$ a multivariate random deviate. Since the single entries $b_i(t)$ stem from test functions with overlapping support, the $B_i(t)$ are not independent random variables and therefore, besides the plain variance $\mathbb{V}\left[B_i(t)\right]$ knowledge of the covariance $\mathbb{COV}[B_i(t), B_j(t)]$ is essential.
We start with the variance σ_{b_n} of the nth entry in the coefficient vector estimator $\hat{b}(t)$, which can be estimated as

$$\begin{aligned} \sigma_{b_n} := \mathbb{V}\left[\hat{b}_n(t)\right] = \frac{\mathbb{V}\left[B_n(t)\right]}{N_p} &= \frac{\mathbb{V}\left[\psi_n\left(X(t)\right)W(t)\right]}{N_p} \\ &\approx \frac{1}{N_p}\,\frac{1}{N_p-1}\sum_{k=1}^{N_p}\left(\psi_n(x_k^t)w_k^t - \frac{1}{N_p}\sum_{k=1}^{N_p}\psi_n(x_k^t)w_k^t\right)^2 \\ &= \frac{1}{N_p}\,\underbrace{\frac{1}{N_p-1}\sum_{k=1}^{N_p}\left(\psi_n(x_k^t)w_k^t - \hat{b}_n(t)\right)^2}_{:=\hat{\sigma}_{b_n}} = \frac{\hat{\sigma}_{b_n}}{N_p}. \end{aligned} \tag{2.81}$$

Here the estimator $\hat{\sigma}_{b_n}$ is the unbiased sample variance and thus and unbiased estimator for $\mathbb{V}\left[B_n(t)\right]$. Its uncertainty can be checked with a re-sampling method such as the jackknife [57]. In complete analogy we estimate the covariance matrix σ_b of B

$$\begin{aligned} (\Sigma_b)_{i,j}(t) = \mathbb{COV}[B_i(t), B_j(t)] &= \mathbb{E}\left[\left(B_i(t) - \mathbb{E}[B_i(t)]\right)\left(B_j(t) - \mathbb{E}[B_j(t)]\right)^\dagger\right] \\ &= \mathbb{E}\left[B_i(t)B_j(t)^\dagger\right] - \mathbb{E}[B_j(t)]\mathbb{E}[B_j(t)]^\dagger \\ &= \mathbb{E}\left[B_i(t)B_j(t)^\dagger\right] - b_i(t)b_j(t)^\dagger \end{aligned} \tag{2.82}$$

with the unbiased covariance estimator, also known as the unbiased sample covariance

$$(\hat{\Sigma}_b)_{i,j}(t) = \frac{1}{N_p - 1} \sum_{k=1}^{N_p} \left(\psi_i\left(x_k(t)\right) w_k^t - \hat{b}_i(t)\right) \left(\psi_j\left(x_k(t)\right) w_k^t - \hat{b}_j(t)\right)^\dagger, \quad \forall i,j = 1,\dots,N_f. \tag{2.83}$$

The † denotes the Hermite adjoint, which is for a matrix the transpose and complex conjugate. Compared to the estimate of the mean in eqn. (2.52), estimating the full covariance matrix (2.83) can be rather expensive because it is always a dense matrix. Thus for an orthogonal basis ψ it is then more efficient to store the sparse matrix $\left(\mathbb{E}\left[B_i(t)B_j(t)\right]\right)_{i,j=1,\dots,N_f}$ and the right hand side $b(t)$.
We a present rather crude method to gain estimates for the covariance matrix and drop the time dependence for sake of notation. If the marginal probability density of X denoted by $p_X(x) = \int_{\mathbb{R}} g(x,v)\ \mathrm{d}v$ is given, and the the sampling density g coincides with f up to a constant $g \cdot m = f$ then Σ_b is known. The constant m normalizing the density f is

$$m := \iint f(x,v)\ \mathrm{d}x\mathrm{d}v = \int \rho(x)\ \mathrm{d}x. \tag{2.84}$$

Then the entries of Σ_b reduce to

$$\begin{aligned}(\Sigma_b)_{i,j} = \mathbb{COV}[B_i, B_j] &= \mathbb{COV}[W\psi_i(X), W\psi_j(X)] = \mathbb{COV}[m\psi_i(X), m\psi_j(X)] \\ &= \mathbb{E}\left[m\psi_i(X)m\psi_j(X)^\dagger\right] - \mathbb{E}\left[m\psi_i(X)\right] - \mathbb{E}\left[m\psi_i(X)\right] \\ &= m^2\mathbb{E}\left[\psi_i(X)\psi_j(X)^\dagger\right] \ - \ b_i b_j^\dagger.\end{aligned} \tag{2.85}$$

The estimator $\hat{b}$ provides an estimate for b but for the second moment there is nothing so far. Using the probability density p_X yields another expression for the second moment

$$\mathbb{E}\left[\psi_i(X)\psi_j(X)^\dagger\right] = \int_{\Omega_x} \psi_i(x)\psi_j(x)p_X(x)\ \mathrm{d}x. \tag{2.86}$$

Here of course p_X is not given but it can be approximated using the discretized charge density ρ_h provided b that is estimated anyhow in the simulation.

$$\begin{aligned}p(x) = \frac{\rho(x)}{m} &\approx \frac{\rho_h(x)}{m} = \frac{1}{m}(\mathcal{M}b)^\dagger\psi(x) \\ &\approx \frac{\hat{\rho}_h(x)}{m} = \frac{1}{m}(\mathcal{M}\hat{b})^\dagger \cdot \psi(x)\end{aligned} \tag{2.87}$$

The finite element approximation of the second moment reads then

$$\mathbb{E}\left[\psi_i(X)\psi_j(X)^\dagger\right] \approx \int_{\Omega_x} \psi_i(x)\psi_j(x)\frac{1}{m}(\mathcal{M}b)^\dagger \cdot \psi(x)\ \mathrm{d}x = \sum_{k=1}^{N_f} \frac{1}{m} \int_{\Omega_x} \psi_i(x)\psi_j(x)\psi_k(x)\ \mathrm{d}x\ (\mathcal{M}b)_k. \tag{2.88}$$

The approximation of the second moment requires only an estimate of b and the integrals $\int_{\Omega_x} \psi_i(x)\psi_j(x)\psi_k(x)\ \mathrm{d}x$ which only involves the basis functions. This means that one obtains by exploiting the Galerkin Ansatz the covariance matrix in eqn. (2.85) directly as a function of the estimator $\hat{b}$ without any additional operation on the samples. Dropping the component wise notation yields the following notation

$$\Sigma_b = \mathbb{COV}\left[m\psi(X)\right] = m^2\mathbb{COV}\left[\psi(X)\right] = m^2\mathbb{E}\left[\psi(X)\psi(X)^\dagger\right] - \underbrace{\mathbb{E}\left[m\psi(X)\right]\mathbb{E}\left[m\psi(X)\right]^\dagger}_{=bb^\dagger} \tag{2.89}$$

Inserting the finite element approximation for ρ_X yields the corresponding matrix for the second moment

$$\begin{aligned}\mathbb{E}\left[\psi(X)\psi(X)^t\right] &= \int_{\Omega_x} \psi(x)\psi(x)^\dagger p(x)\ \mathrm{d}x \\ &\approx \int_{\Omega_x} \psi(x)\psi(x)^\dagger \frac{1}{m}\rho_h(x)\ \mathrm{d}x = \frac{1}{m}\int_{\Omega_x} \psi(x)\psi(x)^\dagger (\mathcal{M}b)^\dagger \psi(x)\ \mathrm{d}x. \quad (2.90)\end{aligned}$$

This can by simplified by defining a matrix Q as

$$Q_{i,j} := \int_{\Omega_x} \psi_i^2(x)\psi_j(x)\ \mathrm{d}x, \quad i,j = 1,\dots,N_f. \quad (2.91)$$

For orthogonal basis functions the matrix Q has the same sparsity pattern as the mass matrix M and is therefore cheap to obtain. If there is not enough memory available to store Q this tensor also can be calculated by a finite element assembly for every b. The second moment reads then

$$\mathbb{E}\left[\psi(X)\psi(X)^t\right] \approx \frac{1}{m}\left((Q\mathcal{M}b) + (Q\mathcal{M}b)^\dagger\right), \quad (2.92)$$

which inserted into eqn. (2.89) yields

$$\Sigma_b = \mathbb{COV}\left[m\psi(X)\right] \approx m\left((Q\mathcal{M}b) + (Q\mathcal{M}b)^\dagger\right) - bb^\dagger. \quad (2.93)$$

For a special case of a uniform charge density and a uniform sampling the covariance matrix can be directly obtained from the mass matrix $\mathcal{M}$. Denote the volume of the domain Ω_x by $|\Omega_x| = \int_{\Omega_x} \mathrm{d}x$. The covariance under uniform sampling is then given in eqn. (2.97).

$$|\Omega_x| := \int_{\Omega_x} \mathrm{d}x \quad (2.94)$$

$$c = \int_{\Omega_x} \psi(x)\ \mathrm{d}x \quad (2.95)$$

$$M = \int_{\Omega_x} \psi(x)\psi(x)^t\ \mathrm{d}x \quad (2.96)$$

$$\mathbb{COV}\left[\psi(X)\right] = \frac{1}{|\Omega_x|}M - \frac{1}{|\Omega_x|^2}bb^t \quad (2.97)$$

Whether reconstructing the charge density ($\mathcal{M}$) or solving the Poisson equation ($\mathcal{K}$), both operations on $b(t)$ are linear. Let $Y = (Y_1,\dots Y_N)^t$ be a multivariate random deviate with covariance matrix $\Sigma_Y = \mathbb{COV}\left[Y\right]$ and let $A \in \mathbb{R}^{N\times N}$ denote a linear operator in form of a matrix. By linear covariance propagation [58][p. 16] Σ_Y can be propagated through the linear operation by

$$\mathbb{COV}\left[AY\right] = A\mathbb{COV}\left[Y\right]A^\dagger = A\Sigma_Y A^\dagger \quad (2.98)$$

Note that nonlinear covariance propagation through a nonlinear function φ with Jacobian J_φ can be approximated by use of the Taylor expansion as

$$\mathbb{COV}\left[\varphi(Y)\right] \approx J_\varphi \mathbb{COV}\left[Y\right]J_\varphi^\dagger. \quad (2.99)$$

By linear covariance propagation, the covariance matrix of the solution vector $a(t) = \mathcal{K}b(t)$ is given as

$$\Sigma_a(t) = \mathcal{K}\Sigma_b(t)\mathcal{K}^\dagger \quad \Leftrightarrow \quad \Sigma_a(t) = \mathcal{K}\left(\mathcal{K}\Sigma_b^\dagger(t)\right)^\dagger. \quad (2.100)$$

Note that the variance of the standard Monte Carlo estimator decreases with N_p such that we denote

$$\Sigma_{\hat{b}} = \frac{1}{N_p}\Sigma_b \quad \text{and} \quad \Sigma_{\hat{a}} = \frac{1}{N_p}\Sigma_a. \tag{2.101}$$

This allows, with more linear algebra from [58], to calculate the variance of the field estimator for $x \in \Omega$.

$$\mathbb{V}[\hat{\Phi}_h(x)] = \mathbb{V}\left[\hat{a}(t)^t\psi(x)\right] = \psi(x)^\dagger\Sigma_{\hat{a}}\psi(x) \quad = \psi(x)\psi(x)^\dagger \circ \Sigma_{\hat{a}} \tag{2.102}$$

Here $A \circ B = \sum_{i,j} A_{i,j}B_{i,j}$ denotes the Hadamard product. The same is possible for the vector valued electric field

$$\mathbb{V}[\hat{E}_h(x)] = \mathbb{V}[-\nabla\hat{\Phi}_h(x)] = \mathbb{V}\left[\hat{a}(t)^t\nabla\psi(x)\right] = \nabla\psi(x)^\dagger\Sigma_{\hat{a}}\nabla\psi(x) \tag{2.103}$$

and the charge density estimator

$$\mathbb{V}[\hat{\rho}_h(x)] = \mathbb{V}\left[(\mathcal{M}\hat{b}(t)) \cdot \psi(x)\right] = \psi(x)^\dagger\mathcal{M}\Sigma_{\hat{b}}(t)\mathcal{M}^\dagger\psi(x). \tag{2.104}$$

For estimation of these quantities during the simulation we just plug in the covariance estimates. We go one step further and expand the calculations from the local error at x to the more global L^2 norm.

Mean integrated squared error (MISE)

The mean integrated squared error (MISE) [51][p. 35] of the density estimator $\hat{\rho}$ is given as the expectation of the squared L^2 error, also referred to as the *integrated squared error* ISE. The MISE is, by Fubinis theorem, equivalent to the L^2 error of the expectation (IMSE). Here we can split up the $MISE$ into two parts.

$$\begin{aligned}\mathrm{MISE}(\hat{\rho}) :=& \mathbb{E}\left[\int \left(\hat{\rho}(x,t) - \rho(x,t)\right)^2 \,\mathrm{d}x\right] \\ =& \int \mathbb{V}\left[\hat{\rho}(x,t)\right] \,\mathrm{d}x \quad + \quad \int \left(\rho(x,t) - \mathbb{E}\left[\hat{\rho_h}(x,t)\right]\right)^2 \,\mathrm{d}x \\ =& \underbrace{\mathrm{IVAR}\left[\hat{\rho}(x,t)\right]}_{\text{integrated variance}} \quad + \quad \underbrace{\int \left(\rho(x,t) - \rho_h(x,t)\right)^2 \,\mathrm{d}x}_{\text{integrated bias}^2}\end{aligned} \tag{2.105}$$

As the variance $\mathbb{V}\left[\hat{\rho}(x,t)\right]$ is known from (2.104) we can calculate the integrated variance accordingly. The definition can be applied on the potential and its gradient using the definition in (2.105).

$$\mathrm{MISE}\left[\hat{\Phi}\right] = \int \mathbb{V}\left[\hat{\Phi}(x,t)\right] \,\mathrm{d}x \; + \; \int \left(\Phi(x,t) - \Phi_h(x,t)\right)^2 \,\mathrm{d}x \tag{2.106}$$

$$\begin{aligned}\mathrm{MISE}\left[\hat{E}\right] &= \mathrm{MISE}(\nabla\hat{\Phi}) \\ &= \int \mathbb{V}\left[\nabla\hat{\Phi}(x,t)\right] \,\mathrm{d}x \; + \; \int \|\nabla\Phi(x,t) - \nabla\Phi_h(x,t)\|^2 \,\mathrm{d}x\end{aligned} \tag{2.107}$$

Here as an extension to Parzen windows [51][p. 40], the main interest lies not in estimating the charge density ρ but in obtaining the electric field after solving the Poisson equation.
For the sake of notation set $\mathcal{M}^\dagger\Sigma_{b(t)}\mathcal{M} = \Sigma = (\sigma_{i,j})_{i,j=1,\dots,N_h}$ and recall the sparse mass

matrix $M_{i,j} = \int_\Omega \psi_i(x)\psi_j(x)dx$, then the integrated variance of the charge density is given as

$$\begin{aligned}\mathrm{IVAR}[\hat{\rho}_h(x)] &= \int_{\Omega_x} \mathbb{V}\left[\hat{\rho}_h(x)\right] \, \mathrm{d}x = \int_{\Omega_x} \psi(x)^\dagger \Sigma \psi(x) \, \mathrm{d}x \\ &= \int_{\Omega_x} \sum_{i,j=1}^{N_h} \psi_j(x)^\dagger \sigma_{i,j} \psi_i(x) \, \mathrm{d}x = \int_{\Omega_x} \sum_{i,j=1}^{N_h} \psi_j(x)^\dagger \sigma_{i,j} \psi_i(x) \, \mathrm{d}x \\ &= \sum_{i,j=1}^{N_h} \sigma_{i,j} \int_{\Omega_x} \psi_i(x)\psi_j(x)^\dagger \, \mathrm{d}x = \sum_{i,j=1}^{N_h} \sigma_{i,j} \, M_{i,j} = \Sigma \circ M. \end{aligned} \tag{2.108}$$

Here $A \circ B$ is called the Hadamard product of two matrices. Applying the same method as in (2.108) yields the integrated variance of the potential estimator

$$\mathrm{IVAR}[\hat{\Phi}_h(x)] = \Sigma_{a(t)} \circ M_{i,j}. \tag{2.109}$$

Using $K_{i,j} = \int_\Omega \nabla\psi_i(x)\nabla\psi_j(x)^\dagger dx$ and setting $\Sigma = \Sigma_{a(t)}$, the integrated variance of the electric field reads

$$\mathrm{IVAR}[\nabla\hat{\Phi}_h(x)] = \sum_{i,j=1}^{N_h} \sigma_{i,j} \int_\Omega \nabla\psi_i(x)\nabla\psi_j(x)^\dagger \, \mathrm{d}x = \sum_{i,j=1}^{N_h} \sigma_{i,j} \, K_{i,j} = \Sigma_{a(t)} \circ K_{i,j}. \tag{2.110}$$

In a last step we can extend this to the variance of the electric field energy (2.3). With $a(t) := \mathcal{K}b(t)$ and the corresponding covariance matrix $\Sigma_{a(t)}$.

$$\hat{H}_E(t) = \int \|\nabla\hat{\Phi}(x,t)\|^2 \, \mathrm{d}x = \int \|a(t)^t \nabla\psi(x)\|^2 \, \mathrm{d}x = \hat{a}(t)^\dagger K \hat{a}(t) \tag{2.111}$$

The electrostatic energy is a quadratic form and its expected value [59][pp. 51] reads

$$\mathbb{E}\left[\hat{H}_E(t)\right] = \mathbb{E}\left[\hat{a}(t)\right]^\dagger K \mathbb{E}\left[\hat{a}(t)\right]^\dagger = a(t)^\dagger K a(t) + \mathrm{tr}\left(K\Sigma_{a(t)}\right) \tag{2.112}$$

and the corresponding variance [59][pp. 75] is

$$\mathbb{V}\left[b^\dagger K b\right] = 2tr\left(K\Sigma_{a(t)}K\Sigma_{a(t)}\right) + 4\mathbb{E}\left[\hat{a}(t)\right]^\dagger K\Sigma_{a(t)}K\mathbb{E}\left[\hat{a}(t)\right]. \tag{2.113}$$

The variance of the electrostatic field energy (2.113) can be an additional useful diagnostic.

2.1.7. Mean field theory and the Vlasov–McKean equation

So far we have learned that each individual particle follows a stochastic process which needs an electric field. Yet now this electric field suffers from some error because it is obtained by taking a sample mean. So it is an open question whether this system of particles will converge to the right electric field. A stochastic answer can be found by considering this propagation of chaos in [60]. Our discretized Vlasov–Poisson system is then merely a special case of a Vlasov–McKean equation. We need some additional form of diffusion, a Brownian motion on the particle trajectories, yet this can be arbitrarily small. The remaining work is the translation of the notations from the laboratory problem in [60] and the Vlasov McKean equation in [17]. A theoretical overview on Vlasov–McKean can be found in [61] along with an detailed explanation concerning the propagation of chaos. This property just states, that for $N_p \to \infty$ the particles become less and less correlated which is the ultimate justification for all Monte Carlo discretizations of Vlasov equations used in this thesis. In this thesis we are interested in the fluctuations, the small discrepancy to the true solution for a finite but

large N_p. From *Section 5. Convergence of the fluctuations for the McKean-Vlasov model* in [61] we learn that these fluctuations, live in an exotic Sobolev space, have the martingale property and converge to an Ornstein–Uhlenbeck process (diffusion) and can be exponentially bounded by the overall simulation time. This means the simulation gets worse over time and we can only assume for the large particle limit diffusion like error propagation. Because all this work is done in the large particle limit it is safe to say that a badly resolved simulation does not simply correspond to a model with large diffusion.
The electric field obtained with the finite elements and the mean over all particles can be written as

$$E(x,t) \approx \hat{E}(x,t,X_1(t),\dots,X_n(t)) = \left[\mathcal{K}\frac{1}{N_p}\sum_{n=1}^{N_p}\psi\left(X_n(t)\right)\right]^t \psi(x), \quad x \in [0,L]. \tag{2.114}$$

The Particle-In-Fourier method provides us with a much more accessible formula because the Poisson equation an be solved directly in Fourier space. The electric field is then directly obtained by the convolution of Fourier modes with the density f. This also works for the finite elements, or any other orthogonal series resulting in a messy notation. Note that we remove the zeroth Fourier mode $k = 0$ because of the constant density background. We leave the series untruncated, but it can be truncated at any time without loss of generality.

$$\begin{aligned} E(x,t) &= \sum_{k\neq 0} \mathrm{e}^{\mathrm{i}kx}\frac{1}{\mathrm{i}k}\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_0^{2\pi}\mathrm{e}^{\mathrm{i}ky}f(y,v,t)\,\mathrm{d}x\mathrm{d}v \\ &= \sum_{k\neq 0}\frac{1}{\mathrm{i}k}\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_0^{2\pi}\mathrm{e}^{\mathrm{i}k(x-y)}f(y,v,t)\,\mathrm{d}x\mathrm{d}v \\ &= \sum_{k\neq 0}\left[\left((x,v)\mapsto\frac{\mathrm{e}^{\mathrm{i}kx}}{\mathrm{i}k2\pi}\right) * f(\cdot,\cdot,t)\right](x) \end{aligned} \tag{2.115}$$

We continue discretizing eqn. (2.115) with the markers $X_n(t)$ yielding the interaction term $\hat{E}$ in eqn. (2.116).

$$\hat{E}(x,t,X_1,\dots,X_n) = \frac{1}{N_p}\sum_{n=1}^{N_p}\sum_{k\neq 0}\frac{\mathrm{e}^{\mathrm{i}kX_n(t)}}{\mathrm{i}k2\pi} \tag{2.116}$$

Note that $\hat{E}(x,t,X_1,\dots,X_{N_p})$ is bounded and Lipschitz continous thus fulfilling the requirements for propagation of chaos in [60][p.172], which can actually be relaxed. We add some small diffusion $\sigma_x,\sigma_v \geq 0$ onto the trajectories by defining an independent Brownian motion $B_t^{m,x}, B_t^{m,v}$ for the velocity and spatial component of every particle. For physical collisions of particles the diffusion only acts in velocity space, yielding the special case $\sigma_x = 0$, which will not be treated separately here.

$$\begin{aligned} \mathrm{d}X^m(t) &= V^m(t) + \sigma_x B_t^{m,x} \\ \mathrm{d}V^m(t) &= \hat{E}(X^m(t),t,X_1,\dots,X_{N_p}) + \sigma_v B_t^{m,v}, \quad m = 1,\dots,N_p \end{aligned} \tag{2.117}$$

As the number of markers increases $N_p \to \infty$ the trajectories in eqn. (2.117) converge to (2.118) according to eqn. (2.119).

$$\begin{aligned} \mathrm{d}\bar{X}^m(t) &= \bar{V}^m(t) + \sigma_x B_t^{m,x} \\ \mathrm{d}\bar{V}^m(t) &= \mathbb{E}\left[\hat{E}(\bar{X}^m(t),t,\bar{X}_1,\bar{X}_2,\dots)|\bar{X}^m\right] + \sigma_v B_t^{m,v}, \quad m = 1,2,\dots \end{aligned} \tag{2.118}$$

$$\sup_{N_p}\sqrt{N_p}\left[\sup_{t\leq T}\left|(X^m(t),V^m(t)) - (\bar{X}^m(t),\bar{V}^m(t))\right|\right] < \infty \tag{2.119}$$

In the next step we revert to the original stochastic process $(X(t), V(t))$ with distribution $f(x, v, t)$, also called the law of $(X(t), V(t))$.

$$\begin{aligned} \mathrm{d}X(t) &= V(t)\mathrm{d}t + \sigma B_t^x \\ \mathrm{d}V(t) &= E(X(t), t)\mathrm{d}t + \sigma B_t^v = \sum_{k\neq 0} \frac{1}{\mathrm{i}k} \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_0^{2\pi} \mathrm{e}^{\mathrm{i}k(X(t)-y)} f(y, v, t)\ \mathrm{d}x\mathrm{d}v\ \mathrm{d}t + \sigma B_t^v \end{aligned} \tag{2.120}$$

The stochastic differential equation in eqn. (2.120) is equivalent to the Vlasov–Poisson system with diffusion in

$$\partial_t f(x, v, t) + v\nabla_x f(x, v, t) + E(x, t)\nabla_v f(x, v, t) = \frac{\sigma^2}{2} \triangle f(x, v, t) \tag{2.121}$$

$$\partial_x E(x, t) = \int_{-\infty}^{\infty} f(x, v, t)\ \mathrm{d}v - 1. \tag{2.122}$$

Note that we can also set the diffusion to zero $\sigma = 0$ in order to obtain the standard Vlasov equation [60]. Instead of adding diffusion one can regularize the Vlasov equation such that some moments are conserved, see [62] for a detailed treatment of the Vlasov–Maxwell system. Here we will not go deeper into mean field theory, but note two things. There exists a stochastic description of our particle method such that it converges to the Vlasov equation we wanted to approximate. It is actually a strong convergence [17]. Second, if the interaction field is modified without changing the assumptions and the mean then the method still converges. This allows the use of variance reduction methods and even multilevel Monte Carlo methods in the time domain [17].

2.2. Sampling and variance

Different physical scenarios can be modeled with the Vlasov–Poisson system, where most test-cases correspond to a unique initial condition along with some parameters. We begin with the most basic test case: Langmuir waves are linear Landau damped with small amplitude $\epsilon = 10^{-2}$ and nonlinear with large amplitude $\epsilon = 0.5$. The initial condition reads

$$f(t = 0, x, v) := (1 + \epsilon \cos(kx)) \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{v^2}{2}}, \quad x \in [0, L], \tag{2.123}$$

where the length of the periodic box is given by the wave vector k as

$$L = \frac{k}{2\pi}, \quad k = 0.5. \tag{2.124}$$

The initial electric energy is

$$\frac{1}{2} \int_0^L (\epsilon \sin(kx))^2\ \mathrm{d}x = \frac{1}{2} \frac{\epsilon^2 L}{2k^2} = \frac{\pi\epsilon^2}{2k^3} \tag{2.125}$$

and the kinetic energy is

$$\frac{1}{2} \int_{\mathbb{R}} \int_0^L \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(v-\mu)^2}{2\sigma^2}}\ \mathrm{d}x\mathrm{d}v = \frac{1}{2} \left(\sigma^2 + \mu^2\right) L. \tag{2.126}$$

The Shannon entropy for the Maxwellian reads

$$\int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(v-\mu)^2}{2\sigma^2}} \ln\left(\frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(v-\mu)^2}{2\sigma^2}}\right)\ \mathrm{d}v = \ln\left(\sigma\sqrt{2\pi\mathrm{e}}\right), \tag{2.127}$$

yet the analytic determination of the entropy of $f(t=0)$ by including the spatial perturbance is difficult so we fall back on numerical values.
Introducing a small portion n_b of fast electrons makes the system unstable, resulting in the bump-on-tail instability [63] with initial condition along with typical parameters given in eqn. (2.128).

$$\begin{aligned} f(x,v,t=0) &:= (1-\epsilon\cos(kx))\,\frac{1}{\sqrt{2\pi}}\left((1-n_b)\mathrm{e}^{-\frac{v^2}{2}}+\frac{n_b}{\sigma}\mathrm{e}^{-\frac{(v-v_0)^2}{2\sigma^2}}\right) \\ L &= \frac{2\pi}{k},\ \sigma = 0.5, n_b = 0.1,\ k = 0.3,\ v_0 = 4.5,\ \epsilon = 0.03 \end{aligned} \tag{2.128}$$

2.2.1. Importance sampling

Although improvements in the convergence with respect to the number of particles N_p can be achieved by enhancing the initial sampling [64], we stay with a rather simple choice for the sampling density g. The estimation of a moment $\mathbb{E}\left[\Theta(X,V)W(t)\right]$ requires the weight $W(t) = \frac{f(X(t),V(t),t)}{g(X(t),V(t),t)}$. In order to keep the variance $\mathbb{V}\left[\Theta(X,V)W(t)\right]$ small for any Θ, the first step is to minimizing the variance of the weights $\mathbb{V}\left[W(t)\right] = \mathbb{V}\left[\frac{f(X(t),V(t),t)}{g(X(t),V(t),t)}\right]$. The optimum of course is found, when $g(x,v,t=0)$ is chosen as the probability closest to $f(x,v,t=0)$, which we denote as importance sampling. In the optimal case, one chooses

$$g(x,v,t) = \frac{f(x,v,t)}{\iint_\Omega f(x,v,t)\ \mathrm{d}x\mathrm{d}v} \Rightarrow \mathbb{V}\left[W(t)\right] = \mathbb{V}\left[\frac{1}{\iint_\Omega f(x,v,t)\ \mathrm{d}x\mathrm{d}v}\right] = 0. \tag{2.129}$$

In the following we give examples for possible choices of g and how to sample directly from g. To gain some flexibility in treating arbitrary initial distributions, one can of course use an accept rejection algorithm [65][p. 11-12] for all test-cases, even in combination with low-discrepancy sequences [66]. Because most of our examples exhibit simple structure we mostly use inverse transform sampling.

2.2.2. Spatial disturbance

Suppose a sampling distribution $g(x) = \frac{1}{L}\left(1+\epsilon cos(k_x x)\right)$ with $\frac{2\pi}{L}n = k_x$ and $\int_0^L g(x)\ dx = 1$. To draw markers $x_k,\ k = 1,\ldots,N_p$ according to g we use inverse transform sampling. The cumulative distribution function $G : [0,L] \to [0,1]$

$$G(y) := \int_0^y g(x)\ dx = \frac{1}{L}\left(y + \frac{\epsilon}{k_x}cos(k_x y)\right) \tag{2.130}$$

with its inverse $G^{-1}(u) = y$ for $u \in [0,1]$. For every $u \in [0,1]$ one can solve for x

$$G(x) = u \Leftrightarrow x + \frac{\epsilon}{k_x}cos(k_x x) = Lu. \tag{2.131}$$

Often the inversion is done by one Picard (fixed point) iteration on eqn. (2.131), see [5][p. 22]. For the sake of exactness we use Newton's method and define $z := Lu - x$ and, therefore, $x = Lu - z$ such that we have to solve

$$F(z) = \frac{\epsilon}{k}sin(k(Lu-z)) - z = 0. \tag{2.132}$$

Note that the derivative is given as $\frac{d}{dz}F(z) = -\epsilon cos(k(Lu-z)) - 1$ which can be inserted in the Newton iteration, eqn. (2.133).

$$z_{k+1} := z_k - \frac{F(z_k)}{\frac{d}{dz}F(z_k)} \tag{2.133}$$

This can be done in parallel for all markers, such that the overall algorithm reads:
For $k = 1, \ldots, N_p$

1. Draw i.i.d. $u_k \sim \mathcal{U}(0,1)$
2. Find $x_k \in [0, L]$ such that $G(x_k) = u_k$

2.2.3. Moment matching

When sampling a density by drawing markers from a random distribution, some analytical moments of this distribution are known. For a time dependent particle simulations analytically known moments are the conserved ones. This knowledge can be used as a direct variance reduction technique, called moment matching [65][p. 15]. Very small modification to the randomly drawn samples can make the discrete Monte Carlo estimators of a moment to estimate a desired value exactly. Suppose $v_1, \ldots, v_{N_p} \sim v \sim \mathcal{N}(\mu_1, \sigma^2)$ are i.i.d.

$$\mu_1 := \mathbb{E}[v], \quad \hat{\mu}_1 = \frac{1}{N_p}\sum_{k=1}^{N_p} v_k, \quad \mu_2 := \sigma^2 - \mu_1^2 = \mathbb{E}[v^2], \quad \hat{\mu}_2 = \frac{1}{N_p}\sum_{k=1}^{N_p} v_k^2 \tag{2.134}$$

Search for a preferably simple transformation $T : \mathbb{R} \to \mathbb{R},\ v^* = T(v)$ such that

$$\hat{\mu}_1^* := \frac{1}{N_p}\sum_{k=1}^{N_p} T(v_k) = \mu_1 \text{ and } \hat{\mu}_2^* := \frac{1}{N_p}\sum_{k=1}^{N_p} T(v_k)^2 = \mu_2. \tag{2.135}$$

A linear Ansatz for T yields for all $k = 1, \ldots, N_p$

$$v_k^* = T(v_k) = (v_k - \mu_1)c + m_1, \quad c := \sqrt{\frac{(m_2 - m_1^2)}{(\mu_2 - \mu_1^2)}}. \tag{2.136}$$

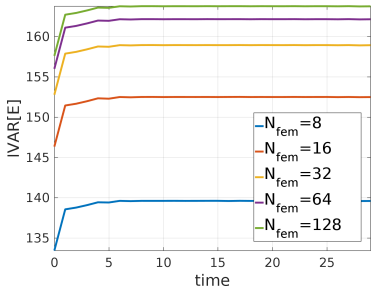
This can be done at initialization [67], since energy and momentum are known exactly. It can be done especially for any distribution with known first and second moment. Under a (Fokker–Planck) collision scheme the velocity of all particles has been stochastically modified, which means it was re-sampled. The momentum $\mu_1 = 0$ is then known but for the kinetic energy only an estimate before the collision step is available. Now a valid choice is to set $m_1 = \mu_1 = 0$ and for the second moment the resulting discrete term $m_2 = \mu_2$. This corrects the impulse, while not changing the kinetic energy. Later we will have a look at this particular example and compare it to variance reduction techniques.
If the phase space position of the markers was not modified but only the weights w_k it is then straight forward to just touch the weights. In general the weights can always be used to match the density.

$$\hat{\mu}_n = \frac{1}{N_p}\sum_{k=1}^{N_p} w_k (v_k)^n,\ \hat{v}_n = \frac{1}{N_p}\sum_{k=1}^{N_p} (v_k)^n,\ \hat{\lambda}_n = \frac{1}{N_p}\sum_{k=1}^{N_p} (w_k)^n$$

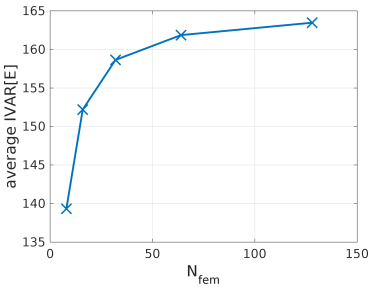
A linear Ansatz for matching μ_1 and μ_2 by only manipulation of the weights yields

$$w_k^* = T(w_k) := \frac{m u_1 \hat{v}_2 - \mu_2 \hat{v}_1}{\hat{\mu}_1 \hat{v}_2 - \hat{\mu}_2 \hat{v}_2} w_k - \frac{\mu_1 \hat{\mu}_2 - \mu_2 \hat{\mu}_1}{\hat{\mu}_1 \hat{v}_2 - \hat{\mu}_2 \hat{v}_1}.$$

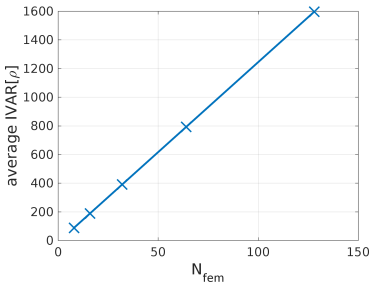
The velocity moments can be set/kept, but mass conservation $\hat{\lambda}_1 = \lambda_1 = 0$ is lost. The most straightforward approach is to match the δ-weights to the corresponding mass [68]. This is just a special case of general re-weighting techniques, where a chapter can be found here [69].



(a) Integrated variance of the electric field.

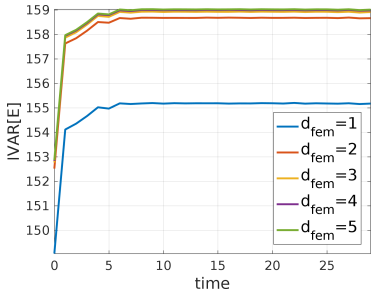


(b) Time averaged integrated variance of the electric field.

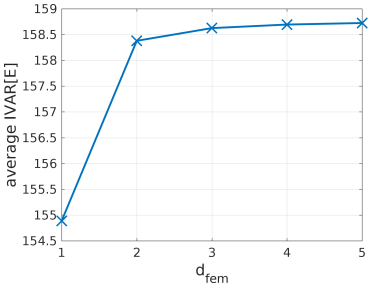


(c) Time averaged integrated variance of the charge density ρ.

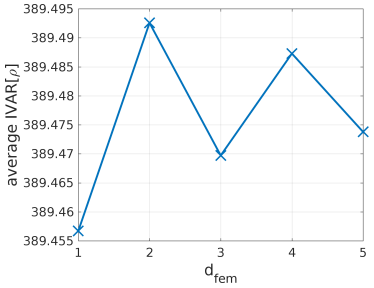
Figure 2.2.: Integrated variance for nonlinear Landau damping simulations using PIC with cubic B-splines ($d_{\text{fem}} = 3$) and varying number of cells N_{fem}.



(a) Integrated Variance of the electric field.



(b) Time averaged integrated variance of the electric field.



(c) Time averaged integrated variance of the charge density ρ.

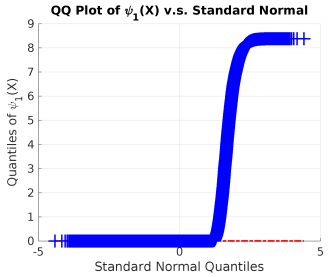
Figure 2.3.: Integrated variance for nonlinear Landau damping simulations using PIC with $N_{\text{fem}} = 32$ cells and B-splines of varying order d_{fem}.

2.2.4. Particles per cell

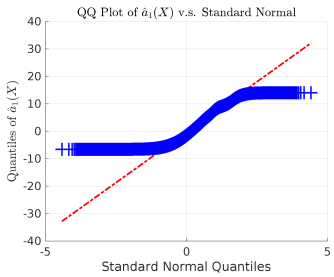
On of the first questions when setting up a simulation is: How many particles are needed? Increasing the cell size - number of finite elements N_{fem} - undoubtedly decreases the bias on the electric field, because of the better discretization. The number of particles per cell should be at least constant, otherwise the variance on the electric field estimator will increase again. Since we have the integrated variance of the electric field available as a diagnostic we can now demonstrate the behavior depending on the number of particles per cell.
We run nonlinear Landau damping $N_{\text{fem}} = 8, \ldots, 128,\ , N_p = 5 \cdot 10^5,\ \Delta t = 0.05$ and $rk3s$. Fig. 2.2a shows a slight increase of the variance with the transition to the nonlinear phase, which can be explained by the additional modes being present. As expected the variance increases with decreasing cell size $h = \frac{L}{N_{\text{fem}}}$. Time averaging the integrated variances shows that the variance of the charge density increases linear with the number of cells, see fig. 2.2c. This implies to keep the number of particles per cell at least constant. But IVAR$[\hat{E}]$ in fig. 2.2b does not increase the same way because the Laplace operator damps the higher modes.

Similar results in the same setting are obtained by varying the spline degree d_{fem}, yet here the increase in variance is very small, see fig. 2.3a, 2.3c and 2.3b.

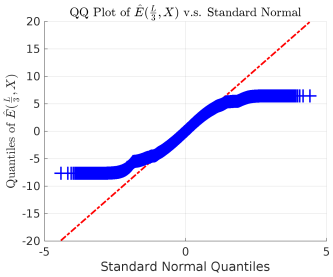
Yet the this is only half of the picture because we do not have an a-posteriori estimate of the discretization error, who's dependence on the cell size is dominated by the interpolation error.



(a) Quantiles for the right hand side Monte Carlo estimator of the first B-spline $\psi_1(X)$.



(b) Quantiles for the Monte Carlo estimator of the first B-spline $\psi_1(X)$ coefficient of Φ_h.



(c) Quantiles for the electric field estimator at the arbitrary position $x_0 = \frac{L}{3}$.

Figure 2.4.: After the variance σ^2 and standard deviation σ of a quantity, here coefficients of various fields, are made it is misleading to jump directly to the standard error bars by adding $\pm\sigma$. This requires the quantity to be normally distributed, which can be tested by comparing the quantiles to the standard normal distribution. The quantiles of the Monte Carlo estimators used in a Landau damping PIC simulations are heavily tailed, such that the error is not normally distributed.

We know the discretization error to be $\mathcal{O}\left(\left(\frac{1}{N_{\text{fem}}}\right)^{d_{\text{fem}}+1}\right)$ so in order to balance variance and bias, it seems reasonable to first go high order and then increasing the cell size, since the variance does only mildly depend on the order. This result suggests that spectral methods with few degrees of freedom but high order might be better suited. An example is orthogonal series density estimation (OSDE), which will be discussed later.

2.2.5. Variance for error estimation

Although, by the strong law of large numbers the standard Monte Carlo estimator approaches asymptotic normality, we do not necessarily observe normality in our estimators. Therefore, it is not suitable to use the standard normal quantiles to obtain confidence intervals using the estimated variance. We plot the quantiles of various estimators versus the respective normal quantiles at $t = 10$ for nonlinear Landau damping $N_{\text{fem}} = 32, N_p = 1 \cdot 10^5$, $\Delta t = 0.01$ and $rk3s$.
Let $x_k,\ k = 1, \ldots, N_p$ be the samples of the distribution X as introduced before. Since we use importance sampling with constant weights, we drop them from the notation. Then we can measure the quantiles for the first Ansatz function of the right hand side $b_1(X) = \psi_1(X)$ by the samples $\psi_1(x_k)$, which diverges far from normality c.f. fig. 2.4a. Since the Poisson equation for the random variable $a(X) := \mathcal{K}\psi(X)$ is linear, we can solve it for every sample particle $a(x_k) := \mathcal{K}\psi(x_k)$ and plot the estimated quantiles of the first entry a_1 of the solution vector a for the electrostatic Potential Φ_h. It results in a heavily tailed distribution, see fig. 2.4b. In the last step we pick an arbitrary position $x_0 = \frac{L}{3}$ and evaluate the electric field for every solution $a(x_k)$ at x_0, which allows us to estimate the quantiles of the electric field estimator $\hat{E}(x_0, X)$, which directly appears in the particle push. It would be better to have now some confidence intervals, since the particle movement depends on the accuracy of the electric field, yet the distribution is again heavily tailed, see fig. 2.4c. The use of confidence intervals based on a Gaussian distribution for code verification like in [70] is then highly questionable.

2.3. Variance reduction

Estimating the fields with a Monte Carlo estimator introduces noise proportional to the variance, that can be reduced by variance reduction methods. The control variate scheme is a common variance reduction technique, which was introduced to PIC codes by [13] as the δf method and has been refined since then [15, 71]. We start with the standard example. The goal is to estimate the integral over a step function

$$f_1 : [0,1] \to [0,1], \quad x \mapsto \left\lfloor \frac{x}{m} \right\rfloor, \quad m = 8 \tag{2.137}$$

by Monte Carlo integration. We draw uniform samples $x_k,\ k = 1, \ldots, N$ of the random deviate $X \sim \mathcal{U}(0,1)$ and estimate the integral

$$\theta = \int_0^1 f_1(x)\ \mathrm{d}x = \mathbb{E}\left[f_1(X)\right] \approx \frac{1}{N_p} \sum_{k=1}^{N_p} f_1(x_k) = \hat{\theta}. \tag{2.138}$$

This situation is depicted in fig. 2.5a. We now use the additional knowledge about the integral of the linear slope $h_1 : [0,1] \to [0,1], \quad x \mapsto x - \frac{1}{2m}$, which is $\int_0^1 h_1(x)\ \mathrm{d}x = \frac{1}{2}$. By subtracting h from the estimator (2.138) and adding again the known value of the integrands we do not change the expectation (2.139).

$$\begin{aligned} \mathbb{E}[f_1(X)] &= \mathbb{E}[f_1(X) - h_1(X) + h_1(x)] \\ &= \underbrace{\mathbb{E}[f_1(X) - h_1(X)]}_{\approx \frac{1}{N_p} \sum_{k=1}^{N_p} (f_1(x_k) - h_1(x_k))} + \underbrace{\int_0^1 h_1(x)\ \mathrm{d}x}_{=\frac{1}{2}} \end{aligned} \tag{2.139}$$

Now the samples x_k merely sample the difference $\delta f := f - h$, see fig. 2.5, yielding a new estimator

$$\hat{\theta}^* := \frac{1}{N_p} \sum_{k=1}^{N_p} \left(f_1(x_k) - h_1(x_k)\right) \tag{2.140}$$

where in the expectation nothing changed.

$$\mathbb{E}\left[\hat{\theta}^*\right] = \mathbb{E}\left[\hat{\theta}\right] = \theta \tag{2.141}$$

One can show that the variance of the δf estimator $\hat{\theta}^*$ is much smaller than the variance of $\hat{\theta}$.

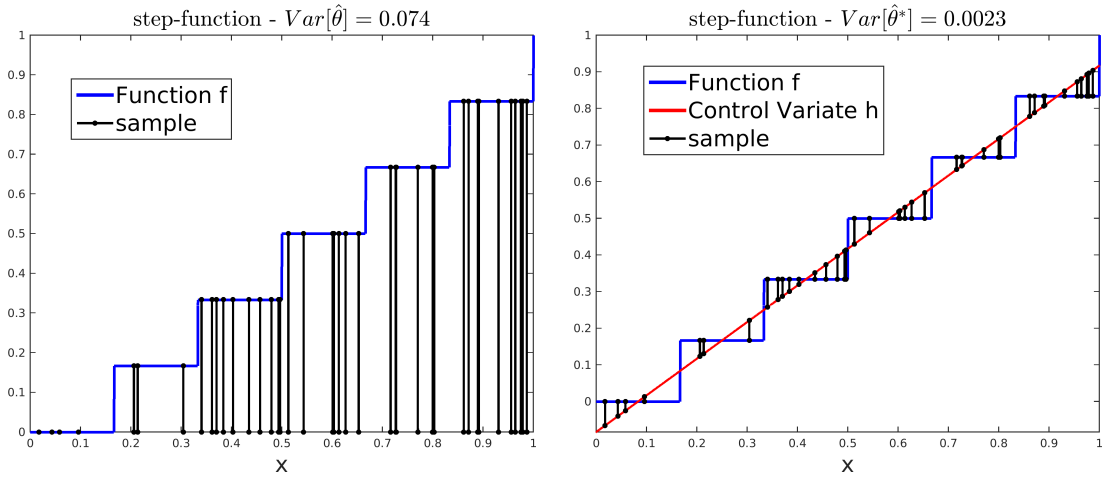
$$\mathbb{V}\left[\hat{\theta}^*\right] \mathbb{V}\left[\hat{\theta}\right] \tag{2.142}$$

$$\mathbb{E}\left[f_1(X)\right] = \int_0^1 f_1(x)\ \mathrm{d}x = \frac{m-1}{2m}, \tag{2.143}$$

$$\mathbb{V}\left[f_1(X)\right] = \int_0^1 \left(f_1(X) - \mathbb{E}[\, f_1(X)\,]\right)^2\ \mathrm{d}x \tag{2.144}$$

$$\mathbb{V}\left[f_1(X) - h_1(X)\right] = \int_0^1 \left(f_1(X) - h_1(X) - \mathbb{E}[\, f_1(X) - h_1(X)\,]\right)^2\ \mathrm{d}x \tag{2.145}$$

In the following we explore decomposition into a rather simple background h_1, a difference $\delta f_1 = f_1 - h_1$ and its application to the Vlasov–Poisson system.



(a) integrating the step function f_1 with uniformly sampled points

(b) integrating the difference between a step function f_1 and a control variate h_1 with uniformly sampled points yields a lower variance

Figure 2.5.: Estimating the integral over a step function f_1 by Monte Carlo integration.

2.3.1. δf Sampling the difference

We are interested in the first Fourier component of a small one dimensional disturbance $f_2(x) := 1 + \epsilon\ cos(2\pi x)$. The first Fourier mode is $\mathcal{F}\{f_2\}(1) = \int_0^1 \mathrm{e}^{\mathrm{i}2\pi x} f_2(x)\ \mathrm{d}x = \epsilon$. With a uniformly distributed random variable $X \sim \mathcal{U}(0,1)$ we introduce the random variable $\theta := \mathrm{e}^{\mathrm{i}2\pi X} f_2(X)$, which yields $\mathbb{E}[\theta] = \mathcal{F}\{f_2\}(1) = \frac{\epsilon}{2}$ and for the second moment

$$\mathbb{E}\left[\theta\theta^t\right] = \int_0^1 \mathrm{e}^{\mathrm{i}2\pi x - i2\pi x} f_2(x)^2\ \mathrm{d}x = \int_0^1 f_2(x)^2 = 1 + \frac{\epsilon^2}{2}. \tag{2.146}$$

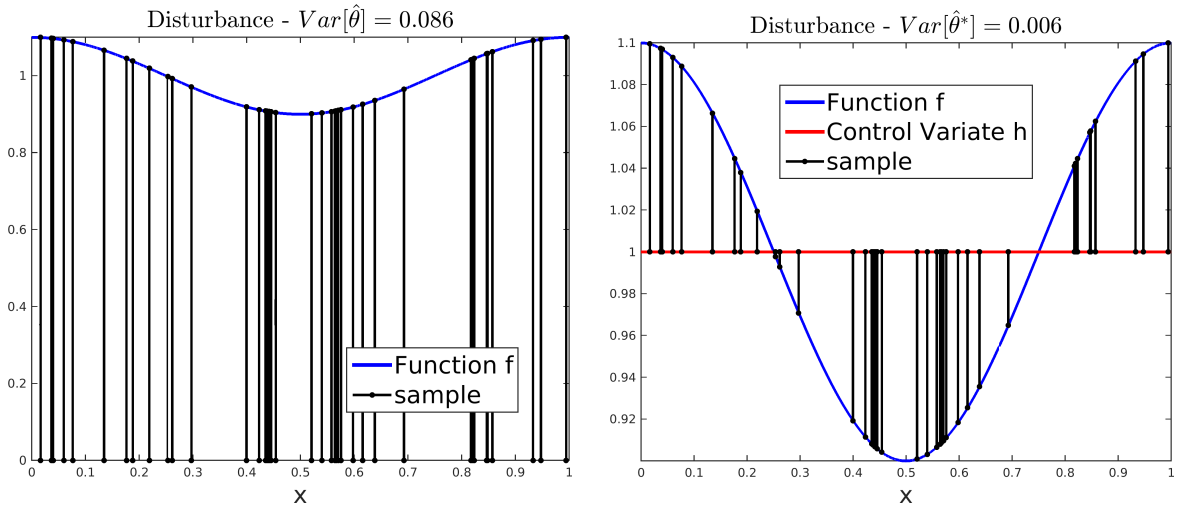
The variance is $\mathbb{V}[\theta] = \mathbb{E}\left[\theta\theta^t\right] - \mathbb{E}[\theta]\,\mathbb{E}[\theta]^t = 1 + \frac{\epsilon^2}{2} - \frac{\epsilon^2}{4} = 1 + \frac{\epsilon^2}{4}$. Let $\hat{\theta}$ be the standard Monte Carlo estimator for θ with N_p samples, which as an estimator for $\mathcal{F}\{f_2\}(1)$ is unbiased. The mean-squared-error, as the expectation of the squared ℓ^2 error, is

$$MSE[\hat{\theta}] := \mathbb{V}\left[\hat{\theta}\right] + \left(\mathbb{E}[\hat{\theta}] - \mathcal{F}\{f_2\}(1)\right)^2 = \mathbb{V}\left[\hat{\theta}\right] + \left(\frac{\epsilon}{2} - \frac{\epsilon}{2}\right)^2 = \frac{1}{N_p}\mathbb{V}[\theta] = \frac{1}{N_p}\left(1 + \frac{\epsilon^2}{4}\right). \tag{2.147}$$

For a small disturbance a relative error has to be examined, which reads here:

$$\frac{RMSE[\hat{\theta}]}{\mathcal{F}\{f_2\}(1)} = \frac{\sqrt{MSE[\hat{\theta}]}}{\mathcal{F}\{f_2\}(1)} = \frac{\sqrt{\frac{1}{N_p}\left(1 + \frac{\epsilon^2}{4}\right)}}{\epsilon} = \frac{\sqrt{1 + \frac{\epsilon^2}{4}}}{\epsilon\sqrt{N_p}} \geq \frac{\frac{1}{\epsilon}}{\sqrt{N_p}}. \tag{2.148}$$

In order to keep the relative error at the same level, the number of markers N_p has to grow quadratically with decreasing amplitude of the perturbation $N_p \sim \frac{1}{\epsilon^2}$. This behavior cannot be changed by different sampling strategies, e.g. importance sampling. In grid based integration this effect does not appear, which is a major disadvantage for particle methods. Nevertheless, this defect can be overcome by the help of a control variate. We seek to remove the leading 1 in f_2 which leads to the $\frac{1}{\epsilon}$ term in eqn. (2.148), which causes the relative error to grow with decreasing amplitude. Subtracting the zeroth Fourier mode $\mathcal{F}\{f_2\}(0) = 1$ from f_2 should solve the problem. For this we define a control variate $h_2(x) = 1$ and $\delta f_2 = f_2 - h_2$



(a) (b)

Figure 2.6.: Estimating a Fourier component of spatial disturbance.

with the corresponding random variable $\theta^* = \mathrm{e}^{\mathrm{i}2\pi X}\delta f_2(X)$. Since $\mathcal{F}\{h_2\}(1) = 0$ is known analytically

$$\mathbb{E}[\theta] = \mathcal{F}\{f_2\}(1) = \mathcal{F}\{f_2 - h_2\}(1) + \underbrace{\mathcal{F}\{h_2\}(1)}_{=0} = \mathbb{E}[\theta^*], \tag{2.149}$$

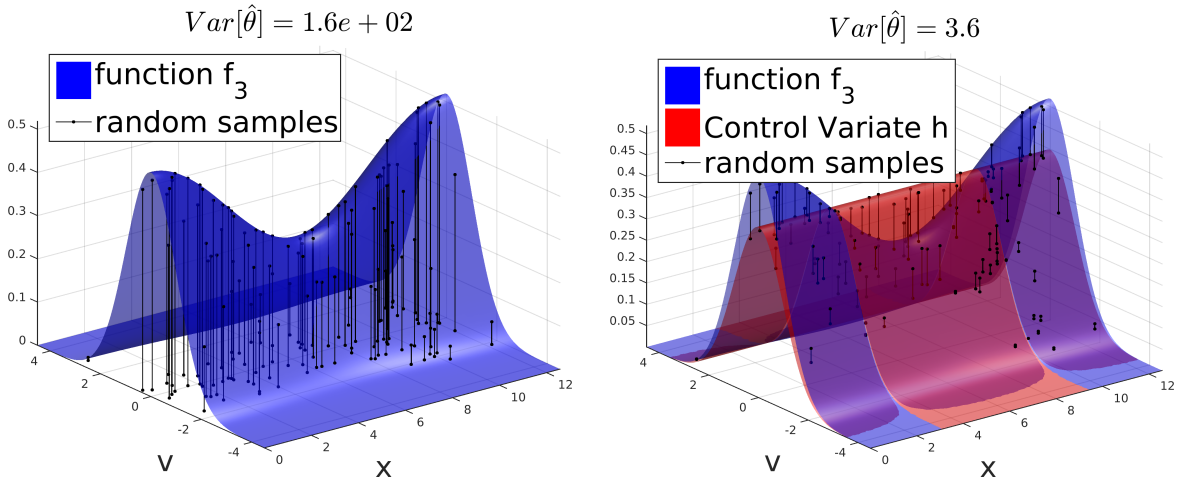
we obtain another estimator for the first Fourier mode. Calculating its variance,

$$\mathbb{V}[\theta^*] = \int_0^1 \mathrm{e}^{\mathrm{i}2\pi x - i2\pi x}\left(f_2(x) - h_2(x)\right)^2 \,\mathrm{d}x - \left(\int_0^1 \mathrm{e}^{\mathrm{i}2\pi x}\left(f_2(x) - h_2(x)\right)\,\mathrm{d}x\right)^2 = \frac{\epsilon^2}{2} - \frac{\epsilon^2}{4} = \frac{\epsilon^2}{4}, \tag{2.150}$$

and with the standard Monte Carlo estimator for $\mathbb{E}[\theta^*]$ the relative error

$$\frac{RMSE[\hat{\theta}^*]}{\mathcal{F}\{f_2\}(1)} = \frac{\sqrt{MSE[\hat{\theta}^*]}}{\mathcal{F}\{f_2\}(1)} = \frac{\sqrt{\frac{1}{N_p}\left(\frac{\epsilon^2}{4}\right)}}{\epsilon} = \frac{1}{2\sqrt{N_p}} \tag{2.151}$$

becomes independent of the amplitude ϵ. Fig. 2.5 shows $N_p = 100$ randomly distributed markers. We estimate integrals θ with $N_p = 100$ randomly distributed markers, uniformly in x and normally in v, and the standard Monte Carlo estimator $\hat{\theta}$. Introduction of a control variate h allows sampling the *difference* $\delta f = f - h$ while not changing the (phase space) position of the markers, hence it is often referred as the δf-method. Figures 2.5, 2.6 and 2.7 depict the marker positions and their weights as lines, without and with control variate. In a simulation the markers are characteristics, so we enhance the estimates on the fields, while not changing the past characteristics. We want to extend this technique to a one dimensional plasma. Consider the density $f_3(x,v) = (1 + \epsilon\cos(2\pi x))\,\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{v^2}{2}}$ consisting of a small spatial perturbation of a Maxwellian background. As we have learned from the previous example we should remove the zeroth spatial Fourier mode, which is $\int_0^1 f_3(x,v)\,dx = 1 \cdot \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{v^2}{2}}$ the Maxwellian background. Therefore, taking the control variate $h_3(x,v) = 1 \cdot \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{v^2}{2}}$ yields the same variance reduction as before. As simulation time passes by the velocity distribution will deviate from the standard Maxwellian. This is modeled by a perturbed Maxwellian velocity distribution $f_4(x,v) := (1 + \epsilon_x\cos(2\pi x))\,(1 + \epsilon_v\cos(k_v v))\,\frac{\mathrm{e}^{\frac{k_v^2}{2}}}{\sqrt{2\pi}}\mathrm{e}^{-\frac{v^2}{2}}$, $k_v = 6\pi$. Since



(a) f_3 (b)

Figure 2.7.: Estimating a Fourier component of spatial disturbance under a Maxwellian velocity background.

the density still factorizes the standard Maxwellian is a good control variate.

$$\int f_4(x,v) - h_3(x,v)\ dv = \epsilon_x \cos(2\pi x) \tag{2.152}$$

When solving a kinetic equation, a certain moment $\theta(t) = \iint \psi(x,v) f(x,v,t)\ \mathrm{d}x\mathrm{d}v$ shall be estimated by the use of a control variate. Instead of plugging in the particle discretization f_p, we define a new particle discretization f_p^*. Here the control variate (background) is subtracted on particle level and then added analytically yielding an unbiased estimate.

$$f_p^*(x,v,t) := f_p(x,v,t) - \alpha \frac{1}{N_p} \sum_{k=1}^{N_p} \underbrace{\frac{h(x_k(t),v_k(t))}{g(t,x_k(t),v_k(t))}}_{:=\gamma_k(t)} \delta\left(x - x_k(t)\right) \delta\left(v - v_k(t)\right) \ + \ \alpha h(x,v) \tag{2.153}$$

In equation (2.153) α denotes the optimization coefficient for the control variate and is set to $\alpha = 1$ if not specified otherwise. Since g is constant along the characteristics the control variate weights reduce to

$$\gamma_k(t) := \frac{h(x_k(t),v_k(t),t)}{g(x_k(t),v_k(t),t)} = \frac{h\left(x_k(t),v_k(t),t\right)}{g\left(x_k(0),v_k(0),t=0\right)} = \frac{h(x_k(t),v_k(t),t)}{g_0^k}. \tag{2.154}$$

To simplify eqn. (2.153) we define the δf weights as

$$\begin{aligned} \delta w_k :=& \frac{f(x_k(t),v_k(t),t) - \alpha h(x_k(t),v_k(t),t)}{g(x_k(t),v_k(t),t)} \\ =& w_k - \alpha \frac{h(x_k(t),v_k(t),t)}{g(x_k(t),v_k(t),t)} \\ =& w_k - \gamma_k(t). \end{aligned} \tag{2.155}$$

This allows us to rewrite the control variate particle discretization f_p^* as

$$f_p^*(x,v,t) = \frac{1}{N_p} \sum_{k=1}^{N_p} \underbrace{\left(w_k - \alpha\gamma_k(t)\right)}_{\delta w_k} \delta\left(x - x_k(t)\right) \delta\left(v - v_k(t)\right) \ + \ \alpha h(x,v). \tag{2.156}$$

Define the control variate estimator $\hat{\theta}^*$ of θ.

$$\begin{aligned}\hat{\theta}^*(t) :=& \iint \psi(x,v) f_p^*(x,v,t)\ \mathrm{d}x\mathrm{d}v \\ =& \frac{1}{N_p}\sum_{k=1}^{N_p} \left(w_k - \alpha\gamma_k(t)\right)\psi(x_k(t),v_k(t)) \ + \ \alpha \iint \psi(x,v)h(x,v)\ dxdv \\ =& \hat{\theta} - \alpha \underbrace{\frac{1}{N_p}\sum_{k=1}^{N_p}\gamma_k(t)\psi(x_k(t),v_k(t))}_{:=\hat{\eta}} \ + \ \alpha \underbrace{\iint \psi(x,v)h(x,v)\ dxdv}_{:=\eta}\end{aligned} \tag{2.157}$$

Here, $\hat{\eta}$ is as a standard Monte Carlo estimator unbiased since $\mathbb{E}\left[\hat{\eta}\right] = \eta$. Therefore also $\hat{\theta}^*$ is unbiased.

$$\begin{aligned}\mathbb{E}\left[\hat{\theta}^*\right] =& \mathbb{E}\left[\hat{\theta} - \alpha\hat{\eta}\right] + \alpha\eta \\ =& \mathbb{E}\left[\hat{\theta}\right] - \alpha\mathbb{E}\left[\hat{\eta}\right] + \alpha\eta = \mathbb{E}\left[\hat{\theta}\right] - \eta + \alpha\eta \\ =& \mathbb{E}\left[\hat{\theta}\right] = \theta\end{aligned} \tag{2.158}$$

This means we did not change the expectation, in particular not the phase space positions of the markers.

$$\mathbb{V}\left[\hat{\theta}^*\right] = \mathbb{V}\left[\hat{\theta} - \alpha\hat{\eta} \ + \ \underbrace{\alpha\eta}_{\text{constant}}\right] = \mathbb{V}\left[\hat{\theta} - \alpha\hat{\eta}\right] = \mathbb{V}\left[\hat{\theta}\right] - 2\alpha\mathrm{COV}\left[\hat{\theta},\hat{\eta}\right] + \alpha^2\mathbb{V}\left[\hat{\eta}\right] \tag{2.159}$$

In this case we want the variance of the new estimator $\hat{\theta}^*$ to be smaller than the original one. With the free parameter α we set up a simple optimization problem to minimize the variance

$$\min_{\alpha\in\mathbb{R}} \quad \mathbb{V}\left[\hat{\theta}\right] - 2\alpha\mathrm{COV}\left[\hat{\theta},\hat{\eta}\right] + \alpha^2\mathbb{V}\left[\hat{\eta}\right]. \tag{2.160}$$

The solution to this quadratic problem is known to be

$$\alpha := \frac{\mathrm{COV}\left[\hat{\theta},\hat{\eta}\right]}{\mathbb{V}\left[\hat{\eta}\right]}. \tag{2.161}$$

In the case were the control variate h has a strong correlation to the density f, we obtain good variance reduction and α will tend to $\alpha = 1$. Since estimating the covariance, and variances in (2.161) yields additional work and uncertainties, we mostly directly choose $\alpha = 1$. Suppose α is known exactly then the variance of the new estimator $\hat{\theta}^*$ can be calculated and therefore, we also know the amount of variance reduction. This directly corresponds to required number of particles.

$$\mathbb{V}\left[\hat{\theta}^*\right] = \mathbb{V}\left[\hat{\theta}\right] - \frac{\mathrm{COV}\left[\hat{\theta},\hat{\eta}\right]^2}{\mathbb{V}\left[\hat{\eta}\right]} = \mathbb{V}\left[\hat{\theta}\right]\left(1 - \frac{\mathrm{COV}\left[\hat{\theta},\hat{\eta}\right]^2}{\mathbb{V}\left[\hat{\theta}\right]\mathbb{V}\left[\hat{\eta}\right]}\right) \tag{2.162}$$

Since α stems from a quadratic minimization problem (variance reduction), it is rather forgiving for small errors [72]. Nevertheless, we can estimate α for every moment θ depending on which moment ψ shall be calculated.

$$\hat{\alpha} = \frac{\frac{1}{N_p-1}\sum_{k=1}^{N_p}\left(w_k\psi\left(x_k(t),v_k(t)\right) - \hat{\theta}\right)\left(\gamma_k(t)\psi\left(x_k(t),v_k(t)\right) - \hat{\eta}\right)}{\frac{1}{N_p-1}\sum_{k=1}^{N_p}\left(\gamma_k(t)\psi\left(x_k(t),v_k(t)\right) - \hat{\eta}\right)^2} \tag{2.163}$$

But using the estimator (2.163) cannot always guarantee variance reduction, because it is an estimator. Hence, below a certain threshold it is advised to set $\alpha = 0$. When using quasi Monte Carlo sampling [72] covariances are not straightforward to estimate, which limits the usage of (2.163).

2.3.2. Stochastic optimization

It has been tried to improve upon the standard Maxwellian control variate by defining a local Maxwellian [71] or the use of separate control variate for every estimator [68]. When using a control variate, the optimization coefficient is calculated in order to have an optimal variance reduction. Since the optimization coefficient is mostly unknown, we have to rely on an estimate.
Let $Z = (X, V)$ and $\psi(Z)$ be a moment we want to calculate, for example $\psi(X, V) = X^3$. In the standard setting we have already a candidate h for a control variate and we seek a variance reduction by calculating the right correlation coefficient. We are interested in the integral

$$\mathbb{E}\left[\left(\frac{f(Z)}{g(Z)}\right)\psi(Z)\right] = \mathbb{E}\left[\left(\frac{f(Z)-\alpha h(Z)}{g(Z)}\right)\psi(Z)\right] = \iint f(z)\psi(z)\mathrm{d}z. \tag{2.164}$$

This leaves us with the following optimization problem

$$\min_{\alpha\in\mathbb{R}} \mathbb{V}\left[\left(\frac{f(Z)-\alpha h(Z)}{g(Z)}\right)\psi(Z)\right] \tag{2.165}$$

Since this is a quadratic problem, we can solve this analytically by calculating the roots of the gradient for $\alpha \in \mathbb{R}$.

$$\begin{aligned} F(\alpha) &:= \mathbb{V}\left[\left(\frac{f(Z)-\alpha h(Z)}{g(Z)}\right)\psi(Z)\right] \\ &= \mathbb{COV}\left[\left(\frac{f(Z)-\alpha h(Z)}{g(Z)}\right)\psi(Z), \left(\frac{f(Z)-\alpha h(Z)}{g(Z)}\right)\psi(Z)\right] \end{aligned} \tag{2.166}$$

The variance can be rewritten as a covariance (bilinear form), which simplifies further calculations and the final implementation.

$$\begin{aligned} F(\alpha) &= \mathbb{V}\left[\frac{f(Z)}{g(z)}\psi(Z)\right] + 2\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), -\frac{\alpha h(Z)}{g(Z)}\psi(Z)\right] + \mathbb{V}\left[\frac{-\alpha h(Z)}{g(z)}\psi(Z)\right] \\ &= \mathbb{V}\left[\frac{f(Z)}{g(z)}\psi(Z)\right] - 2\alpha\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right] + \alpha^2\mathbb{V}\left[\frac{h(Z)}{g(z)}\psi(Z)\right] \end{aligned} \tag{2.167}$$

$$\begin{aligned} \frac{d}{d\alpha}F(\alpha) &= 2\mathbb{COV}\left[\left(\frac{f(Z)-\alpha h(Z)}{g(Z)}\right)\psi(Z), -\frac{h(Z)}{g(Z)}\psi(Z)\right] \\ &= -2\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right] + 2\alpha\mathbb{COV}\left[\frac{h(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right] \\ &= -2\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right] + 2\alpha\mathbb{V}\left[\frac{h(Z)}{g(Z)}\psi(Z)\right] \end{aligned} \tag{2.168}$$

To find the minimum of the quadratic function, we set the first derivative $\frac{d}{d\alpha}F(\alpha)$ to zero, which yields a solution for α.

$$\frac{d}{d\alpha}F(\alpha) = 0 \Leftrightarrow \alpha = \frac{\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right]}{\mathbb{V}\left[\frac{h(Z)}{g(Z)}\psi(Z)\right]} \tag{2.169}$$

To see the variance reduction, we insert α back into the functional.

$$\begin{aligned} F\left(\frac{\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right]}{\mathbb{V}\left[\frac{h(Z)}{g(Z)}\psi(Z)\right]}\right) &= \mathbb{V}\left[\frac{f(Z)}{g(z)}\psi(Z)\right] \\ &\quad - 2\frac{\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right]^2}{\mathbb{V}\left[\frac{h(Z)}{g(Z)}\psi(Z)\right]} + \frac{\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right]^2}{\mathbb{V}\left[\frac{h(Z)}{g(Z)}\psi(Z)\right]} \\ &= \mathbb{V}\left[\frac{f(Z)}{g(z)}\psi(Z)\right]\left(1 - \underbrace{\frac{\mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{h(Z)}{g(Z)}\psi(Z)\right]^2}{\mathbb{V}\left[f(Z)\frac{\psi(Z)}{g(z)}\right]\mathbb{V}\left[h(Z)\frac{\psi(Z)}{g(Z)}\right]}}_{:=\varrho^2}\right) \end{aligned} \tag{2.170}$$

We see that the variance of the original estimator is reduced by a factor $(1-\varrho^2)$, where we call ϱ^2 the correlation coefficient.

Parameterized control variate

In general, we treat a control variate $h(x, v, \alpha)$, which depends on parameters α.

$$h(x, v, \alpha) := \alpha_1 \frac{1}{\alpha_3\sqrt{2\pi}} \mathrm{e}^{-\frac{(v-\alpha_2)^2}{\alpha_3^2}} \tag{2.171}$$

We want to minimize the function F over α.

$$\begin{aligned} F(\alpha) &:= \mathbb{V}\left[\left(\frac{f(Z) - h(Z,\alpha)}{g(Z)}\right)\psi(Z)\right] \\ &= \mathbb{COV}\left[\left(\frac{f(Z) - h(Z,\alpha)}{g(Z)}\right)\psi(Z), \left(\frac{f(Z) - h(Z,\alpha)}{g(Z)}\right)\psi(Z)\right] \end{aligned} \tag{2.172}$$

The gradient,

$$\nabla_\alpha F = -2\mathbb{COV}\left[\left(\frac{f(Z) - h(Z,\alpha)}{g(Z)}\right)\psi(Z), \nabla_\alpha h(Z,\alpha)\frac{\psi(Z)}{g(Z)}\right] \tag{2.173}$$

and the Hessian are given

$$\begin{aligned} \nabla_\alpha^2 F = +2\mathbb{COV}&\left[\nabla_\alpha h(Z,\alpha)\frac{\psi(Z)}{g(Z)}, \nabla_\alpha h(Z,\alpha)\frac{\psi(Z)}{g(Z)}\right] \\ &- 2\mathbb{COV}\left[\left(\frac{f(Z) - h(Z,\alpha)}{g(Z)}\right)\psi(Z), \nabla_\alpha^2 h(Z,\alpha)\frac{\psi(Z)}{g(Z)}\right]. \end{aligned} \tag{2.174}$$

This allows us to employ a Newton method. But since the quantities and the derivatives have to be estimated and have a certain stochastic error, the Newton or gradient method becomes inexact. Nevertheless, using the standard Monte Carlo estimator for $F, \nabla_\alpha F$ and $\nabla_\alpha \alpha^2 F$ yields unbiased estimates. There is a broad field of research for converging algorithms using the unbiased estimators, starting with the stochastic gradient descent and of course Newton and Quasi-Newton methods, see [73, 74].

Kernel density estimation

In case the velocity background $\int_{\Omega_x} f(x,v,t)\ \mathrm{d}x$ cannot be described by a Maxwellian, a general approach is to use a kernel density estimate. For a smoothing kernel [51] $K : \mathbb{R} \to \mathbb{R}$, and a smoothing window σ_v we define the convolution

$$\begin{aligned} f_0(v,t) &:= \int_{\Omega_x} f(x,\tau,t) K\left(\frac{v-\tau}{\sigma_v}\right) \frac{1}{\sigma_v}\ \mathrm{d}x\mathrm{d}\tau \\ &= \frac{1}{\sigma_v}\mathbb{E}\left[K\left(\frac{v-V(t)}{\sigma_v}\right) W(t)\right] \end{aligned} \tag{2.175}$$

yielding an estimator

$$\hat{f}_0(v,t) := \frac{1}{\sigma_v}\frac{1}{N_p}\sum_{k=1}^{N_p} K\left(\frac{v-v_k^t}{\sigma_v}\right) w_k^t. \tag{2.176}$$

Since equation (2.176) is costly, it cannot be applied at every time step, yet the background is not subject to much fluctuation. A suitable control variate is then $h(x,v,t) := \hat{f}_0(v,t)$. In practice an intermediate layer of interpolation is introduced. For a broad enough grid in velocity space $(\bar{v}_n)_{n=1,\dots,N_v}$ the estimator is evaluated at the grid points $\hat{f}_0(\bar{v}_n,t)$ and then subject to cubic spline interpolation.
For six-dimensional simulations, a three-dimensional grid is still computationally feasible, as we use one for the charge density anyhow. It should also be noted that the grid can be quite coarse.

Gauss–Hermite interpolation

Distributions close to a Maxwellian play a dominant role in plasma physics as they form equilibrium states of the Vlasov equation. In a collisionless plasma there are actually many others, but this shall not concern us here. We try to enhance the standard Maxwellian control variate by allowing additional perturbations. The Hermite polynomials H_n form an orthogonal basis of $L^2(\mathbb{R},w)$, where the weight function is defined as a Gaussian $w(x) = \mathrm{e}^{-x^2}$. This allows for an unbounded velocity space discretization, which enables us to reconstruct the velocity density without additional discretization error. It is also used in spectral Vlasov solvers, where only very few polynomials are needed [75]. Here we construct an orthogonal series estimator for the moment density $f(v)$. For detailed derivation, description and stochastic analysis of the method, especially concerning MSE estimates, we refer to [76] and [77]. We recall some properties of the Hermite polynomials, see [78][p. 250]. They are obtained by the recursion formula given in eqn. (2.177).

$$H_n(x) = \sum_{k=0}^{n} a_{n,k} x^k \tag{2.177}$$

$$a_{0,0} = 1, \quad a_{1,0} = 0, \quad a_{1,1} = 2 \tag{2.178}$$

$$a_{n+1,0} = -n 2 a_{n-1,0} \tag{2.179}$$

$$a_{n+1,k} = 2a_{n,k-1} - 2n a_{n-1,k} \quad \text{for } k \geq 0 \tag{2.180}$$

The Hermite polynomials are orthogonal with respect to the Gaussian weight function w.

$$\int_{-\infty}^{\infty} H_n(x) H_m(x) w(x) \mathrm{d}x = 0 \qquad \text{for all } n \neq m \tag{2.181}$$

$$\int_{-\infty}^{\infty} H_n(x)^2 w(x) \mathrm{d}x = 2^n \sqrt{\pi}\ n! \tag{2.182}$$

The definition of the Hermite functions $\varphi_n(x) := H_n(x)\sqrt{w(x)}$ then yields an orthogonal basis of $L^2(\mathbb{R})$. The mass is the sum over all coefficients $a_n, 0$.

$$\int_{-\infty}^{\infty} \varphi_n(x)\ \mathrm{d}x = \int_{-\infty}^{\infty} H_n(x)\mathrm{e}^{-\frac{x^2}{2}}\ \mathrm{d}x = \sqrt{2\pi}|a_{n,0}| \tag{2.183}$$

When weighting data by the Gaussian weight function w calculations can be done directly with the Hermite polynomials. The weight $\sqrt(w(x)) = \mathrm{e}^{-\frac{1}{2}x^2}$ corresponds already to an unnormalized standard Maxwellian. In order to use this orthogonal series of φ_n for a suitable density estimate for a function $f(v)$, we define a centralizing and normalizing coordinate transformation.

$$\begin{aligned} \tilde{v} &= \frac{(v-\mu_v)}{\sigma_v} \\ \mu_v = \int_{-\infty}^{\infty} v f(v)\ \mathrm{d}v \quad &\text{and} \quad \sigma_v = \sqrt{\int_{-\infty}^{\infty} (v-\mu_v)^2 f(v)\ \mathrm{d}v} \end{aligned} \tag{2.184}$$

Note the additional normalizing factor by the coordinate transformation $\frac{\mathrm{d}\tilde{v}}{\mathrm{d}v} = \sigma_v \Rightarrow\ \mathrm{d}v = \frac{\mathrm{d}\tilde{v}}{\sigma_v}$. The truncated density estimator for f then reads

$$\hat{f}(v) = \sum_{n}^{N} c_n H_n(\tilde{v}) w(\tilde{v}) \tag{2.185}$$

$$c_n = \int_{-\infty}^{\infty} H_n(\tilde{v})\sqrt{w(\tilde{v})} f(v)\mathrm{d}v \frac{1}{\sigma_v 2^n \sqrt{\pi}\ n!}. \tag{2.186}$$

The c_n are the coefficients for the linear combination of Hermite function. The total mass is then given by

$$\int \hat{f}(v)\ \mathrm{d}v = \sum_{n=0}^{N} c_n |a_{n,0}| \frac{\sqrt{2\pi}}{\sigma_v}. \tag{2.187}$$

The extension of this single dimensional estimator to a control variate $h(x, v)$ is done as follows, where we normalize the weights by the total mass in order to approximate the local Maxwellian.

$$m = \mathbb{E}\left[\frac{f(X,V,t)}{g(X,V,t)}\right] \tag{2.188}$$

$$\mu_v = \mathbb{E}\left[V\frac{f(X,V,t)}{g(X,V,t)}\right]\frac{1}{m} \tag{2.189}$$

$$\sigma_v = \mathbb{E}\left[(V-\mu_v)^2\frac{f(X,V,t)}{g(X,V,t)}\right]\frac{1}{m} \tag{2.190}$$

$$c_n = \mathbb{E}\left[H_n\left(\frac{V-\mu_v}{\sigma_v}\right)\mathrm{e}^{-\frac{1}{2}\left(\frac{V-\mu_v}{\sigma_v}\right)^2}\frac{f(X,V,t)}{g(X,V,t)}\right]\frac{1}{L}\frac{1}{\sigma_v 2^n\sqrt{\pi}\ n!} \tag{2.191}$$

Another option is to use very few particles and perform a linear least square interpolation problem using the f_k in order to determine the coefficients c_n.

Perfect and multiple control variates

We presented several possible control variates, either for the δf approach, or directly to retain some conservation properties in the method itself. If one can improve upon the standard control variates and find a perfect control variate h^* as a solution to the variance minimization problem eqn. (2.192) then higher rate of convergence such as $\frac{1}{N_p}$ is possible, see [79] and [80].

$$0 = \min_{h\in L^2} \mathbb{V}\left[\frac{f(Z)-h(Z)}{g(Z)}\psi(Z)\right] \tag{2.192}$$

We are already satisfied with something working for nonlinearly perturbed densities f. Here the perfect h^* is f, which we do not have, but we can try to approximate it. Any approximation can start out as a linear combination of basis functions, see eqn. (2.193). This is the same Ansatz for multiple control variates, so we can treat this case too by considering every basis function φ to be a control variate or vice versa.

$$h(z,\alpha) = \alpha^t \varphi(z) := \sum_{n=1}^{N} \alpha_n \varphi_n(z) \tag{2.193}$$

The cost function for the linear combination of control variates is then given in eqn. (2.194).

$$J(\alpha) = \mathbb{V}\left[\frac{f(Z) - h(Z,\alpha)}{g(Z)}\psi(Z)\right] = \mathbb{V}\left[\frac{f(Z) - \sum_{n=1}^{N}\alpha_n\varphi_n(z)}{g(Z)}\psi(Z)\right] \tag{2.194}$$

The solution to the minimization problem $\min_\alpha J(\alpha)$ can be directly computed and is given in eqn. (2.195). With the discrete estimators for mean, variance and covariance eqn. (2.195) then solves also the discrete problem, which enforces variance reduction. Yet this involves the costly assembly and inversion of the matrix $\Sigma = A - bb^t$. In the case of an ill-conditioned covariance matrix Σ removal of the null space is a feasible solution. This just tells the user that there are too many control variates present.

$$\begin{aligned}
\alpha &= \mathbb{COV}\left[\frac{\psi(Z)}{g(Z)}f(Z), \frac{\psi(Z)}{g(Z)}\varphi(Z)\right]\left\{\mathbb{COV}\left[\frac{\psi(Z)}{g(Z)}\varphi(Z), \frac{\psi(Z)}{g(Z)}\varphi(Z)^t\right]\right\}^{-1} \\
&= \left\{\mathbb{E}\left[\left(\frac{\psi(Z)}{g(Z)}\right)^2 f(Z)\varphi(Z)\right] - \mathbb{E}\left[\frac{\psi(Z)}{g(Z)}f(Z)\right]\mathbb{E}\left[\frac{\psi(Z)}{g(Z)}\varphi(Z)\right]\right\}\cdot \\
&\quad \left\{\underbrace{\mathbb{E}\left[\left(\frac{\psi(Z)}{g(Z)}\right)^2 \varphi(Z)\varphi(Z)^t\right]}_{:=A} - \underbrace{\mathbb{E}\left[\frac{\psi(Z)}{g(Z)}\varphi(Z)\right]}_{:=b}\mathbb{E}\left[\frac{\psi(Z)}{g(Z)}\varphi(Z)\right]^t\right\}^{-1}
\end{aligned} \tag{2.195}$$

Note that b can be determined analytically for every test function ψ but not A, since it depends on the sampling distribution which for the Vlasov is unknown for later times. Because of the conservation of phase space volume a constant sampling density will stay constant, except that the boundary of $\mathrm{supp}(g(t)) = z|g(z,t) \neq 0$ support is subject to change. Yet if the initial domain is large enough, one may assume that nothing happens at the boundary. So for $g(z) = C$ the matrix A constitutes the mass matrix of standard Galerkin L^2 projection with test functions $\varphi \cdot \psi$, see eqn. (2.196).

$$b = \mathbb{E}\left[\frac{\psi(Z)}{g(Z)}\varphi(Z)\right] = \int_\Omega \psi(z)\varphi(z)\ \mathrm{d}z \tag{2.196}$$

$$A_{i,j} = \int_\Omega \frac{\psi(z)^2}{g(z)}\varphi_i(z)\varphi_j(z)\ \mathrm{d}z \tag{2.197}$$

$$c_j := \mathbb{COV}\left[\frac{\psi(Z)}{g(Z)}f(Z), \frac{\psi(Z)}{g(Z)}\varphi_j(Z)\right] \text{ for } j = 1,\dots,N \tag{2.198}$$

In order to point out the link between multiple control variates and the Galerkin discretization, the minimization problem in eqn. (2.192) is rewritten in variational form, see

eqn. (2.199). Since the covariance is a bilinear form, this is straightforward.

$$a(u,v) := \mathbb{COV}\left[\frac{u(Z)}{g(Z)}\psi(Z), \frac{v(Z)}{g(Z)}\psi(Z)\right] \tag{2.199}$$

$$L(v) := \mathbb{COV}\left[\frac{f(Z)}{g(Z)}\psi(Z), \frac{v(Z)}{g(Z)}\psi(Z)\right] \tag{2.200}$$

$$\min_{h\in L^2} a(f-h, f-h) \tag{2.201}$$

$$a(h, \delta h) = L(f) \tag{2.202}$$

$$a(\alpha_i\varphi_i, \varphi_j) = L(\varphi_j) \text{ for all } i,j = 1,\dots,N \tag{2.203}$$

The covariance defines a centered positive semi-definite bilinear form by subtracting the respective mean:

$$a(u,v) = \mathbb{COV}\left[\frac{u(Z)}{g(Z)}, \frac{v(Z)}{g(Z)}\right] = \int_\Omega \left(u(z) - \int_\Omega u(z')\mathrm{d}z'\right)\left(v(z) - \int_\Omega v(z')\mathrm{d}z'\right)^t \mathrm{d}z \tag{2.204}$$

On the space of all zero mean random deviates the covariance forms then an inner product. Removing the centering mean from eqn. (2.204) yields the positive definite L^2 scalar product used for the L^2-Galerkin approximation in eqn. (2.205).

$$a(u,v) = \mathbb{E}\left[\frac{u(Z)v(Z)}{g(Z)}\right] = \int_\Omega u(z)v(z)\ \mathrm{d}z \tag{2.205}$$

Independently of the bilinear form a the best approximation in the respective norm $\|u\| = \sqrt{a(u,u)}$ is given by Céas Lemma up to a constant. Here eqn. (2.199) incorporates knowledge about the sampling density whereas eqn. (2.205) completely neglects g, which is closer to eqn. (2.199) when g is constant and we sample uniformly. So choosing f as the best approximation to h under some discretization in $\mathcal{L}^2$ with the standard scalar product will give the best results when g is uniform. Apart from the centralization in eqn. (2.199) there is also the test function ψ. It would be favorable to find a control variate such that it is optimal for all test functions ψ. In a periodic domain we are the most interested in the Fourier modes of the electric field. These modes are damped by the mode number k and in order to include all of them at once they are accumulated. For $x \in [0, L]$ define $\theta = 2\pi\frac{x}{L}$ and notice that

$$\sum_{k=1}^{\infty} \frac{\cos(k\theta)}{k} = \sum_{k=1}^{\infty} -\frac{1}{2}\ln\left(2 - 2\cos(\theta)\right), \quad \sum_{k=1}^{\infty} \frac{\sin(k\theta)}{k} = \frac{\pi-\theta}{2}. \tag{2.206}$$

Since only the positive mode numbers k are needed a general test function ψ can be composed as

$$\psi(z) = \psi\left((x,v)\right) = -\frac{1}{2}\ln\left(2 - 2\cos(\theta)\right) + \mathrm{i}\frac{\pi-\theta}{2}. \tag{2.207}$$

The test function ψ can be extracted by conditioning. Where we minimize only the right term in eqn. (2.208) by $h = \int_\Omega f(z)\mathrm{d}z$. This corresponds to the mass and proved to be ineffective, thus we learn that the variance is located in the second term in the right hand side of eqn. (2.208).

$$\mathbb{V}\left[\frac{f(Z)-h(Z)}{g(Z)}\psi(Z)\right] = \mathbb{V}\left[\frac{f(Z)-h(Z)}{g(Z)}\psi(Z)\middle|\psi(Z)\right] + \mathbb{V}\left[\psi(Z)\middle|\frac{f(Z)-h(Z)}{g(Z)}\right] \tag{2.208}$$

In order to receive an efficient control variate either choose ψ as something similar to eqn. (2.207) or as the actual basis functions of the Poisson solver.

2.3.3. Randomized Quasi Monte Carlo (RQMC)

Using Quasi Monte Carlo (QMC) instead of standard Monte Carlo yields more uniformly distributed particles which improves the simulation drastically. Yet the measure of error for integration with these low discrepancy sequences is the Hardy-Krause variation, see [72] for an overview. Thus when using the standard Monte Carlo variance estimator and QMC numbers, merely the variance of a corresponding random sample is estimated.
Therefore Randomized Quasi Monte Carlo (RQMC) was introduced, so one could gain an error estimate with the standard Monte Carlo variance estimator [81, 82]. The idea is to sample the variance for several independent sets of RQMC numbers. So we divide the ensemble of N_p markers into R sets of N_p/R markers, measure the variance of each and take the mean over the R samples. The number of subsets can be mostly chosen very small $R = 5$. This is particularly interesting for parallelization schemes using domain cloning [83], where R can be greater than number of domains, thus reducing overhead.
Estimating the optimality coefficient for the control variate is then not straightforward and it is even unclear how great the impact of the control variate is, see [72]. High order scrambling by Dick, see [84, 85], leads to convergence rates up to $\frac{7}{2}$ but requires smooth integrands. The Vlasov density f does not exhibit this smoothness such that no improvements can be made in a nonlinear PIC simulation.

2.3.4. Example: Two-stream instability

Two electron beams [63][p. 136], provide a non-Maxwellian background and a good example to test our different sampling techniques, along with the control variate method. The initial parameters for the simulation are

$$f(x, v, t = 0) := (1 - \epsilon \cos(kx)) \frac{1}{\sqrt{2\pi}} v^2 \mathrm{e}^{-\frac{v^2}{2}} \tag{2.209}$$

$$L = \frac{2\pi}{k}, k = 0.5, \epsilon = 0.05, \frac{q}{m} = -1, \Delta t = 0.05, \ \textit{rk3s}\ , N_{\mathrm{fem}} = 32, \ \textit{cubic} \tag{2.210}$$

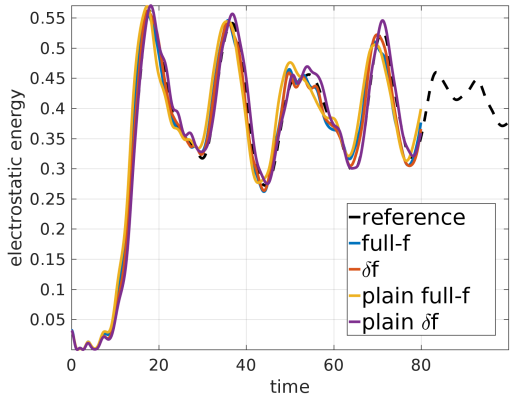
which in the nonlinear phase yields in a vortex between the two beams. First we study different sampling techniques and their impact on the control variate method. Later adaptivity of the before introduced control variates is demonstrated.

Plain sampling

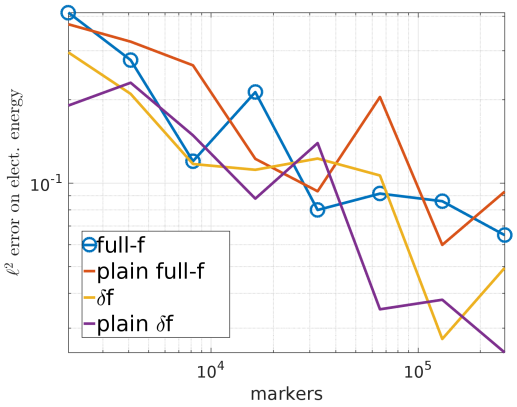
As the initial perturbation is small variance reduction is needed, and since a standard Maxwellian does not apply as a control variate for the two beams, instead the initial value is taken $h(x, v) := f(x, v, t = 0)$. Then we can also discriminate between different sampling options. The *standard* is to sample directly from f such that the markers are actually normally distributed in velocity space. In *plain sampling* the markers are uniformly distributed in the velocity space for $-v_{\max} \leq v \leq v_{\max}$ with $v_{\max} = 5$. The corresponding sampling distributions read

$$g(x, v, t = 0) = \begin{cases} \frac{1}{L} f(x, v, t = 0) & \text{standard} \\ \frac{1}{L} (1 - \epsilon \cos(kx)) \frac{1}{2v_{\max}} \mathbb{1}_{\{v \,:\, |v| \leq v_{\max}\}} & \text{plain} \end{cases} . \tag{2.211}$$

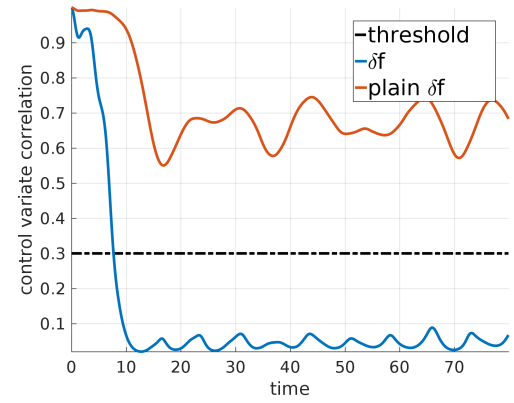
We compare the δf and full-f method for the two sampling variants. The threshold for the δf method is set to 0.3, where it becomes the standard full-f. Since fig. 2.8a does not clearly indicate a superior method, we perform a convergence study varying the number of particles $N_p = 2^{11}, \ldots, 2^{18}$. Figure 2.8b indicates superiority of the δf method even for the nonlinear



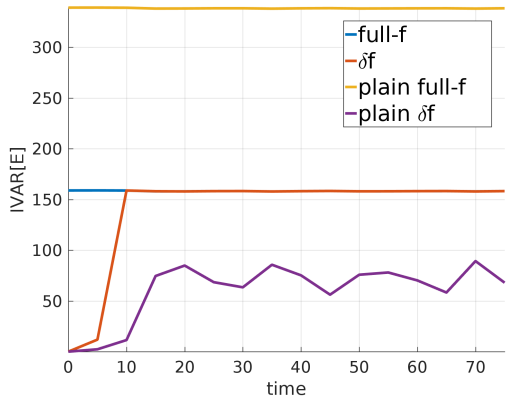
(a) Electrostatic energy for $N_p = 2^{18} = 262144$ markers.



(b) ℓ^2 error of the electrostatic energy using reference solution for $t > 20$ excluding the linear phase.



(c) Plain sampling improves efficiency of the control variate ($N_p = 2^{18}$).



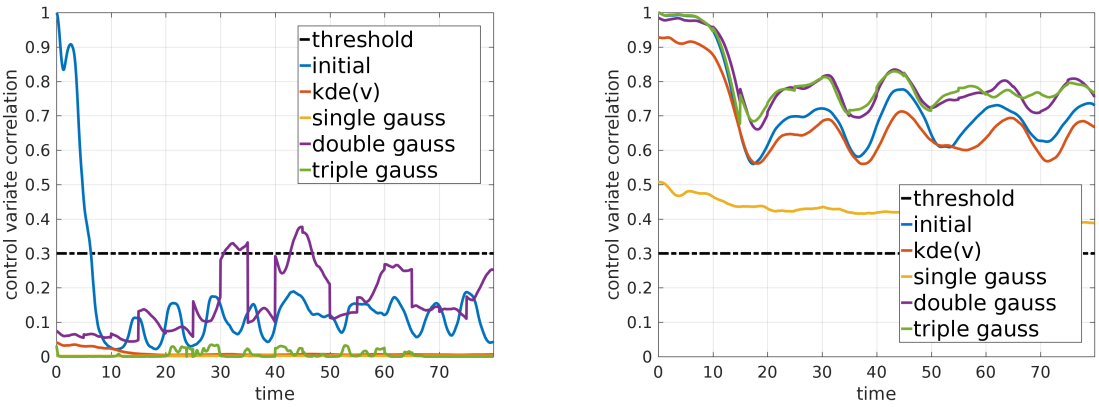
(d) The integrated variance shows *plain* δf as best pick ($N_p = 2^{18}$).

Figure 2.8.: Comparing different sampling options of the Maxwellian for the two stream instability and the effects on the control variate.

phase, where the plain sampling has clearly a negative impact on the full-f method. We look at the effectiveness of the control variate in fig. 2.8c. With the *standard* sampling the control variate lacks impact in the nonlinear phase, and δf becomes full-f. But for *plain* sampling we consider the variance reduction efficient enough throughout the nonlinear phase. In absolute numbers the integrated variance in fig. 2.8d summarizes for this case, that plain δf is the best method, *standard* δf turns to *standard* full-f and *plain* sampling is not advised in the full-f case.

Adaptive control variates

Is it possible to improve over the initial value as control variate?
We consider the initial value, a kernel density estimation in velocity space and and multiple Maxwellians/Gaussians. The parameters are kept as before and the number of particles is set to $N_p = 2 \cdot 10^4$. The Maxwellians are adapted by stochastic optimization to yield maximal variance reductions. We use the MATLABs built in *BFGS* method *fminunc* circumventing expensive Hessian evaluations. The optimization frequency is set to $f_{opt} = 1$ yielding calculations every 20th time-step, thus suppressing the costs to a negligible amount. Although it is not necessary, we perform the optimization with an order magnitude less particles to show



(a) *standard* sampling (b) *plain* sampling

Figure 2.9.: Effectiveness of different adaptive control variates for the two stream instability.

the stability. The kernel density estimation uses the Epanechnikov kernel at 25 grid points with $\triangle v = \frac{10}{25}$ and cubic spline interpolation. The Ansatz functions for the Gaussians with the initial parameters are

$$h(x,v,\alpha) = \alpha_1 \mathrm{e}^{-\frac{1}{2}\frac{(v-\alpha_2)^2}{\alpha_3^2}}, \quad \alpha = (1,0,1) \tag{2.212}$$

$$h(x,v,\alpha) = \alpha_1 \mathrm{e}^{-\frac{1}{2}\frac{(v-\alpha_2)^2}{\alpha_3^2}} + \alpha_4 \mathrm{e}^{-\frac{1}{2}\frac{(v-\alpha_5)^2}{\alpha_6^2}} \quad \alpha = (1, \frac{v_{\min}}{2}, 1,\ 1, \frac{v_{\max}}{2}, 1) \tag{2.213}$$

$$h(x,v,\alpha) = \alpha_1 \mathrm{e}^{-\frac{1}{2}\frac{(v-\alpha_2)^2}{\alpha_3^2}} + \alpha_4 \mathrm{e}^{-\frac{1}{2}\frac{(v-\alpha_5)^2}{\alpha_6^2}} + \alpha_7 \mathrm{e}^{-\frac{1}{2}\frac{(v-\alpha_8)^2}{\alpha_9^2}} \quad \alpha = (1, \frac{v_{\min}}{2}, 1,\ 1, 0, 1,\ 1, \frac{v_{\max}}{2}, 1). \tag{2.214}$$

One can also obtain initial guesses for α parameters by a clustering algorithm such as the very popular k–means. It is not used here because the built in MATLAB implementation is not suitable for the weighted samples. In general it is recommended to use Gaussian mixture models .

In the beginning the initial value is the best control variate. Figure 2.9a shows that for *standard* sampling no control variate is suitable for the nonlinear phase. For *plain* sampling, see fig. 2.9b, the kernel density estimator compares well to the initial value. Obviously the single Gaussian is not a good description, yet the double and triple Gaussian perform very well. The velocity profiles of the control variates at the end of the *plain* sampling simulations are given in fig. 2.10. We can conclude that stochastic optimization is a feasible approach in finding better control variates.

When the control variate h approximates the distribution function f very well, the difference $\delta f = f - h$ is not represented well by the particles distributed according to g, which again lies close to f. Yet for *plain* sampling g is flat, and a better approximation for the flat δf. In consequence on might want to change g such that it fits best to δf, which can be done by particle filtering.

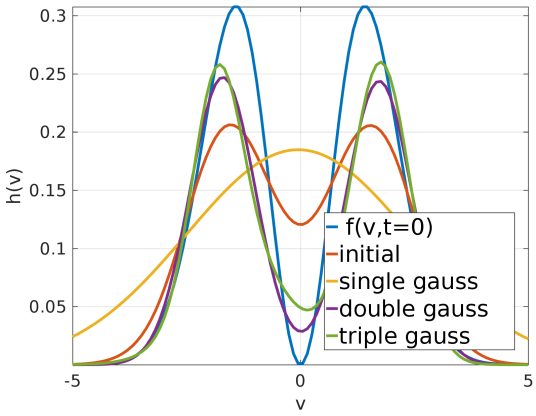


Figure 2.10.: Velocity profile of adaptive control variates for the two stream instability for $t = 80$.

2.3.5. Conditional Monte Carlo

Another popular method for variance reduction is conditional Monte Carlo. Special cases are antithetic sampling and stratification. The control variate reduced variance of an estimator for $\mathbb{E}[X]$ by the exact knowledge of the mean of another correlated random deviate Y. In conditional Monte Carlo the conditional expectation $\mathbb{E}[X|Y]$ is used, which has the same expectation

$$\mathbb{E}\big[\ \mathbb{E}[X|Y]\ \big] = \mathbb{E}[X]\,. \tag{2.215}$$

The conditional expectation can briefly be explained as the expectation of X given Y and is a random deviate. Its mean can be estimated again by standard Monte Carlo estimation. By the law of total variance the variance of the conditional expectation is always less or equal than the original random deviate

$$\mathbb{V}\left[\ \mathbb{E}[X|Y]\ \right] = \mathbb{V}[X] - \underbrace{\mathbb{E}\left[\mathbb{V}[X|Y]\right]}_{\geq 0} \leq \mathbb{V}[X]. \tag{2.216}$$

Instead of going into the probabilistic details, we begin with a motivating example everyone familiar with kinetic plasma simulations can understand.

Example: The gyroaverage operator

Gyrokinetic Particle-In-Cell codes are quite popular [86] and in the framework of gyrokinetic theory the gyroaverage operator appears. This operator averages a density along a circle, which has its origins in the circular gyromotion of a charged particle in a magnetic field. We consider a two dimensional example where we reuse the implementation of the one dimensional Finite element solver by a tensor product. The new mass matrix is then the Kronecker product of the one dimensional mass matrix. We follow the notation from [87], and define the gyroaverage at a position $(x_1, x_2) \in [0, 2\pi]^2$ as the integral over a circle with radius ρ_0 and center (x_1, x_2) as the gyroaveraging operator $\mathcal{J}$,

$$\mathcal{J}(\Phi) := \frac{1}{2\pi} \int_0^{2\pi} \Phi\left(x_1 + \rho_0 \cos(\alpha), x_2 + \rho_0 \sin(\alpha)\right) \mathrm{d}\alpha. \tag{2.217}$$

Set $\vec{\rho}(\alpha) = (\rho_0 \cos(\alpha), \rho_0 \sin(\alpha))$. Let $\mathcal{J}_{N_\alpha}$ denote the discrete approximation of the gyroaverage operator by numerical quadrature

$$\mathcal{J}(\Phi)(x) \approx \ \mathcal{J}_{N_\alpha}(\Phi)(x) := \frac{1}{N_\alpha} \sum_{k=0}^{N_\alpha - 1} \Phi\left(x_1 + \cos\left(k\frac{2\pi}{N_\alpha}\right), x_2 + \sin\left(k\frac{2\pi}{N_\alpha}\right)\right). \tag{2.218}$$

The quadrature points on the circle around the position x are called gyropoints. Let $g(x) = \frac{1}{2\pi}^2$ be a uniform sampling density of the random deviate $X = (X_1, X_2)$ and $\psi(x)$ two dimensional finite element basis functions. The gyroaveraged right hand side b_α is not anymore a two dimensional but three dimensional integral

$$\begin{aligned} b_\alpha =& \mathbb{E}\left[\mathcal{J}(\psi)(X)\frac{f(X)}{g(X)}\right] \\ =& \iint_{[0,2\pi]^2} \ \frac{1}{2\pi}\int_0^{2\pi} f(x)\ \psi\left(X + \vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\mathrm{d}x. \end{aligned} \tag{2.219}$$

Since we already used Monte Carlo integration for the two dimensional spatial integral and given the independence of dimension, it is quite natural to draw also samples of the gyroangle α and rewrite the three dimensional integral as an expectation. So for every particle (x_k) one draws a uniformly distributed gyroangle $\alpha_k \sim A \sim \mathcal{U}(0, 2\pi)$, which corresponds to distributing random points along a circle and use them for integration. We start with the gyroaverage over an expectation, then we use the random deviate A resulting in a expectation for the three dimensional integral. In the end the particle now also carries the gyroangle, which can be redrawn at any time step and does not depend on the characteristics.

$$\begin{aligned} b_\alpha \ &= \frac{1}{2\pi}\int_0^{2\pi} \mathbb{E}\left[\psi\left(X + \vec{\rho}(\alpha)\right)\ \frac{f(X)}{g(X)}\right]\ \mathrm{d}\alpha \\ &= \frac{1}{2\pi}\mathbb{E}\left[\psi\left(X + \vec{\rho}(A)\right)\ \frac{f(X)}{g(X)}\frac{1}{2\pi}\right] \\ &\approx \frac{1}{2\pi}\frac{1}{N_p}\sum_{k=1}^{N_p}\psi\left(x_k + \vec{\rho}(a_k)\right)\frac{w_k}{2\pi} \end{aligned} \tag{2.220}$$

But as a rule of thumb for Monte Carlo integration [88][p. 27], we should do the gyroaverage analytically in order to reduce the dimensionality and variance of the integral (2.219).

$$\int_0^{2\pi} \psi\left(x + \vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha = \mathbb{E}\left[\psi\left(x + \vec{\rho}(A)\right)\frac{1}{2\pi}\right] = \mathbb{E}\left[\psi\left(X + \vec{\rho}(A)\right)\frac{1}{2\pi}|X = x\ \right] \tag{2.221}$$

In (2.222), we use the notation of conditional expectation (2.221), which is Rao-Blackwellization [88][p. 27].

$$\begin{aligned} b_\alpha &= \frac{1}{2\pi}\iint_{[0,2\pi]^2} \mathbb{E}\left[\psi\left(X + \vec{\rho}(A)\right)|X = x\ \right] f(x)\ \mathrm{d}x \\ &= \frac{1}{2\pi}\mathbb{E}\left[\ \mathbb{E}\left[\psi\left(X + \vec{\rho}(A)\right)\frac{1}{2\pi}\ |\ X\right]\ \frac{f(X)}{g(X)}\right] \\ &= \mathbb{E}\left[\frac{1}{2\pi}\int_0^{2\pi}\psi\left(X + \vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \frac{f(X)}{g(X)}\right] \\ &\approx \frac{1}{N_p}\sum_{k=1}^{N_p}\left(\frac{1}{2\pi}\int_0^{2\pi}\psi\left(x_k + \vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \right)w_k \\ &= \frac{1}{N_p}\sum_{k=1}^{N_p}\ \frac{1}{2\pi}\mathbb{E}\left[\psi\left(X + \vec{\rho}(A)\right)\frac{1}{2\pi}\ |\ X = x_k\right]w_k \end{aligned} \tag{2.222}$$

The law of total variance for two random deviates X, Y with bounded variance reads

$$\mathbb{V}[Y] = \mathbb{E}[\mathbb{V}[Y|X]] + \mathbb{V}[\mathbb{E}[Y|X]].$$

Applying above decomposition formula allows us to compare the variance of the three dimensional Monte Carlo integral in (2.220) with the conditional estimator (2.221), which yields

$$\begin{aligned}
&\mathbb{V}\left[\psi\left(X+\vec{\rho}(A)\right)\ \frac{f(X)}{g(X)}\frac{1}{2\pi}\right] \\
&\quad = \mathbb{E}\left[\mathbb{V}\left[\psi\left(X+\vec{\rho}(A)\right)\ \frac{f(X)}{g(X)}\frac{1}{2\pi}|X\right]\right] + \mathbb{V}\left[\mathbb{E}\left[\psi\left(X+\vec{\rho}(A)\right)\ \frac{f(X)}{g(X)}\frac{1}{2\pi}|X\right]\right] \\
&\quad = \underbrace{\mathbb{E}\left[\mathbb{V}\left[\psi\left(X+\vec{\rho}(A)\right)\ \frac{f(X)}{g(X)}\frac{1}{2\pi}|X\right]\right]}_{*} + \underbrace{\mathbb{V}\left[\frac{1}{2\pi}\int_0^{2\pi}\psi\left(X+\vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \frac{f(X)}{g(X)}\right]}_{**}.
\end{aligned} \tag{2.223}$$

This is known as Blackwell's theorem [89], which states that the variance of the conditional estimator is always less or equal than the original estimator, see also [90][pp.107] for more examples.

$$\Rightarrow \mathbb{V}\left[\psi\left(X+\vec{\rho}(A)\right)\ \frac{f(X)}{g(X)}\frac{1}{2\pi}\right]\ \geq\ \mathbb{V}\left[\mathbb{E}\left[\frac{1}{2\pi}\int_0^{2\pi}\psi\left(X+\vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \frac{1}{2\pi}|X\right]\frac{f(X)}{g(X)}\right] \tag{2.224}$$

Suppose the costs of the gyro-integral are neglectable, on should always calculate the gyroaverage separately. But in real applications the gyroradius ρ_0 might be depend on another dimension, like the velocity, so it can be quite costly to construct a gyroaverage for specific basis functions. It also known from [91][p. 17] that a control variate, $(X - \mathbb{E}[X])$, reduces $(**)$ in (2.223), whereas the conditional Monte Carlo estimator eliminates the part $(*)$.
In Fourier space [87], [92] the gyroaverage can be computed for simple domains, by using e.g. Bessel functions. One can also use numerical quadrature [93] on the periodic domain $[0, 2\pi]$ by defining N_α integration points

$$\alpha_l := \frac{2\pi}{N_\alpha}(l-1) \tag{2.225}$$

and then approximating the gyroaverage numerically

$$\begin{aligned}
b_\alpha &= \frac{1}{N_p}\sum_{k=1}^{N_p}\left(\frac{1}{2\pi}\int_0^{2\pi}\psi\left(x_k+\vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \right)w_k \\
&\approx \frac{1}{N_p}\sum_{k=1}^{N_p}\left(\frac{1}{N_\alpha}\sum_{l=1}^{N_\alpha}\psi\left(x_k+\vec{\rho}(\alpha_l)\right)\ \right)w_k
\end{aligned}. \tag{2.226}$$

But now the number of basis function evaluation is multiplied by a factor N_α. That is problematic because these comprise the major costs of the charge assignment. Also it is unclear how many points are needed for a good approximation; [87] suggest $N_\alpha = 16$. An a priori error estimate for the quadrature rule is easily obtained from [93],[94].

$$\sup_{x\in[0,2\pi]^2}\left|\frac{1}{N_\alpha}\sum_{l=1}^{N_\alpha}\psi\left(x+\vec{\rho}(\alpha)\right)\ -\ \mathcal{J}(f)(x)\right| \leq \frac{2\pi}{12N^2}K_2 := \epsilon_\alpha \tag{2.227}$$

Here K^m is an arbitrary bound on the mth derivative of the integrand

$$K_m := \sup_{x\in[0,2\pi]^2}\sup_{\alpha\in[0,2\pi]}|\partial_\alpha^m\psi(x+\vec{\rho}(\alpha))|. \tag{2.228}$$

As main assumption the a priori estimate from [93] needs the integrand to be at least twice continuously differentiable. The circle $\vec{\rho}(\alpha)$ is a smooth mapping, but the basis functions have to be in $\psi \in \mathcal{C}^2$, which excludes linear and quadratic splines. In order to give an estimate of the constant K_2 we calculate the derivatives of $\vec{\rho}$.

$$\begin{aligned}\vec{\rho}'(\alpha) = \rho_0 \left(\sin(\alpha), \cos(\alpha)\right), \quad \vec{\rho}''(\alpha) = \rho_0 \left(-\cos(\alpha), -\sin(\alpha)\right) = -\vec{\rho}(\alpha) \\ \Rightarrow \|\vec{\rho}(\alpha)\| = \|\vec{\rho}'(\alpha)\| = \|\vec{\rho}''(\alpha)\| = |\rho_0| = \rho_0. \end{aligned} \tag{2.229}$$

These are plugged into (2.228) along with the gradient $\nabla\psi$ and Hessian $\nabla^2\psi$.

$$\begin{aligned} K_2 &= \sup_{x\in[0,2\pi]^2} \sup_{\alpha\in[0,2\pi]} \left| \vec{\rho}'(\alpha) \cdot \nabla^2\psi(x + \vec{\rho}(\alpha)) \cdot \vec{\rho}'(\alpha)^t + \vec{\rho}'(\alpha) \cdot \nabla\psi(x + \vec{\rho}(\alpha)) \right| \\ &\leq \rho_0^2 \sup_{x\in[0,2\pi]^2} \|\nabla^2\psi(x)\| + \rho_0 \sup_{x\in[0,2\pi]^2} \|\nabla\psi(x)\| \\ &= \rho_0^2 \|\nabla^2\psi(x)\|_\infty + \rho_0 \|\nabla\psi\|_\infty \\ &\leq \left(\rho_0^2 + \rho_0(2\pi)^2\right) \|\nabla^2\psi(x)\|_\infty \end{aligned} \tag{2.230}$$

For large gyroradius ρ_0 the ρ^2 term dominates and the quadrature error can be approximated by

$$\epsilon_\alpha \approx \frac{2\pi}{12} \|\nabla^2\psi(x)\|_\infty \left(\frac{\rho_0}{N}\right)^2, \tag{2.231}$$

where we immediately see that the number of quadrature points N should be proportional to ρ_0. This is already numerically verified in [87], when choosing the number of gyropoints proportional to the arc length $2\pi\rho_0$. For higher order smoothness, one can gain similar results. Using the Simpson rule for $\psi \in \mathcal{C}^4$ [94][p. 385] the quadrature error can be bounded up to

$$\left(\frac{2\pi}{2N}\right)^5 \frac{K_4}{90}. \tag{2.232}$$

The approximation error of the gyroaverage by numerical quadrature ϵ_α is additional to the discretization error of the basis functions an extra bias when estimating the right hand side. It then becomes clear that this error should be balanced with the particle number. So the noise level introduced by the variance should always dominate and by (2.233) a ratio of $\frac{1}{N_p} \sim \frac{1}{N_\alpha}$.

$$\frac{1}{N_p} \mathbb{V}\left[\frac{1}{2\pi} \int_0^{2\pi} \psi\left(X + \vec{\rho}(\alpha)\right) \, \mathrm{d}\alpha \, \frac{f(X)}{g(X)}\right] > (\epsilon_\alpha)^2 \tag{2.233}$$

In the comparison (2.233) between Monte Carlo integration in two dimensions and a one dimensional quadrature rule, a precise balance is achieved by the knowledge of the included constants, where the variance can be estimated using some subsamples and K_m, defined in eqn. (2.228), can be calculated. Now that the gyroaverage is not analytically known, it is not guaranteed that Rao-Blackwellization (2.222) is always better than (2.220), because the bias can dominate the error.
But we can include such a guarantee, by combining the quadrature rule and Monte Carlo integration, which corresponds to stratified sampling. As a first slight modification we choose A to be uniformly distributed in $[0, 1]$.

$$A \sim \mathcal{U}(0, 1) \tag{2.234}$$

Following [65][p. 16-19], we split the gyroangle integration domain $\Omega = [0, 2\pi]$ into N_α sub domains with $\Omega_l := [(l-1)\frac{2\pi}{N_\alpha}, l\frac{2\pi}{N_\alpha}]$ and integrate over all sub domains. Then all of these

integrals can be written as an expectation over the random deviate A.

$$\begin{aligned} b_\alpha &= \frac{1}{2\pi}\mathbb{E}\left[\int_\Omega \psi\left(X+\vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \frac{f(X)}{g(X)}\right] = \frac{1}{2\pi}\mathbb{E}\left[\sum_{l=1}^{N_\alpha}\int_{\Omega_l} \psi\left(X+\vec{\rho}(\alpha)\right)\ \mathrm{d}\alpha\ \frac{f(X)}{g(X)}\right] \\ &= \frac{1}{2\pi}\mathbb{E}\left[\sum_{l=1}^{N_\alpha}\frac{2\pi}{N_\alpha}\int_0^1 \psi\left(X+\vec{\rho}\left(\alpha\frac{2\pi}{N_\alpha}+(l-1)\frac{2\pi}{N_\alpha}\right)\right)\ \mathrm{d}\alpha\ \frac{f(X)}{g(X)}\right] \\ &= \mathbb{E}\left[\frac{1}{N_\alpha}\sum_{l=1}^{N_\alpha}\mathbb{E}\left[\psi\left(X+\vec{\rho}\left(A\frac{2\pi}{N_\alpha}+(l-1)\frac{2\pi}{N_\alpha}\right)\right)\Big|X\right]\ \frac{f(X)}{g(X)}\right] \\ &= \mathbb{E}\left[\frac{1}{N_\alpha}\sum_{l=1}^{N_\alpha}\psi\left(X+\vec{\rho}\left(A\frac{2\pi}{N_\alpha}+(l-1)\frac{2\pi}{N_\alpha}\right)\right)\ \frac{f(X)}{g(X)}\right] \end{aligned} \tag{2.235}$$

The random deviate A shifts the quadrature points by a random offset (2.225) and quadrature remains the same. But the integration domain is divided, which yields a variance reduction. Note that $\rho(\alpha)$ can also be dependent on x or the velocity v. Most general it can be some parametric curve depending on the phase space position of the marker. The stochastic derivations above are then still valid, except for everything including the Bessel function. In contrast to grid-based methods, the gyropoints are only needed for the numerical quadrature in the charge projection phase. Due to the Laplace operator the potential is much smoother than the charge density. For a given accuracy one needs less quadrature points for the smoother potential and the charge density is not a problem anyhow. Therefore, the absolute number of required gyropoints will differ from a grid-based method, where also the charge density has to be integrated.

A two dimensional numerical example

We start with a small gyroaverage example in two dimensions without particle mesh coupling in order to obtain an easy demonstration. The manufactured reference is a two dimensional perturbation with random coefficients $\beta_{k_x,k_y} \sim \mathcal{U}(0,1)$ and random phase shift $\varphi_{k_x,k_y} \sim \mathcal{U}(0,2\pi)$,

$$f(x,y) = \sum_{k_x=1}^{M}\sum_{k_y=2}^{M}\beta_{k_x,k_y}\cos\left(k_x x + k_y y + \varphi_{k_x,k_y}\right). \tag{2.236}$$

Here the gyroaverage can be easily computed using the Fourier transformation of f [87][p. 487] and the Bessel function of first kind $\mathcal{J}_0$.

$$\mathcal{J}(f)(x,y) = \sum_{k_x=1}^{M}\sum_{k_y=2}^{M}\beta_{k_x,k_y}\cos\left(k_x x + k_y y + \varphi_{k_x,k_y}\right)\mathcal{J}_0\left(\rho_0\sqrt{k_x^2+k_y^2}\right). \tag{2.237}$$

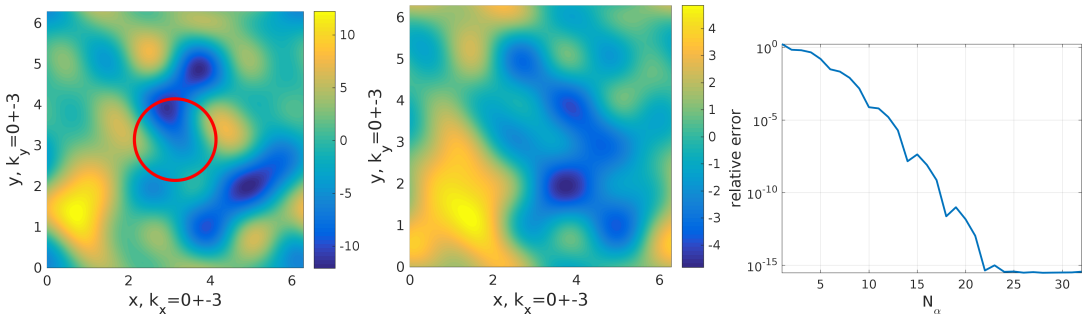
The first test is a convergence study over the number of gyropoints N_α versus the L^2 error integration error

$$\left\|\mathcal{J}(f) - \mathcal{J}_{N_\alpha}(f)\right\|_2. \tag{2.238}$$

We see in fig. 2.11c that for $N_\alpha = 22$ the approximation $\mathcal{J}_{N_\alpha}$ reaches machine precision. This figure shall be used as reference for the following studies.

Here the goal is to find the value of the integral

$$\int_0^{L_x}\int_0^{L_y}\mathcal{J}(f)(x,y)\omega(x,y)\ \mathrm{d}x\mathrm{d}y, \quad \omega(x,y) := (x-L_x)x\ (y-L_y)y. \tag{2.239}$$



(a) Density f with gyroring for $\rho_0 = 1$ (b) Averaged density $\mathcal{J}(f)$ (c) L^2 approximation error for $\mathcal{J}_{N_\alpha}(f)$

Figure 2.11.: Gyroaveraging a perturbed density by a discrete gyroring with N_α gyropoints and a constant gyroradius $\rho_0 = 1$.

We approximate (2.239) by the standard Monte Carlo estimator for four different expectations. Let $X \sim \mathcal{U}(0, L_x)$, $Y \sim \mathcal{U}(0, L_y)$ and be $A \sim \mathcal{U}(0, 1)$ which yields the respective sampling density $g(x, y) := \frac{1}{L_x L_y}$.

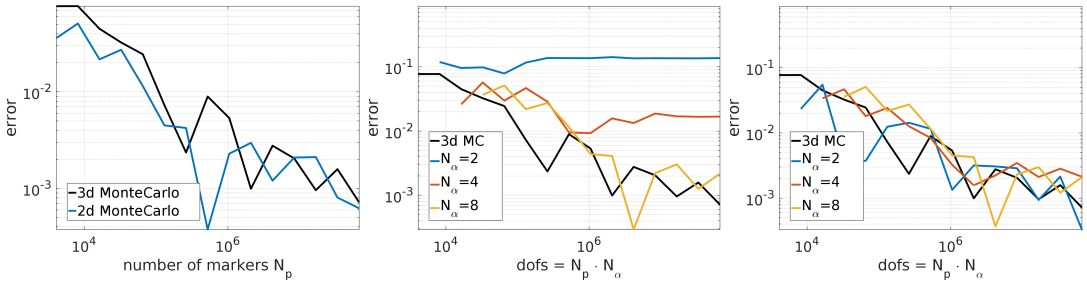
$$\mathbb{E}\left[f\left(X + \rho_0 \cos(2\pi A), y + \rho_0 \sin(2\pi A)\right) \frac{\omega(X,Y)}{g(X,Y)}\right] \tag{2.240}$$

$$\mathbb{E}\left[\mathcal{J}(f)(X,Y)\frac{\omega(X,Y)}{g(X,Y)}\right] = \mathbb{E}\left[\mathbb{E}\left[f\left(X + \rho_0 \cos(2\pi A), y + \rho_0 \sin(2\pi A)\right)\middle| X\right] \frac{\omega(X,Y)}{g(X,Y)}\right] \tag{2.241}$$

$$\mathbb{E}\left[\mathcal{J}_{N_\alpha}(f)(X,Y)\frac{\omega(X,Y)}{g(X,Y)}\right] \tag{2.242}$$

$$\mathbb{E}\left[\frac{1}{N_\alpha}\sum_{l=1}^{N_\alpha} f\left((X,Y) + \vec{\rho}\left(A\frac{2\pi}{N_\alpha} + (l-1)\frac{2\pi}{N_\alpha}\right)\right) \frac{\omega(X,Y)}{g(X,Y)}\right] \tag{2.243}$$

The simplest estimate is the standard three dimensional integral (2.240). Here we can calculate the gyroaveraged density $\mathcal{J}(f)$ of f directly which allows for the two dimensional integration (2.241). In general the gyroaverage can be approximated by numerical quadrature and a fixed amount of gyropoints N_α, which yields a two dimensional Monte Carlo estimator in (2.242). To overcome the bias associated with the fixed number of gyropoints N_α the Rao-Blackwell estimator (2.243) is used. As expected the analytical gyroaverage (2.241) gives a better approximation, see fig. 2.12a, than the standard estimator (2.240). Yet it is hard to compare the actual costs of the gyroaverage, since evaluating a Bessel function is quite expensive. In order to have a fair comparison between (2.240) and (2.242) or (2.243) respectively, we plot the degrees of freedom $N_p N_\alpha$ which is the number of function evaluations and represents the actual costs in the particle mesh coupling. The behavior of the quadrature estimator (2.242) in fig. 2.12b is dominated by bias introduced by the discretization error due to the finite number of gyropoints. For $N_\alpha = 2$ there is no convergence, yet for $N_\alpha = 4$ gyropoints the estimator converges. But when N_p lowers the noise level to $\sim 10^{-2}$ the discretization error of the gyroaverage dominates and convergence stops - the estimator is not asymptotically unbiased. In order to achieve convergence one has to adapt N_α to N_p. There is also little to no gain over the 3d estimator (2.240). The last estimator (2.243), see fig. 2.12c, obviously overcomes the bias limitation, yielding only a variance reduction by construction yet only slight increase in efficiency. For a second test, we fix $N_p = 10^6$ and vary the gyroradius ρ_0. Then we estimate the variances of the standard Monte Carlo estimators, which

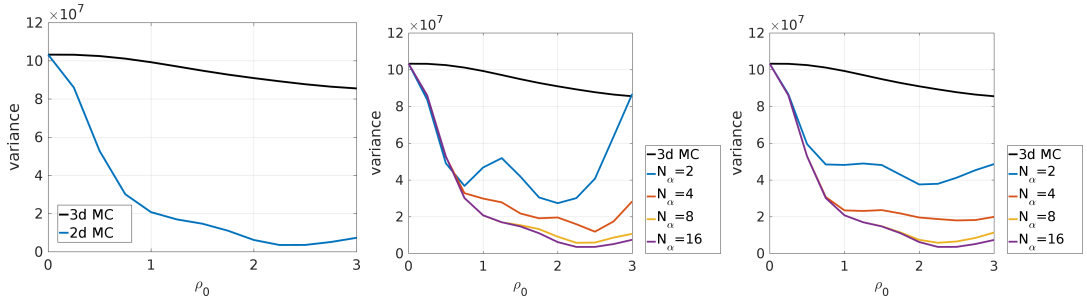


(a) 3d (2.240) and 2d conditional (2.241) estimator.

(b) 2d estimator using quadrature (2.242)

(c) 3d estimator using Rao-Blackwell variance reduction based on quadrature (2.243)

Figure 2.12.: Estimating the integral in eqn. (2.239) with pseudo random numbers and the standard Monte Carlo estimator for different expectations.



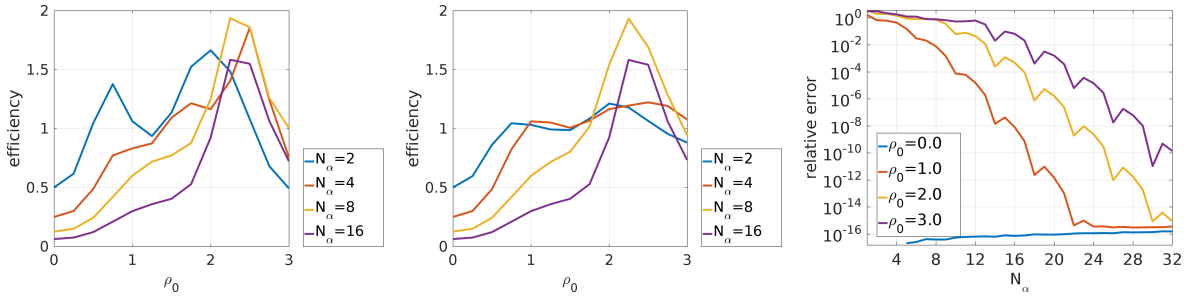
(a) 3d (2.240) and 2d conditional (2.241) estimator.

(b) 2d estimator using quadrature (2.242)

(c) 3d estimator using Rao-Blackwell variance reduction based on quadrature (2.243)

Figure 2.13.: Estimated variances of the standard Monte Carlo estimator for different expectations approximating the integral in eqn. (2.239) with variation over the gyroradius ρ_0.

gives a much better indication of the performance increase than the noisy convergence studies before. The first result in fig. 2.13a reveals once again the variance reduction by reduction of dimensions. For large gyroradii this effect becomes the strongest, but we also notice that in general a larger gyroradius ρ_0 yields a slightly smaller variance. Both the biased, fig. 2.13b, and unbiased, fig. 2.13c, estimators yield better variance reduction with increasing number of gyropoints N_α. Yet the biased estimator shows again strange behavior as for $N_\alpha = 2$ the variance approaches the standard 3d estimate. By Rao Blackwells theorem it is clear that the variance decreases with increasing N_α. Although for a large gyroradius one can achieve quite a high variance reduction, the question of efficiency is unclear. This point is already addressed in fig. 2.12c when the error is plotted against the degrees of freedom $N_\alpha N_p$, which accounts for the cost - degrees of freedom. We can make a similar comparison by dividing the variance of the 3d estimator by N_α and compare with the corresponding 2d estimator. In fig. 2.14a and fig. 2.14b one can see that the use of gyropoints is inefficient for a small gyroradius. With growing gyroradius a growing number of gyropoints reaches efficiency one. The 2d estimators become efficient with large gyroradius and quite many gyropoints. In case there are no significant performance gains, in the rest of the particle code except the particle mesh coupling by using less particles, the safe way out is to use the standard 3d Monte Carlo estimator. But there is also the charge assignment, where a second gyroaverage over the estimator of the electric field $\mathcal{J}(\hat{E})(x)$ is introduced. Here it is clear that the approximation



(a) Gyropoint estimator eqn. (2.242) (b) Rao Blackwell eqn. (2.243) (c) L^2 approximation error for $\mathcal{J}_{N_\alpha}(f)$.

Figure 2.14.: Efficiency of variance reduction via gyropoints and quadrature error under varying gyroradius ρ_0.

should always be below the noise level of the electric field. This noise level is, due to the variance propagation, not the same as the one in the charge assignment. Thus best practice is to use the 3d estimator and then decide on the number of gyropoints for the second gyroaverage adaptively later. But since the particle discretization with the gyroaverage is done at the level of the Lagrangian, one has to use the same gyroaverage technique in both steps. Since the optimal N_α differs in the two steps, one has to use (2.243) in the particle to grid projection in order to not damage the convergence. Then N_α can be chosen to fulfill the inequality

$$\|\mathcal{J}(\hat{E})(x) - \mathcal{J}_{N_\alpha}(\hat{E})(x)\|^2 \leq \mathbb{V}\left[\hat{E}(x)\right], \tag{2.244}$$

where all appearing quantities can be easily estimated. Considering the example here, we can consult fig. 2.14c that the varying gyroradius leads to a one percent error for N_α varying from 8 to 22. The main insight here is to use more gyropoints with increasing number of particles. Not because of the charge projection but because of the charge assignment.

(Post-)Stratification and coarse graining

Stratification, Antithetics and Latin hypercube sampling can greatly reduce the variance. In general they can be applied at the initialization, but they will loose their effect in the nonlinear phase. Although, if we know a dimension to be particularly unperturbed, but important then Latin hypercube sampling may be helpful. Nevertheless, it is easy to sample the initial condition, but very expensive (running the characteristics backwards) to draw a sample at a later time. Therefore, most algorithms are not applicable. First we describe how basic stratified sampling can improve sampling of the initial condition. For this the domain Ω is partitioned into N_s disjunct strata Ω_j

$$\Omega = \dot{\bigcup}_{j=1,\dots,N_s} \Omega_j. \tag{2.245}$$

Stratification uses a known conditional information for a random variable with respect to the stratum namely the probability that the random variable Z is contained in the stratum Ω_j denoted by ω_j.

$$\omega_j := \mathcal{P}\left(Z \in \Omega_j\right) = \int_{\Omega_j} g(z)\mathrm{d}z \tag{2.246}$$

If g is known then ω_j can be calculated, which holds for the initial condition but not at later times. In the following the exact information given in eqn. (2.246) is used for variance

reduction by conditional Monte Carlo. For this the conditioned density g_j reduces g onto Ω_j and reads

$$g_j(z) := \frac{g(z)}{\omega_j} \mathbb{1}_{z \in \Omega_j}. \tag{2.247}$$

Stratification samples $n_j = \frac{N_p}{N_s}$ independent markers $Z_{j,k}, \quad k = 1, \ldots, n_j$ in every stratum Ω_j according to the probability density g_j.

$$\int_\Omega \psi(z) g(z) \, \mathrm{d}z = \mathbb{E}\left[\psi(Z)\right] = \sum_{j=1}^{N_s} \int_{\Omega_j} \psi(z) g_j(z) \, \mathrm{d}z = \sum_{j=1}^{N_s} \mathbb{E}\left[\psi(Z) | Z \in \Omega_j\right] \approx \sum_{j=1}^{N_s} \frac{\omega_j}{n_j} \sum_{k=1}^{n_j} \psi(Z_{j,k}) \tag{2.248}$$

Most important, a variance reduction is guaranteed by the law of total variance, see eqn. (2.249). This is also known as the Rao-Blackwellization [95] or Riemann Monte Carlo and works best if the sampling density is constant in each stratum.

$$\mathbb{V}[\psi(Z)] \;\geq\; \sum_{j=1}^{N_s} \mathbb{V}\left[\psi(Z) | Z \in \Omega_j\right] \tag{2.249}$$

Note that the stratified estimator in eqn. (2.248) can be rewritten as the standard Monte Carlo estimator with an additional weighting factor s_k.

$$\sum_{j=1}^{N_s} \frac{\omega_j}{n_j} \sum_{k=1}^{n_j} \psi(Z_{j,k}) = \frac{1}{N_p} \sum_{j=1}^{N_s} s_k \psi(Z_k), \quad s_k = \frac{N_p \omega_j}{n_j} \text{ for } Z_k \in \Omega_j \tag{2.250}$$

A detailed description of the method is proposed in [69][Chapter 8.4], where also the connection to control variates is made. Essentially the stratification is a control variate comprised of indicator functions for every stratum - phase space boxes. A distribution which is constant within the stratum yields a high correlation and a good control variate. If ω_j is known but we cannot sample from $g_j(z)$ direct post-stratification is the most obvious step. There, already given samples Z_k from g are assigned to their stratum, making n_j a random number. In the following $Z_{j,k}$ denotes the kth marker in the jth stratum. The estimator (2.248) remains unchanged. The given samples correspond to particles at later times sorted into strata. But ω_j is not exactly known because the sampling density is not available at later times.
In order to estimate ω_j the strata can be used as the bins of a histogram. This proportional allocation estimate (2.251) reduces the stratified mean (2.248) again to the sample mean, hence nothing has been gained.

$$\omega_j = \mathcal{P}\left(Z \in \Omega_j\right) \approx \frac{n_j}{N_p} \tag{2.251}$$

Yet the additional information available, but not being used is the value $g(Z_k) = g_k$, which is transported along the characteristics by the markers. Suppose there are very few markers in every stratum, one can just assume them to be uniformly distributed. This key assumption allows for the approximation of the integral of the actual density g over the stratum Ω_j by eqn. (2.252).

$$\omega_j = \int_{\Omega_j} g(z) \, \mathrm{d}z = \int_{\Omega_j} \frac{g(z)}{|\Omega_j|} |\Omega_j| \mathrm{d}z \approx \frac{|\Omega_j|}{n_j} \sum_{k=1}^{n_j} g(Z_{j,k}) \tag{2.252}$$

The estimator in eqn. (2.252) obviously needs some normalization in order to ensure mass conservation $\sum_j \omega_j = 1$. But this falls into the regime of moment matching and re-weighting

techniques. Here we are interested in the connection between eqn. (2.252) and coarse graining like in [15]. In Sonnendrückers coarse graining technique the particles are sorted in a quadtree structure, which corresponds to the strata Ω_j here. One starts with the domain Ω and divides by bisection until there are not more than n_j of markers in every subset Ω_j. So until now no difference to the (post)-stratification. Then it is decided [15], that the distribution function represented by the control variate weights δw should be constant in each stratum. In order to enforce this, the weights δw are smoothed within every stratum similar to what we do with the g_k in eqn. (2.252). Essentially the coarse graining method is not necessary because of the control variate, but because of the Fokker–Planck collisions, which change the constant likelihoods f_k and g_k but not their ratio. Nevertheless the main idea is to neglect information below a certain scale - the size of the strata. Therefore, for coarse graining we suppose that the ratio between f and g should be preserved, leading to the estimator

$$\mathbb{E}\left[\psi(Z)|Z \in \Omega, \frac{f(Z)}{g(Z)} = C\right]. \tag{2.253}$$

By selecting a small enough box in the marker density it is reasonable to assume the markers to be uniformly distributed in that box, hence the approximation uniform density $z \mapsto |\Omega_j|^{-1}$ in (2.252). This approximation leads to a biased estimator which does not guarantee the variance to be less or equal than in standard estimation. On the contrary exact knowledge of ω_j is beneficial. An improvement of the estimator (2.252) is to use a multidimensional quadrature rule on the stratum Ω_j with the given quadrature nodes $Z_{j,k}$ and the function values $g_{j,k}$ for $k = 1, \ldots, n_j$. Eventually something similar to the (RID) in [96]. The most straightforward idea is to interpolate the values of g in the stratum by multivariate Lagrange polynomials and deduce the integral from there. If there are very few markers per stratum this can be hard coded circumventing the Vandermonde matrix. Although the nodes are not equidistantly spaced Runge's phenomenon could cause a problem. Another approach is to use a more expensive quadrature rule based on Voronoi tessellation (VT) which described in [97][p. 643]. But the Voronoi methods lead down another *very* promising path.
Let us consider the extreme case of exactly one marker Z_j in each stratum Ω_j, which requires obviously some advanced form of partitioning the domain Ω. Then the estimate $\omega_j \approx g(Z_{k,j})|\Omega_j|$ needs again the volume of the presumably quite deformed Ω_j. In [97] such a set of strata optimizing the quadrature weights implied by ω_j is called a Voronoi tessellation (VT). In general the Ω_j form a tessellation of Ω. The VT returns Ω_j as convex polyhedra such that the volume is merely the convex hull. Lloyd's algorithm repeatedly adapts the phase space coordinate of the markers Z_j, in the Voronoi context called generators, to the center of the Voronoi cell Ω_j which yields after some iterations the Centroidal Voronoi Tessellation (CVT). The CVT forms an optimal quadrature rule [97][p. 643] but it is NP hard to find. Fast algorithms are given in [98]. Hence it is useful for initialization of particles in complicated geometries. A direct benchmark concerning the improvement of Monte Carlo by Voronoi volumes [99] finds large inaccuracies stemming from the boundary cells. For periodic domains the periodic Voronoi diagram (PVD) [100] is suitable. According to [99] the PVD yields also better convergence. In [101] the connection between Monte Carlo stratification and the Voronoi tessellation is made by the definition of quantizers. In general Voronoi tessellation leads to rather deterministic methods [102, 103]. In [103] markers are merged within a Voronoi cell remarkably with conservation of phase space. Since this also modifies the phase space position it does not anymore fall into the regime of stratification.
Stratification combines grids and Monte Carlo which according to [69] can improve the convergence order to $\mathcal{O}(N_p^{-\frac{1}{2}-\frac{1}{d}})$ where d denotes the dimension. Thus, it is most effective in low dimensions, which is precisely the regime most PIC codes are operating in. By antithetic

sampling within a stratum an order of $\mathcal{O}(N_p^{-\frac{1}{2}-\frac{2}{d}})$ is possible. Stratification, when restricted to the initial condition, competes there also with the Latin hypercubes and Quasi Monte Carlo sequences. As we will see later their nice properties are somehow lost during the non-linear simulation defaulting back to the square root $\sqrt{N_p}$ convergence. Here the possibility of **post**-stratification is clearly interesting because it does not touch the characteristics but merely changes the weighting. This approach also avoids degenerating effects, since the main information in the likelihoods f_k and g_k is never changed.

Stratified control variates

Post-stratification can be reformulated as a control variate technique. The entire derivation is given in [104][pp.15-18]. Then given the strata $(\Omega_j)_{j=1,\dots,N_s}$ the condition $Z \in \Omega_j$ is used as a control variate of piecewise constant indicator functions in eqn. (2.254).

$$\begin{aligned}\mathbb{E}\left[\frac{f(Z)}{g(Z)}\psi(Z)\right] &= \mathbb{E}\left[\frac{f(Z)}{g(Z)}\psi(Z) - \sum_{j=1}^{N_s}\alpha_j \mathbb{1}_{Z\in\Omega_j}\right] + \sum_{j=1}^{N_s}\alpha_j \int_{\Omega_j} g(z)\mathrm{d}z \\ &= \sum_{j=1}^{N_s}\mathbb{E}\left[\frac{f(Z)}{g(Z)}\psi(Z) \;\middle|\; Z\in\Omega_j\right]\end{aligned} \tag{2.254}$$

According to [104] variance minimization yields the coefficients α as

$$\alpha_j = \frac{\mathbb{E}\left[\frac{f(Z)}{g(Z)}\psi(Z) \;\middle|\; Z\in\Omega_j\right]}{\int_{\Omega_j} g(z)\mathrm{d}z}, \tag{2.255}$$

which is the reason why the control variate in eqn. (2.254) is equivalent to post-stratification. This result underlines the generality of the control variate method. Yet the main difficulty of post-stratification mechanism, the estimation of $\int_{\Omega_j} g(z)\mathrm{d}z$ from the likelihoods g_k remains. The combination of piecewise definitions motivates the combination of the traditional control variate h and conditional Monte Carlo by decomposition of the control variate onto several strata. Given a control variate $h(z)$ in the δf framework and strata $(\Omega_j)_{j=1,\dots,N_s}$ the decomposition of the single control variate h into N_s control variates h_j is defined as

$$h_j(z) := \begin{cases} h(z) & \text{for } z\in\Omega_j \\ 0 & \text{else}\end{cases}, \quad \text{for } j = 1,\dots,N_s. \tag{2.256}$$

The new control variate reads then

$$\begin{aligned}\mathbb{E}\left[\frac{f(Z)}{g(Z)}\psi(Z)\right] &= \mathbb{E}\left[\frac{f(Z)-\alpha h(Z)}{g(Z)}\psi(Z)\right] + \alpha\int_\Omega h(z)\psi(z)\mathrm{d}z \\ &= \mathbb{E}\left[\frac{f(Z)-\sum_{j=1}^{N_s}\alpha_j h_j(Z)}{g(Z)}\psi(Z)\right] + \sum_{j=1}^{N_s}\alpha_j\int_{\Omega_j} h_j(z)\psi(z)\mathrm{d}z.\end{aligned} \tag{2.257}$$

The (Ω_j) form a disjoint decomposition of ω_j, the optimality coefficients (α_j) can be calculated on basis of sparse matrix algebra yielding a fast implementation.

2.3.6. Control variates and geometric integration

Geometric integration is a vast research area that yields for solving systems of ODEs [37]. For applications in plasma physics including the Vlasov equation, see we refer to Kraus'

work [14]. Extensions to stochastic differential equations are mostly made for the general Langevin equation [105, 106] and can thus be adapted for Vlasov–Poisson–Fokker–Planck or the Ornstein–Uhlenbeck process respectively. The original concept of a particle Lagrangian for PIC dates back to Lewis[8], but we recommend the recent review of the canonical variational PIC scheme provided in [10]. Here we already rely on structure preserving methods for the Vlasov equation, since the discrete phase space volume shall be conserved such that the likelihoods f and g are propagated correctly, see eqn. (2.9). Stochastic differential equations can be solved with methods of geometric integration yet it is unclear how time dependent variance reduction techniques such as the control variate can be included. First the equations of motions which are the characteristics $(x(t), v(t))$ transporting the initial condition $f(x_0, v_0, 0) = f(x(t), v(t), t)$ shall be obtained by a Euler-Lagrange principle, respectively Euler–Poincare reduction. This includes the definition of the flow $(x(t), v(t)) = \varphi(x_0, v_0, t)$. We use the Lagrangian for our electron Vlasov–Poisson system [14][p. 119] given as

$$L(x, \dot{x}, v, \dot{v}, \Phi, \dot{\Phi}) = \int f(x_0, v_0, 0) \left[x\dot{v} - \frac{1}{2}v^2 - \Phi(x,t) \right] \mathrm{d}x_0 \mathrm{d}v_0 + \frac{1}{2} \int (\partial_x \Phi(x,t))^2 \, \mathrm{d}x. \tag{2.258}$$

For a Lagrangian $L(q, \dot{q}, \Phi, \dot{\Phi})$ depending on coordinates $q, \dot{q}$ and the equations of motion and the field equations can be derived by the Euler–Lagrange equations:

$$\begin{aligned} &\frac{\partial L}{\partial q}(q, \dot{q}, \Phi, \dot{\Phi}) - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{q}}(q, \dot{q}, \Phi, \dot{\Phi}) = 0 \\ &\qquad \left\langle \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{\Phi}} + \frac{\partial}{\partial x} \frac{\partial L}{\partial (\partial_x \Phi)}, \delta\Phi \right\rangle = \left\langle \frac{\partial L}{\partial \Phi}, \delta\Phi \right\rangle \quad \forall \ \delta\Phi. \end{aligned} \tag{2.259}$$

The particle discretization replaces the distribution function $f(x_0, v_0, 0)$ at initial time and the integral over (x_0, v_0) in eqn. (2.258) with an expectation including the random deviates X_0, V_0 or by inserting the Klimontovich density $f_p(x, v, t) = \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \delta(x - x_n^t) \delta(v - v_n^t)$. This includes a change of notation since the flow transports each initial condition separately $x_n^t, v_n^t = \varphi(x_n^0, v_n^0)$ resulting in many characteristics.

$$L_p(x, v, \Phi, \dot{\Phi}) = \frac{1}{N_p} \sum_{n=1}^{N} \underbrace{\frac{f(x_n^0, v_n^0, 0)}{g(x_n^0, v_n^0, 0)}}_{=w_n} \left[x_n \dot{v}_n^t - \frac{1}{2}(v_n^t)^2 + \Phi(x_n^t, t) \right] - \frac{1}{2} \int |\partial_x \Phi(x,t)|^2 \, \mathrm{d}x. \tag{2.260}$$

By applying the Euler Lagrange equations (2.259) for each particle and the fields the standard equations of motion for each particle and the weak form of the Poisson equation are recovered in eqn. (2.261).

$$\frac{\mathrm{d}}{\mathrm{d}t} x_n^t = v_n^t \tag{2.261}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} v_n^t = \partial_x \Phi(x_n^t, t) \tag{2.262}$$

$$\int \partial_x \Phi(x,t) \cdot \partial_x \delta\Phi(x) \, \mathrm{d}x = \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \delta\Phi(x_n^t) \text{ for all } \ \delta\Phi \tag{2.263}$$

Mostly the same discretizations for the test function $\partial\Phi$ and the Ansatz function Φ is chosen. The Lagrangian (2.260) does not contain any control variate. Implementing the control variate on top of the obtained scheme (2.261) changes its properties in an unknown way. Therefore, the control variate shall be introduced in this framework. We recall from eqn. (2.153)

the Klimontovich density containing f_p^* a control variate

$$f_p^*(x,v,t) := f_p(x,v,t) - \alpha \frac{1}{N_p} \sum_{n=1}^{N_p} \underbrace{\frac{h(x_n^t, v_n^t)}{g(x_n(t), v_n(t), t)}}_{:=\gamma_n^t = \delta w_n^t - w_n} \delta\left(x - x_n^t\right) \delta\left(v - v_n^t\right) \;+\; \alpha h(x,v), \tag{2.264}$$

which can be inserted into the original Lagrangian (2.258) yielding

$$\begin{aligned} L_p^*(x,v,\Phi,\dot{\Phi}) = \frac{1}{N_p} \sum_{n=1}^{N} \underbrace{\frac{f(x_n^0, v_n^0, 0) - \alpha h(x_n^0, v_n^0)}{g(x_n^0, v_n^0, 0)}}_{=\delta w_n^0} \left[x_n \dot{v}_n^t - \frac{1}{2}(v_n^t)^2 + \Phi(x_n^t, t) \right] \\ - \frac{1}{2} \int |\partial_x \Phi(x,t)|^2 \, \mathrm{d}x + \alpha \int h(x_0, v_0) \left[x\dot{v} - \frac{1}{2} v^2 - \Phi(x,t) \right] \mathrm{d}x_0 \mathrm{d}v_0. \end{aligned} \tag{2.265}$$

Here a difficulty arises when applying the Euler Lagrange principle to eqn. (2.265) since there is no discretization for h present. We suppose that h follows the same Vlasov equation under the same flow φ as the particles and obtain the equations of motions as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} x_n^t &= v_n^t \\ \frac{\mathrm{d}}{\mathrm{d}t} v_n^t &= \partial_x \Phi(x_n^t, t) \\ \partial_t h(x,v,t) &= -v \partial_x h(x,v,t) + \partial_x \Phi(x,t) \partial_v h(x,v,t) \\ \int \partial_x \Phi(x,t) \cdot \partial_x \delta\Phi(x) \, \mathrm{d}x &= \frac{1}{N_p} \sum_{n=1}^{N_p} \delta w_n^0 \delta\Phi(x_n^t) + \alpha \int h(x,v,t) \delta\Phi(x) \mathrm{d}x \mathrm{d}v \quad \forall \, \delta\Phi, \end{aligned} \tag{2.266}$$

which are for the case of $h(x,v,t) = h(v)$ being an equilibrium reduced to

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} x_n^t &= v_n^t \\ \frac{\mathrm{d}}{\mathrm{d}t} v_n^t &= \partial_x \Phi(x_n^t, t) \\ \int \partial_x \Phi(x,t) \cdot \partial_x \delta\Phi(x) \, \mathrm{d}x &= \frac{1}{N_p} \sum_{n=1}^{N_p} \delta w_n^0 \delta\Phi(x_n^t) + \alpha \int h(v) \delta\Phi(x) \mathrm{d}x \mathrm{d}v \quad \forall \, \delta\Phi. \end{aligned} \tag{2.267}$$

There are for sure cleaner and more elaborate ways to arrive at eqn. (2.267), since the flow φ under the Euler-Poincare reduction has to be treated more detailed, see [14][p. 119]. Nevertheless the system (2.267) is not what we desired, since the particle weight δw_n^0 stays constant over time and the control variate is only applied at the initial condition. The particles immediately decorrelate from their initial condition rendering this form of control variate useless. But any time discretization is performed by discretizing the action, a time integral

$$\int_t^{t+\Delta t} L(q, \dot{q}, \Phi, \dot{\Phi}, \tau) \mathrm{d}\tau \approx \Delta t L(q, \dot{q}, \Phi, \dot{\Phi}, 0) \tag{2.268}$$

with a quadrature rule and obtaining e.g. the symplectic Euler. The desired scheme is mostly derived for one time step, such that e.g. phase space volume is conserved during one step yielding a long term stable method when combining many of them. The critical point here is that it is safe to assume that the particles decorrelate much less in a single time step, which

means keeping the weights constant during a time step does not violate conservation laws. So the control variate weights should be updated between time steps. Phase space conservation is our most important property, which is easily kept by using a symplectic integrator during one time step. Then it is mostly argued that changing the weight δw_n in between time steps changes the phase space volume and thus ruining our discretization. But it is possible to modify the weights with an additional control variate enforcing that the overall mass $\frac{1}{N_p}\sum_n^{N_p} w_n = \frac{1}{N_p}\sum_n^{N_p} \delta w_n + \int h(v)\ \mathrm{d}v$ stays constant. But this is completely useless since phase space volume conservation is not about conserving the overall volume, but any volume. Thus it is really easy to conserve the volume with point particles and constant weight. So one could stratify the phase space and enforce the mass conservation in each stratum, but still this does not correspond to the generality of conserving any volume. Unless we do not account for the phase space volume fluctuating into the control variate in our Lagrange formalism we can just hope for the best by following the system eqn. (2.267) knowing that at least each time step itself is fine. Nevertheless fig. 2.15 shows that this is not a good long-term option. Eventually one has to include the variation of the weights. We aim to define a manifold that describes the application of a control variate $(x, v) \mapsto h(x, v)$. This approach is promising because geometric integration on manifold appears to be possible, see [107]. The main idea of the control variate is that the discrepancy between the exact value of an integral and its Monte Carlo approximation by a given set of markers is used in order to improve other Monte Carlo estimate using the same set of markers. In the standard control variate estimator this discrepancy is simply subtracted from the estimated mean, which improves the estimate. Here we seek for fields, such that this discrepancy is eliminated, which is formulated as a constraint or a manifold.

$$\begin{aligned}\Big\{x(t) = (x_1^t, \dots, x_N^t) \in \mathbb{R}^N, v(t) = (v_1^t, \dots, v_N^t) \in \mathbb{R}^N, \Phi(t) \in \mathcal{C}^1(0, L) \mid \\ \frac{1}{N}\sum_{n=1}^{N} \frac{h(x_n^t, v_n^t, t)\Phi(x_n^t, t)}{g(x_n(0), v_n(0), t=0)} = \iint h(x, v, t)\Phi(x, t)\ \mathrm{d}x\mathrm{d}v\Big\} \\ := \{x(t), v(t), \Phi(t) | \sigma(x(t), v(t)) = 0\}\end{aligned} \tag{2.269}$$

Then the set given in eqn. (2.269) is a manifold in the particle phase space $\mathbb{R}^{2N} \times \mathcal{C}(0, L)$ and can be used as a constraint on the system of ODEs arising from the Lagrangian in eqn. (2.260).

$$\sigma(x, v, \Phi) = -\frac{1}{N}\sum_{n=1}^{N} \frac{h(x_n^t, v_n^t, t)\Phi(x_n^t, t)}{g_n} + \iint h(x, v, t)\Phi(x, t)\ \mathrm{d}x\mathrm{d}v \tag{2.270}$$

The constraint is introduced into the Lagrangian L via a Lagrange multiplier λ, following [107] yielding the extended Lagrangian

$$\begin{aligned}&\mathcal{L}\left(x, v, \dot{x}, \dot{v}, \Phi, \dot{\Phi}, \lambda, \dot{\lambda}\right) := \\ &\quad \frac{1}{N}\sum_{n=1}^{N} \underbrace{\frac{f_n}{g_n}}_{=w_n} \left[x_n^t \dot{v}_n^t - \frac{1}{2}(v_n^t)^2 + \Phi(x_n^t, t)\right] + \frac{1}{2}\int |\partial_x \Phi(x, t)|^2\ \mathrm{d}x\ +\ \lambda\sigma(x, v, \Phi) \\ &\quad = \frac{1}{N}\sum_{n=1}^{N} \frac{f_n}{g_n} \left[x_n^t \dot{v}_n^t - \frac{1}{2}(v_n^t)^2 + \Phi(x_n^t, t)\right] + \frac{1}{N}\sum_{n=1}^{N} \underbrace{\frac{f_n - \lambda h(x_n^t, v_n^t, t)}{g_n}}_{:=\delta w_n^t} \Phi(x_n^t, t) \\ &\qquad + \frac{1}{2}\int |\partial_x \Phi(x, t)|^2\ \mathrm{d}x + \lambda \iint h(x, v, t)\Phi(x, t)\ \mathrm{d}x\mathrm{d}v.\end{aligned} \tag{2.271}$$

The variation in Φ already gives us the control variate estimator for the right hand side of the weak Poisson equation, which is what we desired.

$$\begin{aligned}\int \partial_x \Phi(x,t) \cdot \partial_x \delta\Phi(x)\ \mathrm{d}x =& \frac{1}{N}\sum_{n=1}^{N} \frac{f_n}{g_n}\delta\Phi(x_n^t) \\ &+ \lambda\left(\frac{1}{N}\sum_{n=1}^{N}\frac{h(x_n^t, v_n^t, t)\delta\Phi(x_n^t,t)}{g(x_n(0), v_n(0), 0)} - \iint h(x,v,t)\delta\Phi(x,t)\ \mathrm{d}x\mathrm{d}v\right) \\ =& \frac{1}{N}\sum_{n=1}^{N}\delta w_n^t \delta\Phi(x_n^t) - \iint h(x,v,t)\delta\Phi(x,t)\ \mathrm{d}x\mathrm{d}v\end{aligned} \tag{2.272}$$

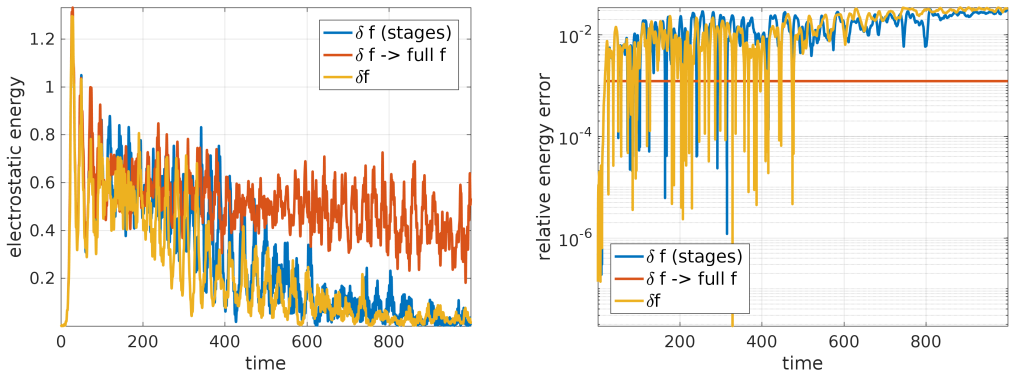
When discretizing Φ with basis functions, the constraint should be applied for every basis function in order to get a constraint for each basis function. This also implies that λ is not a scalar anymore. So far it looks good, but the corresponding equations of motion in eqn. (2.273) make no sense as they do not converge to the original system for large number of particles.

$$0 = \sigma(x, v, \Phi) \tag{2.273}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}v_n^t = \left(1 - \lambda\frac{h(x_n^t, v_n^t, t)}{f_n}\right)\partial_x\Phi(x_n^t, t) - \lambda\frac{\partial_x h(x_n^t, v_n^t, t)}{f_n}\Phi(x_n^t, t) \tag{2.274}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}x_n^t = v_n^t + \lambda\frac{\partial_v h(x_n^t, v_n^t, t)}{f_n}\Phi(x_n^t, t). \tag{2.275}$$

So eventually the complete variance minimization problem has to be incorporated in the Lagrangian or the integrator making α another free variable. Yet this extends the scope of this thesis, such that we leave this problem to professionals in geometric integration. We stress again that every scheme used should at least preserve the volume of phase space in the single particle case, because long time integration for single particles is a big difficulty in plasma physics with particles where a plethora of solutions is already available.



(a) electrostatic energy (b) relative energy error

Figure 2.15.: For a bump-on-tail instability the initial condition is an unsuitable control variate when importance sampling is used, such that it can be turned off when the variance reduction lies below a certain threshold. This corresponds to $\delta f \rightarrow full - f$, where long term stability of the electrostatic energy is recovered and the energy error stays bounded, because the integrator being used is fully symplectic once the control variate is turned off. If the control variate shall not be turned off by a threshold, the correlation coefficient will still be low. Then there are two options, keeping the δw_n constant during a stage δf (stages) resulting in a symplectic time step or changing δw_n also during every time step. Both of these options damage the long term stability to the same extent. $\epsilon = 0.03,\ k = 0.3,\ N_p = 10^4\ , \Delta t = 0.05,\ N_f = 32,\ rk3s$

2.4. Linearized Vlasov–Poisson

Let $f_0(v)$ be a steady state solution to the Vlasov–Poisson system. Then the linearization of the Vlasov–Poisson system [108], around that state $f_0(v)$ reads,

$$\partial_t f(x,v,t) + v\partial_x f(x,v,t) - \partial_x \Phi(x,t)\partial_v f_0(v) = 0 \tag{2.276}$$

$$-\partial_{xx}\Phi(x,t) = 1 - \int f(x,v,t)\ \mathrm{d}v. \tag{2.277}$$

Equation (2.276) contains a forcing term such that we cannot use the method of characteristics. Yet if we allow a weight evolution it is still possible to define equations of motions in (2.278).

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = V(t) \tag{2.278}$$

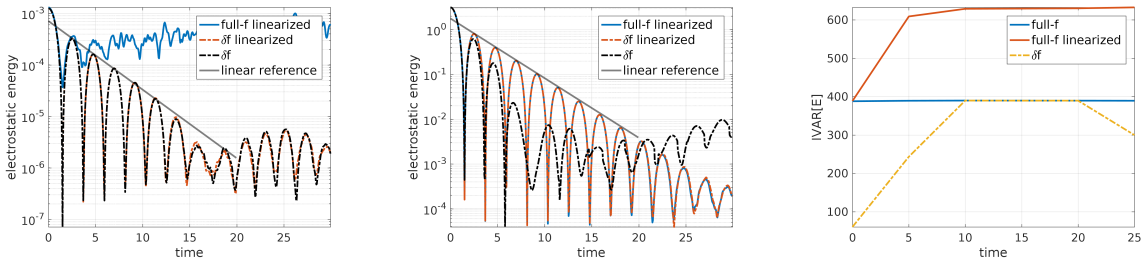
$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = 0 \tag{2.279}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} f(X(t),V(t),t) = \partial_x \Phi(X(t),t)\cdot\partial_v f_0(V(t)) \tag{2.280}$$

The velocity derivate of the equilibrium $\partial_v f_0$ is known in closed form and can just be used. In this case, the markers distributed according to g do not follow the same Vlasov equation as f, but eqn. (2.281).

$$\partial_t g(x,v,t) + \partial_x g(x,v,t) = 0. \tag{2.281}$$

Then system combining eqn. (2.278) and eqn. (2.281) exhibits a time development of the weights $w_k = \frac{f_k}{g_k}$. We can already see that the velocities of the markers stays constant in time, therefore, stratified sampling yields already a lasting variance reduction. Furthermore,



(a) Weak Landau damping (b) Strong Landau damping (c) Strong Landau damping

Figure 2.16.: Linearized Vlasov–Poisson for linear $\epsilon = 0.01$, $N_p = 10^4$ and nonlinear $\epsilon = 0.5$, $N_p = 10^5$ Landau damping in comparison to the nonlinear system. The integrated variance for electric field IVAR $[E]$ is given in order to measure the variance reduction by the linearization.($\Delta t = 0.05$, $N_f = 32$)

the number of particles can be reduced by SIR. This is also quite effective, since the system is linearized.

2.4.1. Particle noise and variance reduction

The investigation of the linearized Vlasov–Poisson system (2.276) brings helpful insight in the basic dynamics at low costs by excluding the nonlinear coupling. Thus the system is much easier to solve such that a variance reduction is expected. But an implementation of the linearization by following the equations of motion in eqn. (2.278) also changes the likelihoods f_n thus in increasing the variance of the mass. This happens independent of an additional control variate. The equations of motion (2.278) were implemented by a standard second order Runge Kutta scheme, such that the time integrator constitutes the only difference in the code. The schemes used here are full-f and δf with the same time integrator. The linearizing integrator yields then the linearized full-f and the linearized δf. Note again, that the δf method can never be worse than the full-f scheme. In fact for the nonlinear Landau damping the linearization does not reduce the integrated variance on the electric field, see fig. 2.16c. The other results are as expected, the linearization differs significantly in the nonlinear case where the additional δf does not have any effect, see fig. 2.16b. Yet the linearization cannot overcome the problem of the small amplitude $\epsilon = 0.01$ thus δf is needed for the linear Landau damping, see fig. 2.16a.

2.4.2. Dispersion relations

The main purpose of the linearization is to obtain a system that can be easily treated with analysis in order to obtain a Dispersion relation. We want to find eigenvalues of the Vlasov–Poisson system, related to eigenvalues of the Poisson equation thus we call them modes. These modes are damped and or growing over time. They can be found as complex roots of a dispersion relation $D(\omega, k)$, which depends on the initial condition used for the linearization. The complete theory is found in [109], but it is strongly recommended to start with Sonnendrückers lecture notes [110], which are much more comprehensible. Since Maxwellians play an important role the plasma dispersion function Z is mostly required, given as

$$Z(x) := \sqrt{\pi}\, \mathrm{e}^{-x^2} \left(\mathrm{i} - \operatorname{erfi}(x)\right), \tag{2.282}$$

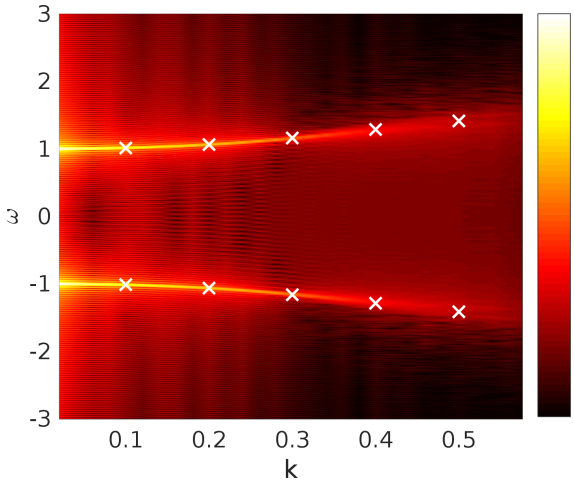
where erfi denotes the complex inverse error function. A general initial condition for the Vlasov–Poisson system is the sum of multiple Maxwellians

$$f_0(x,v) = \sum_n \frac{\alpha_n}{\sqrt{2\pi} v_{th,n}} \mathrm{e}^{-\frac{(v-v_n)^2}{2v_{th,n}^2}}, \tag{2.283}$$

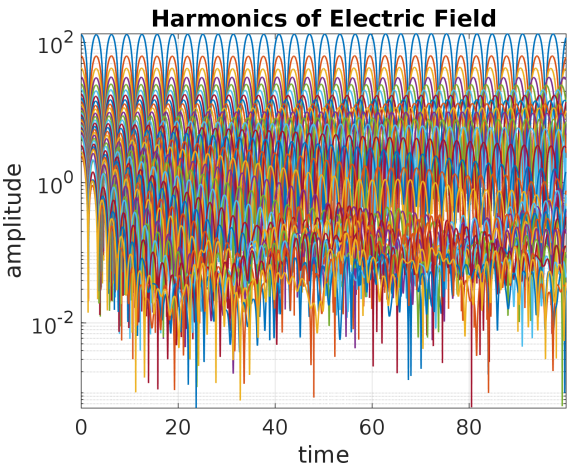
with size α_n thermal velocity $v_{th,n}$, mean velocity v_n. For the plasma frequency ω_p (in this chapter $\omega_p = 1$) the dispersion relation reads

$$D(\omega,k) = 1 + \sum_n \alpha_n \left(\frac{\omega_p}{k v_{th,n}}\right)^2 \left[1 + \frac{1}{\sqrt{2} v_{th,n}} \left(\frac{\omega}{k} - v_n\right) Z\left(\frac{1}{\sqrt{2} v_{th,n}} \left(\frac{\omega}{k} - v_n\right)\right)\right]. \tag{2.284}$$

The first approach is of course complex root search on the corresponding dispersion relation $D(\omega,k)$ which can be done efficiently e.g. by rational interpolation [111, 112] or using MATLABs variable precision arithmetic **vpasolve()**. Results from this method are used as reference values. Instead of solving the eigenvalue problem directly an analysis of the electric field as output of a short time simulation can be rewarding. For this all spatial modes in the density f are weakly excited for the linearized system and very weakly excited if no linearized model is at hand. After simulating over the linear phase one takes a time and spatial Fourier transform of the obtained electric field which yields an informative diagram, see fig. 2.17a. Examples of such wave dispersion diagrams for the Vlasov equation obtained with similar methods can be found in [113, 114, 115, 116].
In the beginning the domain length L is fixed to the smallest wave vector k_0, the longest wave length $\frac{2\pi}{k_0}$ that shall be resolved by $L = \frac{2\pi}{k_0}$. Let the discretized system allow N modes, because e.g. the electric field solver uses $2N$ cells. By spatially Fourier transform the electric field we obtain time dependent modes $\hat{E}(k,t)$ for $k \in \{k_0, 2k_0, \ldots, Nk_0\}$. Completely analog the slowest frequency is obtained by $\omega_0 = \frac{2\pi}{t_{\max}}$, which yields a the space-time Fourier transformed field $\tilde{E}(k,\omega)$ for $\omega \in \{\omega_0, 2\omega_0, \ldots, \frac{2\pi}{\Delta t}\}$. At this point it is quite common, see e.g. [116], to state that $\tilde{E}(k,\omega)$ is plotted and omitting the color axis. But this is not exactly what is done. The time Fourier transform is the origin of a lot of problems, because damped modes are absolutely not periodic which is the very reason why in the analytical derivation the Laplace transform is used. The discrete counterpart to Laplace transforming is the Z-transform. It can be obtained by Pronys method, which is the art of fitting damped modes and a rather parametric approach such that we consider it later. A remedy for the Fourier transform is to make $\hat{E}(x,t),\ t \geq 0$ periodic by using a *butterfly* by mirroring the field at the $t = 0$ axis, $\hat{E}(x,t) = \hat{E}(x,-t)$. Here zero padding of the discrete Fourier transform up to values $16 \cdot N$ is quite beneficial. This crude post-processing continues by plotting the logarithm $\log(|\tilde{E}(k,\omega)|)$, where we use mostly the squared logarithm because it damps the small noisy modes $\log(|\tilde{E}(k,\omega)|)^2$. At this point the color values are meaningless such that they can be omitted. Here the in-time filtering by Hann windows was also tried but we did not see much improvement in smearing out ω.
In PIC codes it is quite common to excite all modes by the natural noise level and then just wait until something happens. For unstable schemes one might catch then a numerical instability but in general this is a troublesome course of action since the result heavily depends on the particle number and the grid points making it hard to compare to other codes. When using the background $f_0(v)$ as control variate nothing happens because the initial condition matches perfectly. In oder to get the same results for grid-based and PIC methods we define the initial conditions by exciting all modes to a level ϵ with a non mandatory uniformly



(a) Dispersion diagram obtained by space-time Fourier transform of the electric field and additional post processing as described. White crosses indicate known roots of $D(\omega, k)$.



(b) All 32 spatial Fourier modes of the electric field. The weakly damped modes oscillate perfectly, while the strongly damped ones end in noisy behavior.

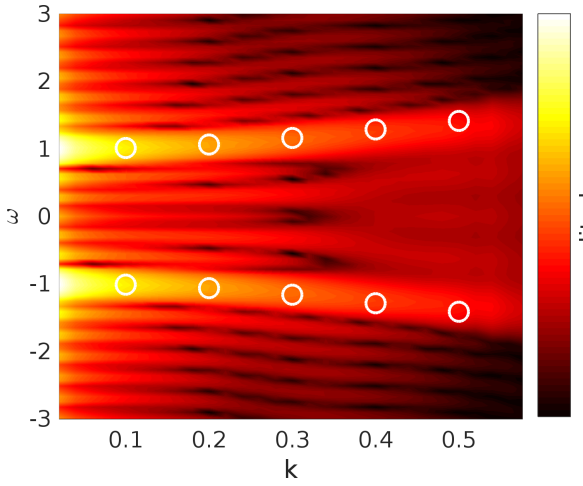
Figure 2.17.: Obtaining the dispersion relation for Langmuir waves with linearized Vlasov–Poisson δf PIC using fifth order B-splines and the parameters $k_0 = 0.02,\ \epsilon = 10^{-2},\ N_p = 10^6,\ N_f = 64,\ \Delta t = 0.02$ and $t_{\max} = 100$.

randomly drawn phase shift u.

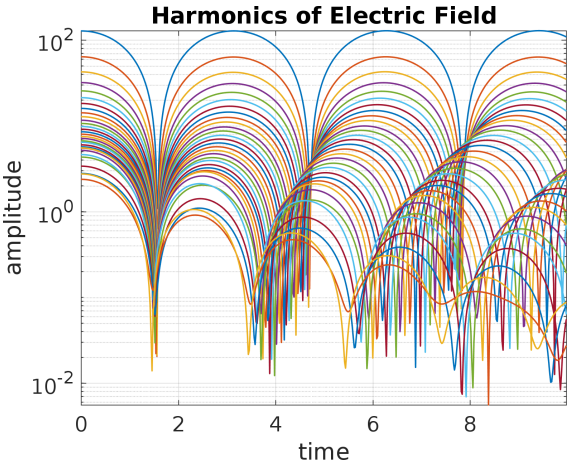
$$f(x, v, t = 0) = \left[1 + \epsilon \sum_{n=1}^{N} \cos\left(n \frac{2\pi}{L} x + u_n\right)\right] f_0(v), \quad u_i \sim \mathcal{U}(0, L), \forall i = 1, \ldots, N$$

In order to stay in the linear phase as long as possible ϵ should be chosen very small. This initial condition does not excite the exact generalized eigenvalues of the linearized system. The true eigenvalues form then after some time-steps such that the first can be disregarded in the post-processing. Most important is a large final time $t_{\max}$ such that the discrete Fourier transform can work on many oscillations. Thus the time step width is less important. High order B-splines counterfeit aliasing in the high modes. The strongly damped modes appear very weak and smeared out in fig. 2.17a. This problem is specific for PIC, because the strong damping does not continue over the entire simulation time but stops at some noise level, see fig. 2.17a. For grid based solvers it suffices to stay within the recurrence time. Here the only solution is to use either much more particles or a smaller simulation time $t_{\max}$. The latter degrades then the time Fourier transform. Bad quality spectral plots are quite common [116], therefore we seek another method. Amplitude and phase estimation based on Pronys method has been successfully applied in [117]. In general for case signals consisting only of a few damped sinusoids [118] gives a great overview of an abundance of methods for estimating the damping rate and frequency including detailed MATLAB examples. These methods are conceptually the discrete version of the time Laplace transform. For noisy signals consisting of few frequency components the matrix pencil method is quite robust [118, 119] thus we settle for this method in our parametric estimates. In the following a parametric estimate of the time frequency content of $\hat{E}(k, t)$ is made for every mode k. The parametric method may be more expensive in the post-processing but it has the enormous advantage, that results from simulating a very short time period $t_{\max}$ suffice for accurate estimates. For Langmuir waves (fig. 2.18) and the Bump-on-tail instability (fig. 2.19) few oscillations are needed to in order to reproduce the dispersion diagram. In both cases the standard spectral

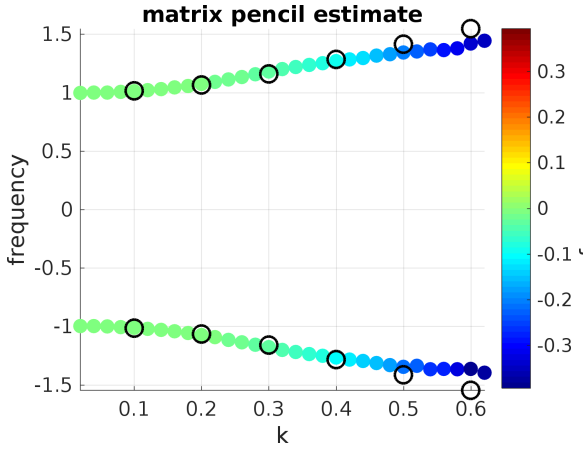
analysis returns only a vague diagram, whereas the parametric estimate is able to extract growing and damped modes for every k at the same time.



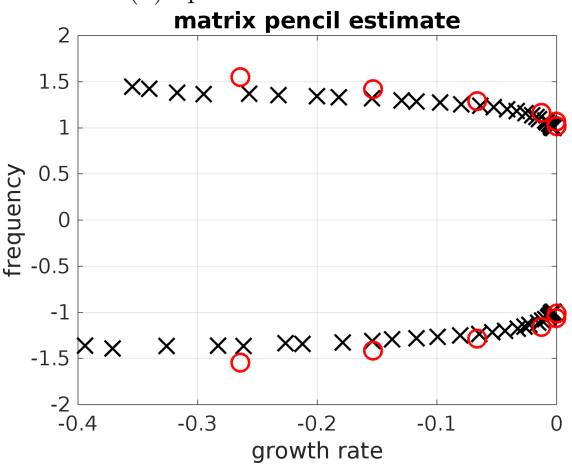
(a) Space time harmonics of E



(b) Spatial harmonics of E

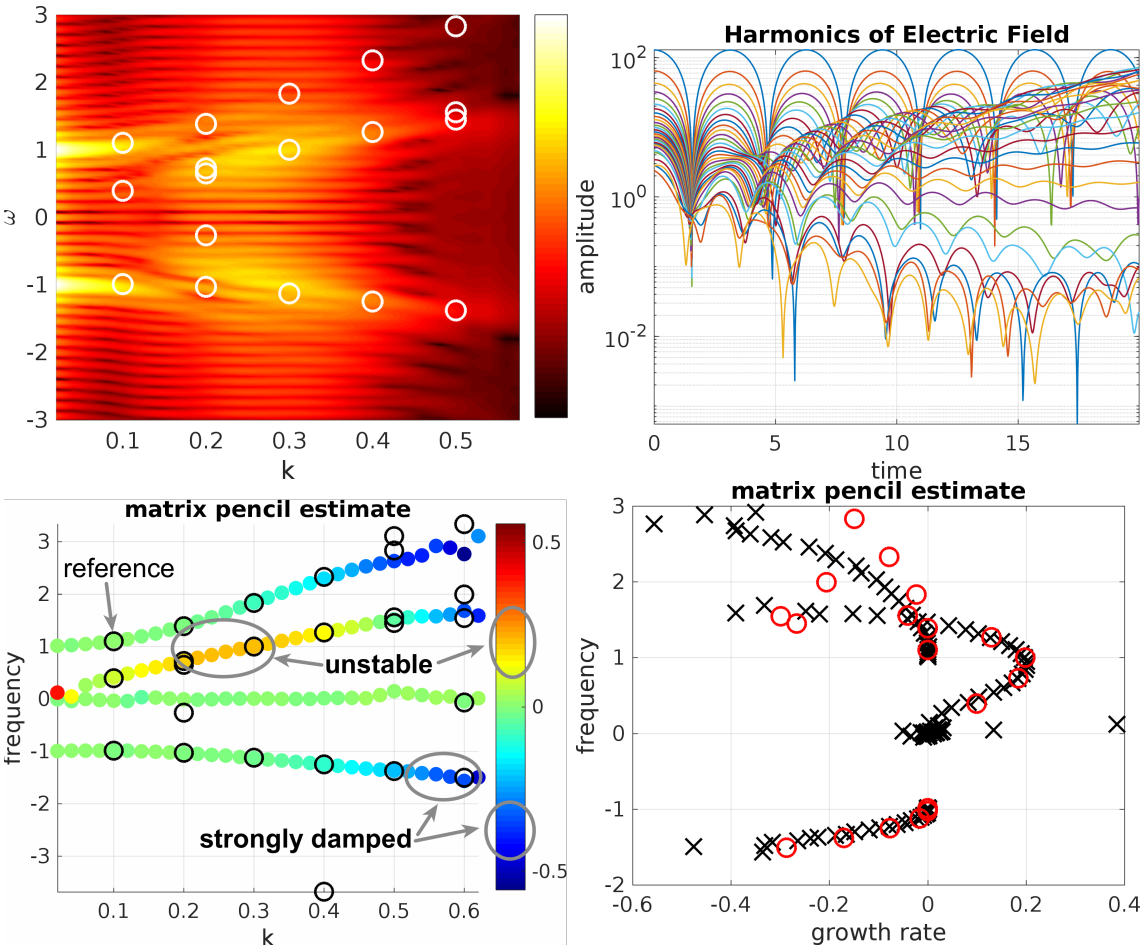


(c) Frequency content for every spatial Fourier mode of E. Color denotes growth rate.



(d) Matrix pencil estimated roots of $D(\omega, k)$.

Figure 2.18.: Estimating the dispersion relation for Langmuir waves using the matrix pencil method for short time such that only a few periods of oscillation are observed. The strongly damped modes appear clearly (blue) in the parametric estimate in contrast to the under-resolved diagram of the space time harmonics. The circles denote reference values. For large k the modes are too strongly damped for PIC which explains the small disagreement. (δf PIC using fifth order B-splines, $k_0 = 0.02$, $\epsilon = 10^{-2}$, $N_p = 10^6$, $N_f = 64$, $\Delta t = 0.02$ and $t_{\max} = 10$.)



(a) Estimated frequencies, color denotes growth rate.

(b) Estimated roots of $D(\omega, k)$.

Figure 2.19.: Estimating the dispersion relation for the Bump-on-tail instability with Langmuir waves in the background. This is difficult since it includes strongly damped and growing modes for every wave number. The two Langmuir wave branches sandwich the Bump-on-tail branch which is very unstable in the orange region $0.2 < k < 0.4$. Since no exact eigenvalue is excited a fourth branch at $\omega = 0$ accounts for the remainder and can be safely disregarded. The circles denote an incomplete set of reference values obtained by root search. Disagreement can only be seen by the modes that turn already nonlinear in the short time period and small amplitudes which PIC cannot resolve. (δf PIC using fifth order B-splines, $k_0 = 0.02,\ \epsilon = 10^{-2},\ N_p = 10^6,\ N_f = 64,\ \Delta t = 0.02$ and $t_{\max} = 20$.)

2.4.3. Conditioning by Gaussian quadrature

The constant velocity in eqn. (2.278) allows for even greater variance reduction by hybrid Monte Carlo methods. A quadrature method for V is combined with Monte Carlo samples for X. Let (X, V) be random deviates with density $g(x, v)$, then X is distributed according to the marginal density g_x.

$$g_x(x) := \int_{-\infty}^{\infty} g(x, v) \, \mathrm{d}x\mathrm{d}v. \tag{2.285}$$

The integral of the density f and a test function ψ over the velocity domain is approximated by numerical quadrature with N_v Gauss points v_j and corresponding weight w_j, see eqn. (2.286).

$$\int_{-\infty}^{\infty} f(x, v)\psi(x, v)dv \approx \sum_{j=1}^{N_v} w_j f(x, v_j)\psi(x, v_j) \text{ for } x \in [0, L] \tag{2.286}$$

Possible candidates are the Gauss–Hermite quadrature for the unbounded domain and Gauss–Legendre for a truncated domain. These are spectrally accurate methods such that the density f can be recovered from very few points v_j by the corresponding spectral interpolation rule, see [54]. Spatially independent collisions can then also be included on basis of the spectral discretization. For the Monte Carlo estimator let $X_1, X_2, \dots$ be i.i.d. according to to X, which means that identical samples are used for every v_j. The interpretation as conditional Monte Carlo is emphasized in eqn. (2.287).

$$\begin{aligned}
\int_{-\infty}^{\infty}\int_{0}^{L} f(x, v)\psi(x, v) \, \mathrm{d}x\mathrm{d}v &= \mathbb{E}\left[\frac{f(X, V)}{g(X, V)}\psi(X, V)\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{f(X, V)}{g(X, V)}\psi(X, V) \mid V\right]\right] \\
&= \int_{-\infty}^{\infty} \mathbb{E}\left[\frac{f(X, V)}{g(X, V)}\psi(X, V) \mid \{V = v\}\right] \, \mathrm{d}v \\
&= \int_{-\infty}^{\infty} \mathbb{E}\left[\frac{f(X, v)}{g(X, v)}\psi(X, v)\right] \, \mathrm{d}v = \int_{-\infty}^{\infty} \mathbb{E}\left[\frac{f(X, v)}{g_x(X)}\psi(X, v)\right] \, \mathrm{d}v \\
&\approx \sum_{j=1}^{N_v} w_j \, \mathbb{E}\left[\frac{f(X, v_j)}{g_x(X)}\psi(X, v_j)\right] \\
&\approx \sum_{j=1}^{N_v} w_j \, \frac{1}{N_p}\sum_{n=1}^{N_p} \, \frac{f(X_n, v_j)}{g_x(X_n)}\psi(X_n, v_j)
\end{aligned} \tag{2.287}$$

The variance of the basic estimator in eqn. (2.287) is given in eqn. (2.288) and includes covariances, which can be negative thus reducing the overall variance.

$$\begin{aligned}
&\mathbb{V}\left[\sum_{j=1}^{N_v} w_j \, \frac{1}{N_p}\sum_{n=1}^{N_p} \, \frac{f(X_n, v_j)}{g_x(X_n)}\psi(X_n, v_j)\right] \\
&\qquad = \sum_{i=1}^{N_v}\sum_{j=1}^{N_v} \frac{w_i w_j}{N_p} \mathbb{COV}\left[\frac{f(X, v_i)}{g_x(X)}\psi(X, v_i), \frac{f(X, v_j)}{g_x(X)}\psi(X, v_j)\right]
\end{aligned} \tag{2.288}$$

But here the evolution of the particles depends on corresponding the velocity v_j thus independent particles for all velocity Gauss points $(v_j)_{j=1,\dots,N_v}$ are needed. The number of markers can vary over the different quadrature nodes yielding more flexibility for variance reduction. In the following $X_{1,j}, \dots, X_{N_{p,j},j}$ denote the markers belonging to a Gauss point v_j which are i.i.d.$\sim X_j$, where X_j is distributed according to $g_{x,j}(x)$. All markers are drawn

independently and the sampling distribution $g_{x,j}$ as well as number of markers $N_{p,j}$ can be chosen arbitrarily. The overall number of samples, directly proportional to the computational costs, is $N_p = \sum_{j=1}^{N_v} N_{p,j}$.

$$\begin{aligned}\int_{-\infty}^{\infty}\int_{0}^{L} f(x,v)\psi(x,v)\ \mathrm{d}x\mathrm{d}v &\approx \sum_{j=1}^{N_v} w_j\ \mathbb{E}\left[\frac{f(X_j,v_j)}{g_{x,j}(X_j)}\psi(X_j,v_j)\right] \\ &\approx \underbrace{\sum_{j=1}^{N_v} w_j\ \frac{1}{N_{p,j}}\sum_{n=1}^{N_{p,j}}\ \frac{f(X_{n,j},v_j)}{g_{x,j}(X_{n,j})}\psi(X_{n,j},v_j)}_{:=\mathcal{I}(f\cdot\psi)} \end{aligned} \tag{2.289}$$

The overall variance of (2.289) in eqn. (2.290) can be reduced by adapting the number of samples.

$$\mathbb{V}\left[\mathcal{I}(f\cdot\psi)\right] = \sum_{j=1}^{N_v}\frac{w_j^2}{N_{p,j}}\mathbb{V}\left[\frac{f(X_j,v_j)}{g_{x,j}(X_j)}\psi(X_j,v_j)\right] \tag{2.290}$$

The straightforward variance reduction is to choose $g_{x,j}(x)$ close to $f(x,v_j)$. It is most naturally to take the number of particles $N_{p,j}$ for each quadrature point v_j proportional to the corresponding normalizing constant.

$$N_{p,j} \sim (w_j\ \gamma_j)^2, \quad \gamma_j := \int_0^L f(x,v_j)\ \mathrm{d}x \tag{2.291}$$

This keeps the absolute error on the same level, yet we are more interested in the relative error also known as the coefficient of variance, such that

$$N_{p,j} \ \sim (w_j\ \gamma_j)$$

is a better rule of thumb. Unfortunately none of these quadrature rules guarantee convergence for $N_p \to \infty$ and a fixed N_v. We encountered this situation already in the discussion of the gyroaverage, which combined periodic quadrature with Monte Carlo. The problem was resolved by introduction of a random shift onto the quadrature nodes. The same can be done in non-periodic domains and is again basically stratification with only one particle per stratum. The sample particles can be drawn according to eqn. (2.292), which guarantees a variance reduction but not necessarily a gain in efficiency.

$$\begin{aligned} V_{n,j} &= \left(\frac{j-U_n}{N_v}\right)(v_{\max}-v_{\min})+v_{\min}, \quad j=1,\dots,N_v \\ U_n &\sim \mathcal{U}(0,1)\ i.i.d \text{ for all } n=1,\dots,N_p \\ w_{n,j} &= \frac{(v_{\max}-v_{\min})}{N_v} \\ X_{n,j} &\sim X \end{aligned} \tag{2.292}$$

When interpreted as a randomly shifted Riemann sum the Monte Carlo estimator for the samples obtained from eqn. (2.292) yields the quite unattractive convergence rate of $\mathcal{O}(N_v)$ for $N_v \to \infty$, although the rate for small N_v might be much higher. Up to now it is unclear how to randomize the Gauss–Hermite quadrature with its attractive convergence rate. We suspect that high order Quasi Monte Carlo techniques with an interlacing factor in the velocity dimension can yield much better convergence rates, which is demonstrated for Gaussians in [84]. The variance in V being much larger than the variance in X can be denoted by

$$\mathbb{E}\left[\mathbb{V}\left[\frac{f(X,V)}{g(X,V)}\psi(X,V)\mid X\right]\right] \gg \mathbb{E}\left[\mathbb{V}\left[\frac{f(X,V)}{g(X,V)}\psi(X,V)\mid V\right]\right]. \tag{2.293}$$

In this case the conditioning (2.289) becomes a very efficient replacement for the Monte Carlo estimator. But when estimating the electric field for Vlasov–Poisson with importance sampling the weights reduce to a constant C and the test-function is only spatially dependent. By the law of total variance we can see that the above criteria is not met in the latter case:

$$\mathbb{E}\left[\mathbb{V}\left[\frac{f(X,V)}{g(X,V)}\psi(X) \mid X\right]\right] = \mathbb{E}\left[\mathbb{V}\left[C\psi(X) \mid X\right]\right] = C^2\mathbb{E}\left[\mathbb{V}\left[\psi(X) \mid X\right]\right] = 0, \tag{2.294}$$

$$\mathbb{E}\left[\mathbb{V}\left[\frac{f(X,V)}{g(X,V)}\psi(X) \mid V\right]\right] = C^2\Big(\mathbb{V}\left[\psi(X)\right] - \mathbb{E}\left[\mathbb{V}\left[\psi(X)|V\right]\right]\Big). \tag{2.295}$$

We conclude that the linearized Vlasov–Ampère is a better candidate for conditioning. Nevertheless these are only expectations such that improvements might still be possible.
Note that this is also possible for the three dimensional linearized Vlasov–Maxwell system without external magnetic field B_0. A homogeneous constant magnetic field B_0 leads via the Lorenz force $v \times B_0$ to the rotating gyromotion changing the velocity coordinate v. But the part of the velocity v parallel to the magnetic field $v_\parallel \parallel B_0$ stays untouched thus allowing one dimensional quadrature rule along $v_\parallel$. Gyrokinetic theory supposes the existence of a coordinate transformation - near identity transform - in order to transform the six dimensional system in a coordinate system such that one dimension is constant over space and time yielding a reduction from six to five dimensions. Then it turns out that the fifth dimension, the magnetic moment, stays constant over time also in the nonlinear case, which is our entry point to massive variance reduction. There is an abundance of literature concerning gyrokinetic theory available targeted mostly to physicists how are already familiar with the topic, such that we recommend [26] for readers of this work.
To our surprise neither the gyrokinetic particle codes (ORB [71], EUTERPE [120]) nor the Eulerian (GENE, GKW [121], AstroGK [122]) or Semi-Lagrangian (GYSELA [123]) solvers employ a rapidly converging quadrature rule like Gauss–Hermite onto the μ component in gyrokinetic theory, thus on could probably improve both Lagrangian and Eulerian codes.
For the conditioning of the linearized Vlasov–Poisson system Sobol numbers are used in x and different Gaussian quadrature rules in v. The Quasi Monte Carlo sequence is chosen, because it is trivial to improve over the standard Monte Carlo by much simpler methods such as stratification. Scatter plots of the density for the standard Sobol numbers, the Gauss–Legendre and Gauss–Hermite quadrature as well as the randomized midpoint rule are found in fig. 2.20. For Gauss–Hermite a second alternative is also shown, where the proportional allocation for each quadrature node is used. The recurrence phenomenon observed in fig. 2.22 stems from the mesh based representation. Since the grid in v is deterministic and the Poisson equation linear it is possible to calculate the integrated variance of the electric field on each grid point. This shows the need for that for the Bump-on-tail the region between the bump and the Maxwellian needs more particles. For the Landau the Gauss–Hermite proportional allocation yields improvements, although the outliers should have been treated separately.
We already know from theory that Vlasov–Poisson is not a good candidate to demonstrate this method, which is also underlined by fig. 2.21. Yet in practice the δf method is used, such that better results can be achieved for linear Landau damping and the Bump-on-tail instability, see fig. 2.22.

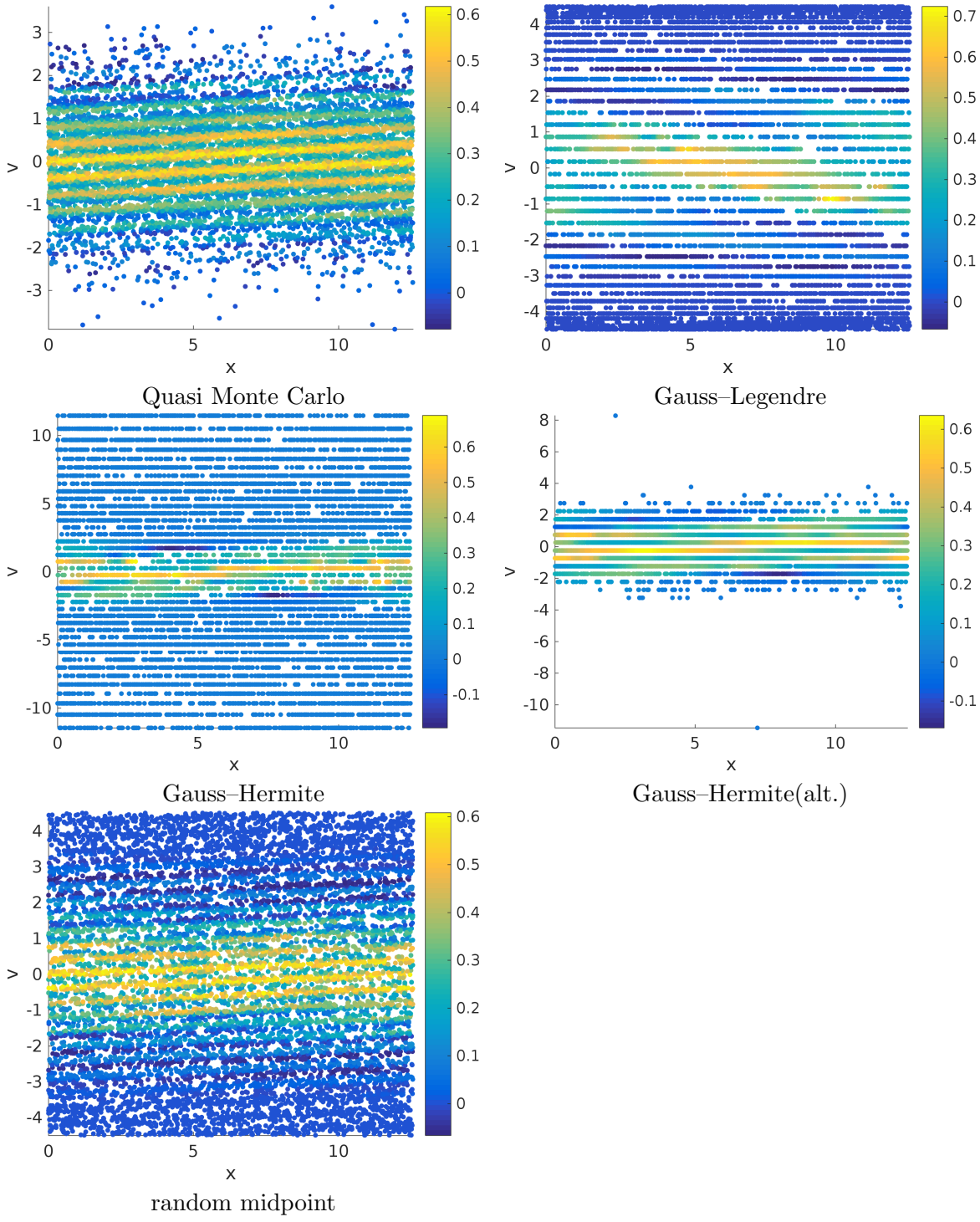


Figure 2.20.: Lagrangian particles transporting the color of the initial condition for strong Landau damping at $t_{\max}$ using conditional Monte Carlo. For QMC and the random midpoint rule the filamentation caused by the Landau damping can be clearly seen. For the Gauss methods the particles are obviously randomly distributed in x and deterministic in v. Contrary to the other Gauss methods the random midpoint appears to be non-deterministic and does not suffer from the recurrence. Gauss–Legendre, Gauss–Hermite and the midpoint method seem waste particles on areas where the density f is small (blue). For Gauss-Legendre this is extreme as many particles accumulate at $v \approx \pm 4$. In Gauss–Hermite (alt.) this problem is circumvented by proportional allocation but the recurrence remains such that the filamentations are not visible anymore.

	QMC	Gauss–Legendre	G.-Hermite	G.-Hermite(alt.)	Midpoint
Weak Landau	4.009e-06	3.591e-06	3.768e-06	2.378e-06	2.739e-06
Strong Landau	0.02539	0.06531	0.1143	0.02611	0.04467
Bump-on-tail	4.848	90.81	59.26	256.3	53.59

Figure 2.21.: Integrated variances of the electric field at $t_{\max}$ for the linearized Vlasov–Poisson and a constant total number of particles. The variance increases by the conditioning implying that the method fails to improve the QMC estimate.

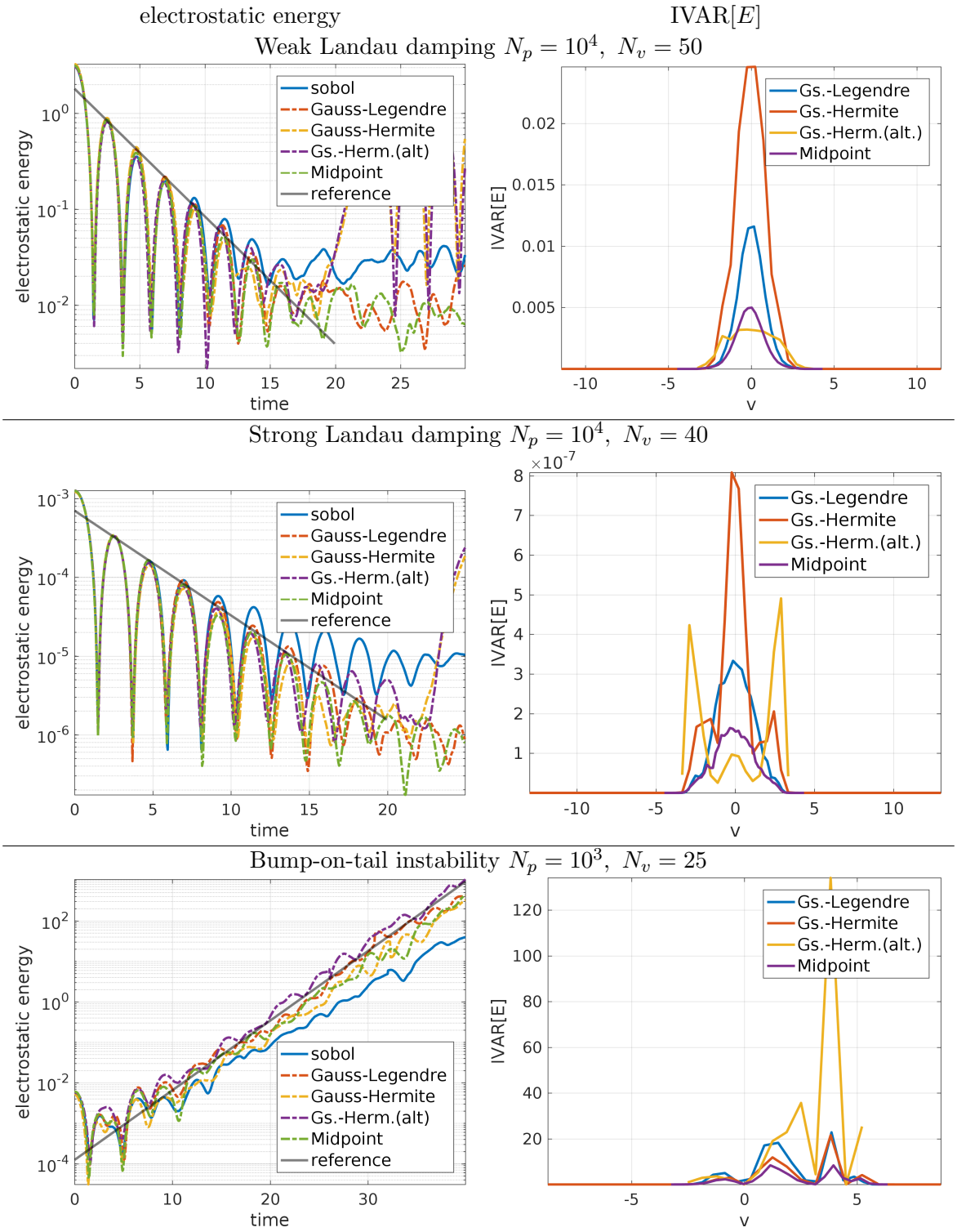


Figure 2.22.: Linearized Vlasov–Poisson with conditioning in the velocity domain. The proportional allocation for Gauss–Hermite quadrature successfully reduces variance although it leads to spiking variance estimates for $N_{p,j} < 5$.

2.5. Fokker–Planck collisions

To implement collisions in the Vlasov–Poisson model we follow [15] and introduce the Fokker–Planck equation (2.296),

$$\frac{\partial f(x,v,t)}{\partial t} = \theta \frac{\partial}{\partial v}\left((v-\mu) f(x,v,t)\right) + \frac{\sigma^2}{2}\frac{\partial^2 f(x,v,t)}{\partial v^2} \tag{2.296}$$

with parameters $\theta(x)$, $\mu(x)$ and $\sigma(x)$. In the literature one often finds the diffusion coefficient $D = \frac{\sigma^2}{2}$ and $\gamma = \theta$. The equilibrium solution is

$$f_{eq}(x,v,t) = \sqrt{\frac{\theta(x)}{\pi\sigma(x)^2}}\, \mathrm{e}^{-\theta\frac{(v-\mu(x))^2}{\sigma(x)^2}} = \frac{1}{\sqrt{2\pi}\frac{\sigma(x)}{\sqrt{2\theta}}}\, \mathrm{e}^{-\frac{1}{2}\left(\frac{v-\mu(x)}{\frac{\sigma(x)}{\sqrt{2\theta}}}\right)^2} = \sqrt{\frac{\theta}{2\pi D}}\, \mathrm{e}^{-\frac{1}{2}\frac{\theta(v-\mu(x))^2}{D}}. \tag{2.297}$$

Note that we are only interested in a solution for $t \in [0, \Delta t]$ since we use (2.296) in combination with the Vlasov equation (2.1). The Vlasov–Fokker–Planck equation reads

$$\frac{\partial f}{\partial t} + \underbrace{v\cdot\nabla_x f - (E + v\times B)\cdot\nabla_v f}_{\text{advection}} = \underbrace{\nabla_v\cdot\left[\theta(v-\mu)f - \frac{\sigma^2}{2}\nabla_v f\right]}_{\text{collisions}}. \tag{2.298}$$

For the purpose of implementation we are only interested to solve (2.296) during a splitting step $t \in [0, \Delta t]$. Let $g(x,v,t)$ be a probability distribution that solves the same Fokker–Planck equation as f

$$\frac{\partial g(x,v,t)}{\partial t} = \theta \frac{\partial}{\partial v}\left((v-\mu) g(x,v,t)\right) + \frac{\sigma^2}{2}\frac{\partial^2 g(x,v,t)}{\partial v^2}. \tag{2.299}$$

Let $(x_k(t=0), v_k(t=0))$ be i.i.d. according to $g(t=0,x,v)$ for $k = 1,\ldots,N_p$. The corresponding stochastic differential equation to eqn. (2.299) is an Ornstein–Uhlenbeck process (2.300), which is one way of solving the Fokker–Planck equation [124][p. 99].

$$\mathrm{d}V_t = \theta\left(\mu - V_t\right)\mathrm{d}t + \sigma \mathrm{d}W_t \tag{2.300}$$

The time development of the stochastic process for the Vlasov equation was prescribed by the characteristics (2.6), which we can add now to (2.300) to obtain the process corresponding to the Vlasov–Fokker–Planck equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}X(t) = V(t) \tag{2.301}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}V(t) = -\left(E(t,X(t)) + V(t)\times B(t,X(t))\right) + \theta\left(\mu - V(t)\right) + \sigma\mathrm{d}W_t \tag{2.302}$$

We define a time discretization along time steps Δt and change the standard notation to

$$x_n^t := x_n(t\Delta t),\ v_n^t := v_n(t\Delta t), \quad t \in \mathbb{N}. \tag{2.303}$$

2.5.1. Integrating the Ornstein–Uhlenbeck process

Since the transition probability of the Ornstein–Uhlenbeck process is known, we can integrate eqn. (2.300) exactly. The transition probability (2.304) from a state v' at time t' to a state v at time t' for $t > t'$ is adapted from [124][p. 100].

$$P\left(v,t,v',t'\right) = \sqrt{\frac{\theta}{2\pi D\left(1-\mathrm{e}^{-2\theta(t-t')}\right)}}\exp\left[-\theta\frac{\left(v-\mu\left(1-\mathrm{e}^{-\theta(t-t')}\right)-v'\mathrm{e}^{-\theta(t-t')}\right)^2}{2D\left(1-\mathrm{e}^{-2\theta(t-t')}\right)}\right] \tag{2.304}$$

One also obtains the equilibrium distribution in the limit of large time.

$$\lim_{t\to\infty} P\left(v,t,v',t'\right) = f_{eq}(v) \tag{2.305}$$

Knowing the transition probability allows us to set up a stochastic process by sampling this transition as

$$\begin{aligned} V_t &= V_0\mathrm{e}^{-\theta t} + \mu\left(1-\mathrm{e}^{-\theta t}\right) + \frac{\sigma\mathrm{e}^{-\theta t}}{\sqrt{2\theta}} W_{\mathrm{e}^{2\theta t}-1} \\ &= V_0\mathrm{e}^{-\theta t} + \mu\left(1-\mathrm{e}^{-\theta t}\right) + \frac{\sigma}{\sqrt{2\theta}} W_{1-\mathrm{e}^{-2\theta t}} \\ &= V_0\mathrm{e}^{-\theta t} + \mu\left(1-\mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} W_{1-\mathrm{e}^{-2\theta t}}, \end{aligned} \tag{2.306}$$

where $W_t \sim \mathcal{N}(0,t)$ denotes a standard Wiener process. Resolving eqn. (2.306) for V_0 yields a backward propagation in eqn. (2.307).

$$\begin{aligned} V_0 &= V_t\mathrm{e}^{\theta t} + \mu\left(1-\mathrm{e}^{-\theta t}\right) - \mathrm{e}^{\theta t}\sqrt{\frac{D}{\theta}} W_{1-\mathrm{e}^{-2\theta t}} \\ &= V_t\mathrm{e}^{\theta t} + \mu\left(1-\mathrm{e}^{\theta t}\right) \underbrace{-}_{=\pm} \sqrt{\frac{D}{\theta}} W_{\mathrm{e}^{2\theta t}-1} \end{aligned} \tag{2.307}$$

Note that the Wiener process is symmetric, such that the change of sign is irrelevant. For the backward equation (2.307) the transition probability is be described for $t < t'$ by eqn. (2.308).

$$P\left(v,t|v',t'\right) = \sqrt{\frac{\theta}{2\pi D\left(\mathrm{e}^{2\theta(t'-t)}-1\right)}} \exp\left[-\theta\frac{\left(v-\mu\left(1-\mathrm{e}^{\theta(t'-t)}\right)-v'\mathrm{e}^{\theta(t'-t)}\right)^2}{2D\left(\mathrm{e}^{2\theta(t'-t)}-1\right)}\right] \tag{2.308}$$

The long time limit of the backward transition probability (2.308) is obtained in eqn. (2.309).

$$\begin{aligned} \lim_{t'\to\infty} P\left(v,0|v',t'\right) &= \lim_{t'\to\infty} \sqrt{\frac{\theta}{2\pi D\left(\mathrm{e}^{2\theta t'}-1\right)}} \exp\left[-\theta\frac{\left(v-\mu\left(1-\mathrm{e}^{\theta t'}\right)-v'\mathrm{e}^{\theta t'}\right)^2}{2D\left(\mathrm{e}^{2\theta t'}-1\right)}\right] \\ &= \lim_{t'\to\infty} \underbrace{\sqrt{\frac{\theta\mathrm{e}^{-2\theta t'}}{2\pi D\left(1-\mathrm{e}^{-2\theta t'}\right)}}}_{\to 0} \underbrace{\exp\left[-\theta\frac{\left(v\mathrm{e}^{-\theta t'}-\mu\left(\mathrm{e}^{-\theta t'}-1\right)-v'\right)^2}{2D\left(1-\mathrm{e}^{-2\theta t'}\right)}\right]}_{\to\exp\left[-\theta\frac{(v'-\mu)^2}{2D}\right]} = 0 \end{aligned} \tag{2.309}$$

In the semi-discretization of eqn. (2.306) we substitute the Wiener process by a normally distributed random variable $U \sim \mathcal{N}(0,1)$ which reads

$$V_t = V_0\mathrm{e}^{-\theta t} + \mu\left(1-\mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}}\sqrt{1-\mathrm{e}^{-2\theta t}}U. \tag{2.310}$$

Here V_t is just linear combination of the two random deviates V_0 and U. In the following two ways are shown, how to use the samples V_t for Monte Carlo integration of an arbitrary function h. For this the integral over h is rewritten in two different ways using that the

probability densities are normalized. In eqn. (2.311) the probability density of V_0 is denoted by $v_0 \mapsto g(v_0, t=0)$.

$$\begin{aligned}
&\int_{\mathbb{R}} h(v_t)\ \mathrm{d}v_t \\
&\qquad = \int_{\mathbb{R}} h(v_t)\ \mathrm{d}v_t \underbrace{\int_{\mathbb{R}} \frac{1}{2\pi} \mathrm{e}^{-\frac{u^2}{2}}\ \mathrm{d}u}_{=1} = \int_{\mathbb{R}}\int_{\mathbb{R}} h(v_t) \frac{1}{2\pi} \mathrm{e}^{-\frac{u^2}{2}}\ \mathrm{d}v_t \mathrm{d}u \\
&\qquad = \int_{\mathbb{R}} h(v_t)\ \mathrm{d}v_t \underbrace{\int_{\mathbb{R}} g(v_0, t=0)\ \mathrm{d}v_0}_{=1} = \int_{\mathbb{R}}\int_{\mathbb{R}} h(v_t) g(v_0, 0)\ \mathrm{d}v_t \mathrm{d}v_0
\end{aligned} \tag{2.311}$$

There are two possibilities for a change of variable that substitutes v_t yielding only an integral over v_0 and u, see eqn. (2.312).

$$\begin{aligned}
v_t &= v_0 \mathrm{e}^{-\theta t} + \mu\left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} u \\
\mathrm{d}v_t &= \mathrm{e}^{-\theta t}\ \mathrm{d}v_0 \\
\mathrm{d}v_t &= \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}}\ \mathrm{d}u
\end{aligned} \tag{2.312}$$

The first line of eqn. (2.311) yields an expectation, where the probability density of the diffusion is completely eliminated.

$$\begin{aligned}
&\int_{\mathbb{R}}\int_{\mathbb{R}} h\big(\underbrace{v_0 \mathrm{e}^{-\theta t} + \mu\left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} u}_{= v_t(v_0, u)} \big) \frac{1}{2\pi} \mathrm{e}^{-\frac{u^2}{2}} \mathrm{e}^{-\theta t} \mathrm{d}v_0 \mathrm{d}u \\
&= \mathbb{E}\left[\frac{h(V_0 \mathrm{e}^{-\theta t} + \mu\left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} U)}{g(V_0, t=0) \frac{1}{2\pi} \mathrm{e}^{-\frac{U^2}{2}}} \frac{1}{2\pi} \mathrm{e}^{-\frac{U^2}{2}} \mathrm{e}^{-\theta t} \right] = \mathbb{E}\left[\frac{h(V_t)}{g(V_0, t=0) \mathrm{e}^{\theta t}} \right]
\end{aligned} \tag{2.313}$$

This estimator is suited for weak collisions. Substitution of v_t in the second line of eqn. (2.311) yields an expectation in eqn. (2.314) which is entirely independent of g and therefore, suited for strong collisions.

$$\begin{aligned}
&\int_{\mathbb{R}}\int_{\mathbb{R}} h\big(\underbrace{v_0 \mathrm{e}^{-\theta t} + \mu\left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} u}_{= v_t(u, v_0)} \big) \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} g(v_0, 0)\ \mathrm{d}u \mathrm{d}v_0 \\
&\qquad = \mathbb{E}\left[\frac{h(V_0 \mathrm{e}^{-\theta t} + \mu\left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} U)}{g(V_0, t=0) \frac{1}{2\pi} \mathrm{e}^{-\frac{U^2}{2}}} g(V_0, 0) \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} \right] \\
&\qquad\qquad = \mathbb{E}\left[\frac{h(V_t)}{\frac{1}{2\pi} \mathrm{e}^{-\frac{U^2}{2}} \sqrt{\frac{\theta}{D(1 - \mathrm{e}^{-2\theta t})}}} \right]
\end{aligned} \tag{2.314}$$

Applying eqn. (2.306) to the particles yields

$$\begin{aligned}
v_n^{t+\Delta t} &= v_n^t \mathrm{e}^{-\theta(x_n^t)\Delta t} + \mu(x_n^t)\left(1 - \mathrm{e}^{-\theta(x_n^t)\Delta t}\right) + \frac{\sigma \mathrm{e}^{-\theta(x_n^t)\Delta t}}{\sqrt{2\theta(x_n^t)}} \sqrt{\mathrm{e}^{2\theta(x_n^t)\Delta t} - 1} u_n^t \\
&= \mu(x_n^t) + \left(v_n^t - \mu(x_n^t)\right) \mathrm{e}^{-\theta(x_n^t)\Delta t} + \frac{\sigma(x_n^t)}{\sqrt{2\theta(x_n^t)}} \sqrt{1 - \mathrm{e}^{-2\theta(x_n^t)\Delta t}}\ u_n^t
\end{aligned} \tag{2.315}$$

where $u_n^t \sim \mathcal{N}(0,1)$ i.i.d. for all n, l.

2.5.2. Likelihood integration

For the Vlasov equation we know that the values of the distribution function stay constant along the characteristics. Thus it suffices to define the sampling likelihood $g_n^0 = g(0, x_n(0), v_n(0))$ and the distribution likelihood $f_n^0 = g(0, x_n(0), v_n(0))$. With the Fokker–Planck equation $g(t, x_n(t), v_n(t))$ is not anymore a constant of time, but still the ratio $\frac{f(t,x_n(t),v_n(t))}{g(t,x_n(t),v_n(t))}$ is.
With the transition probability (2.304) we can calculate the evolution of the likelihoods by

$$\begin{aligned} f(x,v,t) &= \int_{\mathbb{R}} f(x,v',0) P\left(v,t,v',0\right) \mathrm{d}v' \\ &= \int_{\mathbb{R}} f(x,v',0) \sqrt{\frac{\theta}{2\pi D\left(1-\mathrm{e}^{-2\theta t}\right)}} \exp\left[-\frac{\theta\left(v-\mu\left(1-\mathrm{e}^{-\theta t}\right)-v'\mathrm{e}^{-\theta t}\right)^2}{2D\left(1-\mathrm{e}^{-2\theta t}\right)}\right] \mathrm{d}v'. \end{aligned} \tag{2.316}$$

Inserting the Klimontovich density yields

$$\begin{aligned} f(x,v,t) &= \frac{1}{N_p}\sum_{n=1}^{N_p} \delta(x-x_n^0) P\left(v,t,v_n^0,0\right) w_n \\ &= \frac{1}{N_p}\sum_{n=1}^{N_p} \delta(x-x_n^0) w_n \sqrt{\frac{\theta}{2\pi D\left(1-\mathrm{e}^{-2\theta t}\right)}} \exp\left[-\theta\frac{\left(v-\mu\left(1-\mathrm{e}^{-\theta t}\right)-v_n^0\mathrm{e}^{-\theta t}\right)^2}{2D\left(1-\mathrm{e}^{-2\theta t}\right)}\right]. \end{aligned} \tag{2.317}$$

After the particle has been redrawn according to the Ornstein–Uhlenbeck process the spatial coordinate did not change, therefore, $x_n^t = x_n^0$. The updated likelihoods read

$$\begin{aligned} f_n^t := f(x_n^t, v_n^t, t) &:= \frac{1}{N_p} P\left(v_n^t, t, v_n^0, 0\right) w_n^0 \\ &= \frac{f_n^0}{g_n^0}\frac{1}{N_p}\sqrt{\frac{\theta}{2\pi D\left(1-\mathrm{e}^{-2\theta t}\right)}} \exp\left[-\theta\frac{\left(v_n^t-\mu\left(1-\mathrm{e}^{-\theta t}\right)-v_n^0\mathrm{e}^{-\theta t}\right)^2}{2D\left(1-\mathrm{e}^{-2\theta t}\right)}\right]. \end{aligned} \tag{2.318}$$

The $\frac{1}{N_p}$ term causes the likelihoods to peak, which is an unnatural behavior. Therefore, we introduce an additional smoothing kernel K_h in spatial direction.

$$\begin{aligned} f_n^t := f(x_n^t, v_n^t, t) &:= \frac{1}{N_p}\sum_{m=1}^{N_p} K_h\left(x_n^t - x_m^0\right) P\left(v_n^t, t, v_m^0, 0\right) w_m \\ &= \sqrt{\frac{\theta}{2\pi D\left(1-\mathrm{e}^{-2\theta t}\right)}}\frac{1}{N_p}\sum_{m=1}^{N_p} w_m K_h\left(x_n^t - x_m^0\right) \exp\left[-\theta\frac{\left(v_n^t-\mu\left(1-\mathrm{e}^{-\theta t}\right)-v_m^0\mathrm{e}^{-\theta t}\right)^2}{2D\left(1-\mathrm{e}^{-2\theta t}\right)}\right] \end{aligned} \tag{2.319}$$

Since we are working with the Klimontovich density, this is, of course, only an estimator for the value of the density. The problematic point here is the sum over all particles, for all particles, for every collision step yielding costs $\mathcal{O}(N_p^2)$. The value of the weight is smoothed with a Gaussian in every collision step. This already corresponds to a N-body problem formulation, where the costs can be reduced to $\mathcal{O}(N_p)$ by the fast multipole method.
Let us first sketch another approach originating from the δf method applied onto the gyrokinetic Vlasov–Fokker–Planck system [125]. The equilibrium solution f_{eq} to the Fokker–Planck equation stays constant over every time step of the splitting. One can use the particles, which are distributed according to g and solve the Ornstein–Uhlenbeck process (2.306). The

likelihood $f_{eq}(x_n^t, v_n^t)$ of any state (x_n^t, v_n^t) can be evaluated at every time t because f_{eq} is known and does not depend on time. When the Ornstein–Uhlenbeck process (2.315) is applied on the particles, they are depending on the collision frequency θ, mildly displaced or scattered over the entire phase space. This corresponds to the evolution of the sampling density $g(x, v, t)$ under the Fokker–Planck equation. Now f_{eq} and g follow the same time evolution and therefore,

$$\frac{f_{eq}(x_n(t), v_n(t))}{g(x_n(t), v_n(t), t)} = \frac{f_{eq}(x_n(0), v_n(0))}{g(x_n(0), v_n(0), 0)} = \text{const. for all } t. \tag{2.320}$$

This allows to determine the sampling likelihood g_n^t after the collision step as

$$g(x_n(t), v_n(t), t) = g(x_n(0), v_n(0), 0) \frac{f_{eq}(x_n(t), v_n(t))}{f_{eq}(x_n(0), v_n(0))}, \tag{2.321}$$

which can also be applied to f yielding

$$f(x_n(t), v_n(t), t) = f(x_n(0), v_n(0), 0) \frac{f_{eq}(x_n(t), v_n(t))}{f_{eq}(x_n(0), v_n(0))}. \tag{2.322}$$

Essentially this is the same procedure as in [15], yet this scaling of the likelihoods has nothing to do with the control variate. There is no control variate present during the collision step, not here and also not in [15]. A priori the control variate is independent of the equilibrium of the collision operator. For strong collisions or long times, the solution will tend to a certain equilibrium, therefore, the equilibrium can serve as a suitable control variate. Of course, f_n^t is now a likelihood for a specific particle and does not necessarily represent the distribution function well at that point. Although eqn. (2.313) and eqn. (2.314) are the unbiased and therefore correct likelihoods it is mostly better to use eqn. (2.322) instead, because it can be interpreted as conditioned Monte Carlo estimator on the sigma algebra

$$\left\{ f(X(t), V(t), t) = f(X(0), V(0), 0) \frac{f_{eq}(X(t), V(t))}{f_{eq}(X(0), V(0))} \right\}. \tag{2.323}$$

In [125] and [15] coarse graining is used to smear out the f_n^t over the particles, which is effective once the solution is close to the equilibrium. Since the equilibrium is a long term solution, the propagation described in eqn. (2.322) is a feasible choice for $t \to \infty$, yet this needs some further rigorous explanation.
We try a motivation by Bayes' theorem (2.324), which gives a relation for the probability $\mathcal{P}$ of two events A, B and their mixed conditional probabilities.

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(B|A)P(A)}{P(B)} \;\Rightarrow\; P(B) = \frac{\mathcal{P}(B|A)}{\mathcal{P}(A|B)}\, P(A) \tag{2.324}$$

Since a marker (X^t, V^t) subject to the Ornstein–Uhlenbeck process is a random deviate distributed according to the probability density f, we can identify the probabilities of observing this marker as

$$\mathcal{P}\left(\{V^t = v \wedge X^t = x\}\right) = f(x, v, t) \;\text{ and }\; \mathcal{P}\left(\{V^0 = v' \wedge X^0 = x'\}\right) = f(x', v', 0). \tag{2.325}$$

The marker position (X^t, V^t) at time $t > 0$, depends obviously by the Ornstein–Uhlenbeck process on the position (X^0, V^0) at time $t = 0$ and vice versa. In the equilibrium state, Risken [124][p. 101, eqn. 5.32] shows that only for large times, the joint distribution of (V^t) and (V^0) factorizes and the two states become then actually independent. This perfectly makes sense when one thinks of collisions as destructors of the initial information content.

Applying Bayes' theorem (2.324) onto the two events $\{V^t = v \wedge X^t = x\}$ and $\{V^0 = v' \wedge X^0 = x'\}$ yields

$$\mathcal{P}\left(\{V^t = v \wedge X^t = x\}\right) = \frac{\mathcal{P}\left(\{V^t = v \wedge X^t = x\} \mid \{V^0 = v' \wedge X^0 = x'\}\right)}{\mathcal{P}\left(\{V^0 = v' \wedge X^0 = x'\} \mid \{V^t = v \wedge X^t = x\}\right)} \mathcal{P}\left(\{V^0 = v' \wedge X^0 = x'\}\right). \tag{2.326}$$

The spatial position does not change in the collision step, therefore, it is very convenient to drop the spatial dependency from eqn. (2.326) by setting $x = x'$. In the case of spatially dependent collisions, one has to drag x along, yet the outcome is the same.

$$\mathcal{P}\left(\{V^t = v \wedge X^t = x\}\right) = \underbrace{\frac{\mathcal{P}\left(\{V^t = v\} \mid \{V^0 = v'\}\right)}{\mathcal{P}\left(\{V^0 = v'\} \mid \{V^t = v\}\right)}}_{:=\eta} \mathcal{P}\left(\{V^0 = v' \wedge X^0 = x\}\right) \tag{2.327}$$

The involved conditional probabilities in η, (2.327), are in fact already known as the transition probability P of the Ornstein–Uhlenbeck process (2.304).

$$\mathcal{P}\left(\{V^t = v\} \mid \{V^0 = v'\}\right) = P(v, t|v', 0) \quad \text{and} \quad \mathcal{P}\left(\{V^0 = v'\} \mid \{V^t = v\}\right) = P(v', 0|v, t) \tag{2.328}$$

This is even more convenient, since now we can easily insert eqn. (2.304) into the ratio ν, which reads

$$\eta = \frac{P(v, t|v', 0)}{P(v', 0|v, t)} = \underbrace{\sqrt{\frac{\mathrm{e}^{2\theta t} - 1}{1 - \mathrm{e}^{-2\theta t}}}}_{=\mathrm{e}^{\theta t}} \exp\left[-\theta \frac{\left(v - \mu\left(1 - \mathrm{e}^{-\theta t}\right) - v'\mathrm{e}^{-\theta t}\right)^2 - \left(v\mathrm{e}^{-\theta t} - \mu\left(\mathrm{e}^{-\theta t} - 1\right) - v'\right)^2}{2D\left(1 - \mathrm{e}^{-2\theta t}\right)}\right] = \mathrm{e}^{\theta t} \tag{2.329}$$

If we suppose that backward diffusion is the same as forward diffusion one may falsely identify the probabilities as

$$\mathcal{P}\left(\{V^t = v\} \mid \{V^0 = v'\}\right) = P(v, t|v', 0) \quad \text{and} \quad \mathcal{P}\left(\{V^0 = v'\} \mid \{V^t = v\}\right) = P(v', t|v, 0). \tag{2.330}$$

This is justified when the system is equilibrium, because in equilibrium there is no direction of time such that forward diffusion is the same as backward diffusion. This yields then

$$\begin{aligned}
\eta = \frac{P(v, t|v', 0)}{P(v', t|v, 0)} &= \exp\left[-\theta \frac{\left(v - \mu\left(1 - \mathrm{e}^{-\theta t}\right) - v'\mathrm{e}^{-\theta t}\right)^2 - \left(v' - \mu\left(1 - \mathrm{e}^{-\theta t}\right) - v\mathrm{e}^{-\theta t}\right)^2}{2D\left(1 - \mathrm{e}^{-2\theta t}\right)}\right] \\
&= \exp\left[-\theta \frac{\left((v - \mu) - (v' - \mu)\mathrm{e}^{-\theta t}\right)^2 - \left((v' - \mu) - (v - \mu)\mathrm{e}^{-\theta t}\right)^2}{2D\left(1 - \mathrm{e}^{-2\theta t}\right)}\right] \\
&= \exp\left[-\theta \frac{(v - \mu)^2 + (v' - \mu)^2\mathrm{e}^{-2\theta t} - (v' - \mu)^2 - (v - \mu)^2\mathrm{e}^{-2\theta t}}{2D\left(1 - \mathrm{e}^{-2\theta t}\right)}\right] \\
&= \exp\left[-\theta \frac{\left(1 - \mathrm{e}^{-2\theta t}\right)\left((v - \mu)^2 - (v' - \mu)^2\right)}{2D\left(1 - \mathrm{e}^{-2\theta t}\right)}\right] \\
&= \exp\left[-\theta \frac{(v - \mu)^2 - (v' - \mu)^2}{2D}\right] = \frac{f_{eq}(v)}{f_{eq}(v')}.
\end{aligned} \tag{2.331}$$

After some elemental operations one obtains the simple propagation defined in eqn. (2.322), which was used in [125, 15]. One can successively update and store the η for every particle, as a third weight, serving as a multiplicator for the likelihoods. We chose the straightforward approach of having this information in g_n^t and f_n^t. The only thing to remember is that we are not in the Klimontovich case anymore, where f_n^t is the actual value of the density at the particle position. The δf method heavily relies on this, therefore, from time to time, one can restore this state by e.g., coarse graining, or another smoothing or interpolation technique. In the next section we show that there is also another way in handling these likelihoods.

2.6. Sequential importance re-sampling (SIR)

In the nonlinear dynamics of the Vlasov–Poisson system some particles might be more important than others. For the random markers, sampling the transport in the Vlasov equation is important, but first we need to obtain the transport model. Our first concern is to reduce the variance in the Poisson equation, as this gives us the electric field and thus the characteristics for the transport. There is a trade off between the Vlasov (2.1) and the Poisson (2.2) equation that is not yet understood. Therefore, we focus on the Poisson equation. Under the control variate, the weights $W(t)$ are not constant anymore but start to change as $\delta W(t)$ and become truly time dependent stochastic process. The simplest idea, is to split particles with large weight and delete the ones with negligible contribution. This, of course, makes only sense if the split particles do not follow the same characteristics. As the characteristics depend only on the phase space position, just creating two new particles with half the weight at the same position does not change the result, unless we have weak or strong collisions. But as Fokker–Planck collisions are involved this poses no problem.
To conclude, we want to manipulate the samples in a way such that the variance of the weights ω_k, which we belief to be highly relevant, is small.

$$\omega_n := \delta w_n = \frac{f_n - \alpha h(z_n)}{g_n} \tag{2.332}$$

2.6.1. From N_p to M_p markers

We start with a simple example suited for Vlasov–Fokker–Planck. Suppose we have an ensemble of weighted markers $(z_n(t), w_n(t))$, $n = 1, \ldots, N_p$ distributed according to the unknown probability density $g(t, z)$ and representing the density $f(t, z) > 0$ by weights $w_n = \frac{f(t,z_n(t))}{g(t,z_n(t))} > 0$. This setting corresponds to a standard particle simulation, disregarding any control variate. With f being a density, we know $w_n(t) \geq 0, \quad \forall n = 1, \ldots, N_p$. At some point in time we want to change the sample size from N_p markers to M_p. The Sequential Importance Re-sampling (SIR) or often referred to as *bootstrap filter* implements an urn model [126].

A discrete cumulative probability distribution is constructed from the samples and (2.337) is then essentially the discrete inverse transform sampling. The main drawback here is that even in the case $M_p = N_p$, we lose some phase space resolution. The same marker for a single characteristic can be drawn twice, while some others might not be drawn at all. Example: With the new particle ensemble $\{z_m^*{*}, f_m^*{*}, g_m^*{*}\}$ the standard Monte Carlo estimator for the mass becomes

$$\iint f(x, v, t)\, \mathrm{d}x\mathrm{d}v \approx \frac{1}{M_p} \sum_{m=1}^{M_p} \frac{f_m^*{*}}{g_m^*{*}}. \tag{2.340}$$

Algorithm 1 Sequential Importance Re-sampling (SIR) for full f

Input: Ensemble $\{(z_n, f_n, g_n),\ n = 1, \ldots, N_p\}$,
Weights $w_n = \frac{f_n}{g_n} \geq 0 \quad$ for all $n = 1, \ldots, N_p$
Output: Ensemble $\{(z_m^{**}, f_m^{**}, g_m^{**}),\ m = 1, \ldots, M_p\}$

1: Normalize the weights

$$\bar{w}_n := \frac{w_n}{\sum_{l=1}^{N_p} w_l}, \quad \forall n = 1, \ldots, N_p. \tag{2.333}$$

This creates a discrete probability distribution, where the probability of drawing then kth marker is $\bar{w}_n$.

$$\mathcal{P}\left(''\text{ Draw } (z_n, w_n)''\right) = \bar{w}_n \tag{2.334}$$

2: Determine the corresponding discrete cumulative distribution function by the cumulative sum p_l of $\{\bar{w}_n,\ n = 1 \ldots, N_P\}$

$$p_l := \sum_{n=1}^{l} \bar{w}_n, \quad l = 1, \ldots, N_p. \tag{2.335}$$

3: Draw M_p uniformly identically independently distributed numbers

$$u_m \sim \mathcal{U}(0, 1), \quad \text{for} \quad m = 1, \ldots, M_p. \tag{2.336}$$

4: Draw M_p markers (z_m^*, f_m^*, g_m^*) from the ensemble $\{(z_n, f_n, g_n),\ n = 1, \ldots, N_p\}$ by setting for all $m = 1, \ldots, M_p$

$$(z_m^*, f_m^*, g_m^*) := (z_n, f_n, g_n), \quad k = \max\{l : p_l \leq u_m\} \tag{2.337}$$

5: Re-normalize the weights and likelihoods. The position of the markers did not change therefore

$$(z_m^{**}, f_m^{**}) = (z_m^*, f_m^*) \text{ for all } m = 1, \ldots, M_p. \tag{2.338}$$

Recalling the normalization, eqn. (2.333) alters only the likelihood g_m^{**}.

$$\left(\sum_{l=1}^{N_p} w_l\right) = w_m^{**} = \frac{f_m^{**}}{g_m^{**}} \ \Rightarrow\ g_m^{**} := \frac{f_m^{**}}{\left(\sum_{l=1}^{N_p} w_l\right)} = \frac{f_m^*}{\left(\sum_{l=1}^{N_p} w_l\right)} \tag{2.339}$$

Also the second moment

$$\iint f(x,v,t)^2 \,\mathrm{d}x\mathrm{d}v = \iint \frac{f(x,v,t)^2}{g(x,v,t)} g(x,v,t) \,\mathrm{d}x\mathrm{d}v \approx \frac{1}{M_p} \sum_{m=1}^{M_p} \frac{(f_m^* *)^2}{g_m^* *} \tag{2.341}$$

can also be recovered, allowing application of moment matching techniques.

2.6.2. Extension to δf

Once algorithm 1 is applied onto a ensemble, a second application yields no additional effect since the weights, the likelihood ratios, do not change over time. Suppose an ensemble of belief-weighted markers $(z_n(t), \omega_n(t))$, $n = 1, \ldots, N_p$ distributed according to the unknown probability density $g(t,z)$ with the likelihoods f_n, g_n, h_n as described in (2.332). We modify algorithm 1 to show its full potential with time dependent weights.

2.6.3. Choice of importance weight

In a δf particle simulation the weights under the control variate δw_n are much more important for sampling based variance reduction techniques. Algorithm 2 provides a tool for this, which works also for negative weights disregarding the sign. But here the weights $\delta w_n(t)$ can be highly oscillating over time. A particle characteristic might always be important at a later point in time, so when disregarded to early a part of the solution will be lost. Possible remedies are additional filter algorithms on the time dependent belief $\omega_n(t) = \delta w_n(t)$.
With Algorithm 2, one is free to choose any bounded belief function to emphasize the importance of certain particles.

Algorithm 2 Sequential Importance Re-sampling (SIR) for δf

Input: Ensemble $\{(z_n, f_n, g_n),\ n = 1, \ldots, N_p\}$,
Weights $\omega_n = \frac{f_n - h(z_n)}{g_n}$
Output: Ensemble $\{(z_m^{**}, f_m^{**}, g_m^{**}),\ m = 1, \ldots, M_p\}$

1: Normalize the non-negative weights

$$\bar{\omega}_n := \frac{\omega_n}{\sum_{l=1}^{N_p} |\omega_l|}, \quad \text{for all } n = 1, \ldots, N_p. \tag{2.342}$$

2: Determine the corresponding discrete cumulative distribution function by the cumulative sum p_l of $\{\bar{\omega}_n,\ n = 1 \ldots, N_P\}$

$$p_l := \sum_{n=1}^{l} \bar{\omega}_n, \quad l = 1, \ldots, N_p. \tag{2.343}$$

3: Draw M_p uniformly identically independently distributed numbers

$$u_m \sim \mathcal{U}(0, 1), \quad \text{for } \ m = 1, \ldots, M_p. \tag{2.344}$$

4: Draw M_p markers $(z_m^*, f_m^*, g_m^*, \omega_m^*)$ from the ensemble $\{(z_n, f_n, g_n, \omega_n),\ n = 1, \ldots, N_p\}$ by setting for all $m = 1, \ldots, M_p$:

$$(z_m^*, f_m^*, g_m^*, \omega_m^*) := (z_n, f_n, g_n, \omega_n), \quad k = \max\{l : p_l \leq u_m\} \tag{2.345}$$

5: Re-normalize the weights and likelihoods. The position of the markers did not change therefore

$$(z_m^{**}, f_m^{**}) = (z_m^*, f_m^*) \text{ for all } m = 1, \ldots, M_p. \tag{2.346}$$

Again we recall the normalization (2.342) to recover the weight

$$|\omega_m^{**}| = 1 \cdot \sum_{l=1}^{N_p} |\omega_l|, \tag{2.347}$$

which shall be defined in the usual δf notation

$$|\omega_m^{**}| := \left| \frac{f_m^{**} - h(z_m^{**})}{g_m^{**}} \right|. \tag{2.348}$$

Thus, we alter only the sampling likelihood to g_m^{**}

$$g_m^{**} := \frac{|f_m^{**} - h(z_m^{**})|}{\sum_{l=1}^{N_p} |\omega_l|} \operatorname{sgn}(\omega_m^*) \tag{2.349}$$

2.7. Numerical results

At last the following simulation results combine different methods previously introduced in different ways, such that we have to treat them at the end. It is always problematic to use a highly resolved reference solution using the same solver, because the Monte Carlo convergence rate might be observed although there is a bug in the code. Therefore, if not specifically mentioned otherwise, the reference solution is obtained by a spectral solver based on Fourier transformation in x and v, and symplectic time integration. A description of the method can be found in [21].

2.7.1. Monte Carlo PIC

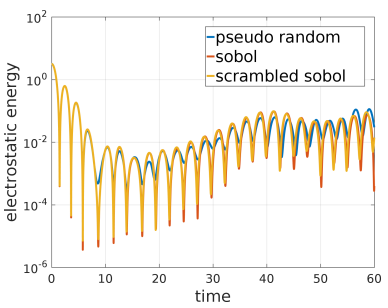
Basic questions concerning a standard PIC are many times related to conservation of entropy, convergence rates, uncertainties and choice of an appropriate time step. In the following sections these basic properties are investigated using the stochastic tools we have derived before.

Entropy estimation

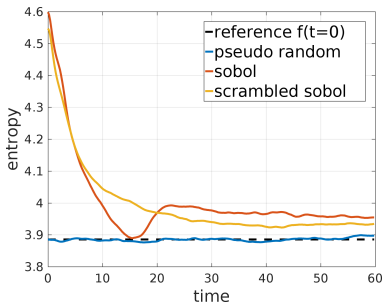
We estimate the entropy during a nonlinear Landau damping simulation using the nearest neighbor estimator (2.39). As this estimator is designed for random numbers, we are also interested in the case of the Sobol (QMC) and scrambled Sobol (RQMC) sequence Parameters are with cubic splines $k = 0.5,\ \epsilon = 0.5,\ \Delta t = 0.01,\ N_{\text{fem}} = 32,\ N_p = 10^5$.
When including Fokker–Planck collisions, we fix the collision frequency to $\theta = 0.05$ and $\sigma = 2\sqrt{2\theta}$. This setting pulls the distribution function towards the corresponding Fokker–Planck equilibrium formed by the Maxwellian $\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{1}{2}\left(\frac{v}{2}\right)^2}$, which corresponds to heating the plasma.
For collision-less Vlasov–Poisson (fig. 2.23) the nearest neighbor estimator finds the correct entropy, which also seems to be conserved during the simulation (fig. 2.23b). The low discrepancy sequences indicate a higher entropy in the beginning of the simulation, which perfectly makes sense as the sequences exhibit more order than the random numbers. Over time QMC and RQMC seem to loose the low discrepancy property and converge to the Monte Carlo entropy estimate. Although, the electrostatic energy in fig. 2.23a indicates faster convergence for (R)QMC, this seems to have little influence on the particle entropy. The Kullback Leibler entropy estimate indicates that the distribution function differs strongly from the initial condition.
Under heating the entropy raises correctly to the precalculated value (fig. 2.24b). Differing from the previous case the low discrepancy properties seems to be destroyed immediately by the random instantiation of the Ornstein–Uhlenbeck process. Due to the incorrect weight propagation, the Shannon estimator is unable to recover the correct limit (fig. 2.24d). We also see an increase in integrated variance (fig. 2.24d), although the initial field energy seems to be entirely dissipated (fig. 2.24a).

Convergence rates for MC and RQMC using $full\ f$ and δf

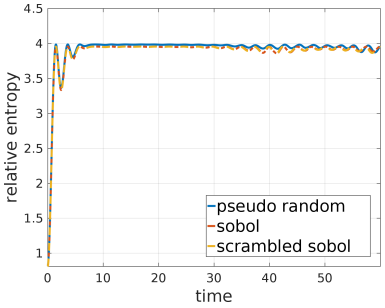
By convergence study with $N_p = 2^{12}, \ldots, 2^{21},\ \Delta t = 0.05,\ N_{\text{fem}} = 32$ and the third order symplectic Runge Kutta [32], we want to investigate the order of convergence for random and low discrepancy particles. A highly resolved reference solution of the nonlinear Landau damping problem is provided by a spectral solver. The expected convergence rates $\frac{1}{2}$ for MC and $1 - \epsilon$ for QMC are found in every result. The measure of error is the difference in



(a) Electrostatic energy for nonlinear Landau damping.

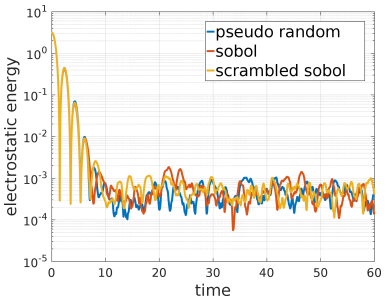


(b) Shannon entropy by nearest neighbor estimator.

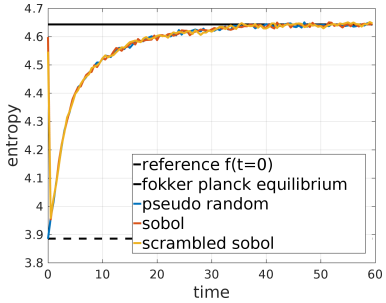


(c) Kullback-Leibler entropy with respect to the initial condition.

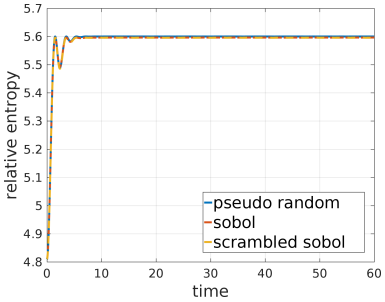
Figure 2.23.: Entropy estimates for collision-less nonlinear Landau damping considering random numbers and low discrepancy sequences.



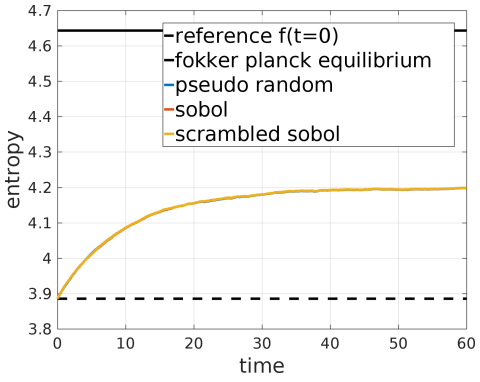
(a) Electrostatic energy for nonlinear Landau damping with heating.



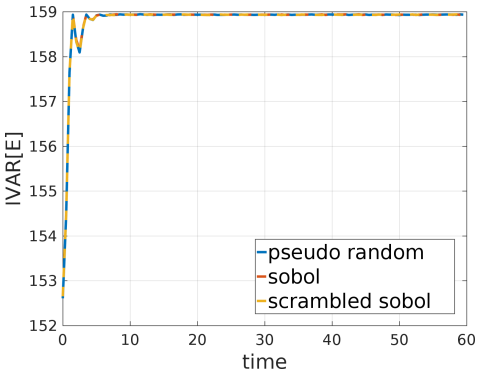
(b) Shannon entropy by nearest neighbor estimator.



(c) Kullback-Leibler entropy with respect to the initial condition.



(d) Incorrect Shannon entropy estimate by false propagation of sampling weights.



(e) Integrated variance of the electric field.

Figure 2.24.: Entropy estimates nonlinear Landau damping with strong collisions, considering random numbers and low discrepancy sequences. The entropy approaches the correct value for the Fokker–Planck equilibrium.

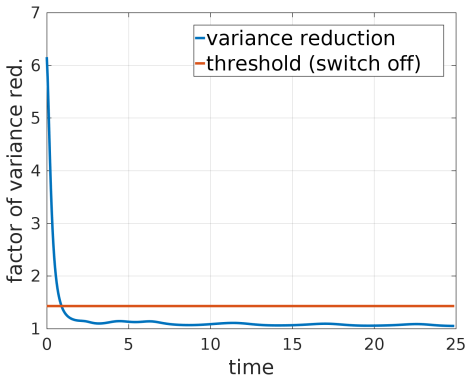
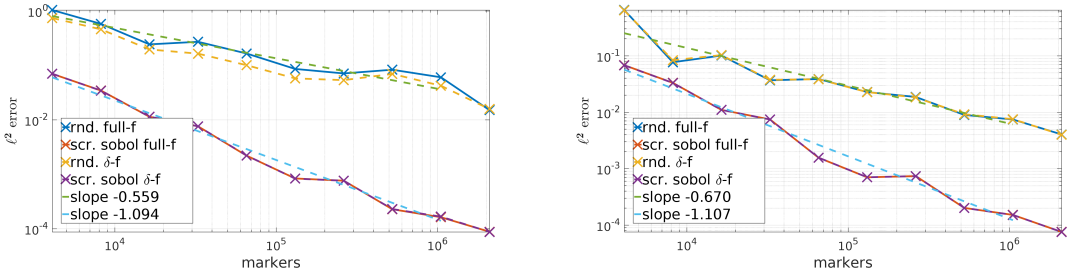


Figure 2.25.: The estimated amount of variance reduction for nonlinear Landau damping. Below the threshold the control variate is turned off. This happens in the beginning of the nonlinear phase.

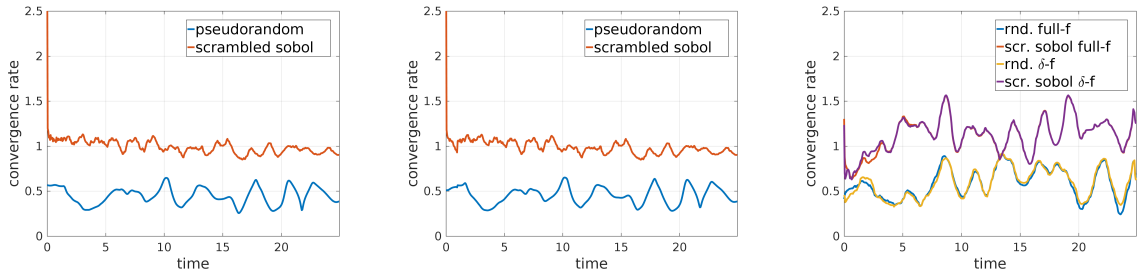


(a) Absolute ℓ^2 error for $t \in [0, 25]$ on E

(b) Absolute ℓ^2 error for $t \in [10, 25]$ on E

Figure 2.26.: Convergence diagram for PIC simulations of nonlinear Landau damping at different points in time. The Monte-Carlo and Quasi-Monte convergence can be observed independent of the control variate, which is effectively turned off after the linear phase.

electrostatic energy. But due to the initial damping see fig. 2.23a, the error in the linear phase dominates. The control variate, efficient in the linear phase (fig. 2.25), reduces the variance and, therefore, lowers the ℓ^2-error offset for the δf methods in fig. 2.26a. This effect is the strongest for the random numbers, but can be barely seen for the low-discrepancy sequence. Generously excluding the linear phase in fig. 2.26b exhibits similar convergence rates, yet - as expected - δf has no impact anymore. We are interested in a possible decay of the convergence rate over time, as a sign of degeneration. For this we estimate the convergence rate by fitting a slope for every point in time. In figures 2.27a and 2.27b the respective simulation with the largest amount of markers is used as a reference, whereas fig. 2.27c uses the standard reference, which required a simple moving average smoothing with time bandwidth of one, to make this plot readable.



(a) Convergence rate for every time step with the last sample itself as reference. (*full f*)

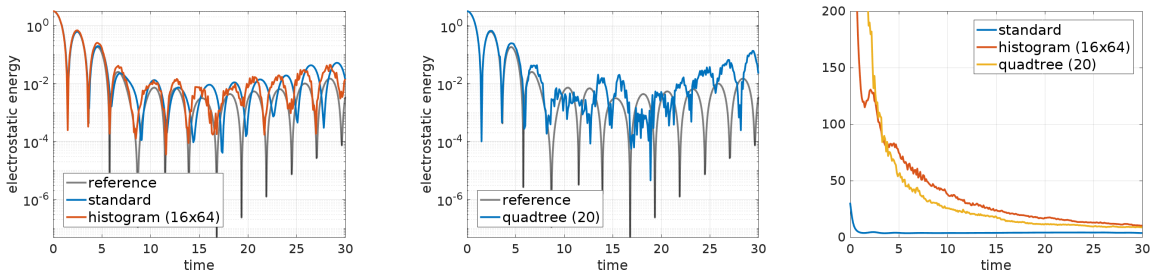
(b) Same as (a) but for δf with Maxwellian control variate.

(c) Convergence rates with respect to spectral reference solution.

Figure 2.27.: Slightly smoothed convergence rates for every time step for strong Landau damping and random and quasi-random sequences. When the last sample, the one with the largest number of particles, is taken as a reference the convergence rate is easier to obtain. Nevertheless the correct $\mathcal{O}\left(\frac{1}{N_p}\right)$ is observed for pseudo-random and $\mathcal{O}\left(\frac{1}{\sqrt{N_p}}\right)$ for quasi-random numbers, which appears to be stable over time.

Post-stratification

We want so see what improvements can be achieved by different post-stratification techniques. The implementation of the histogram method is trivial and the quad-tree method yields just another arrangement for the strata. Stratification works best on pseudo-random numbers and has in general little to no effect on quasi Monte Carlo sequences. It is hard to find a control variate for the nonlinear phase of Landau damping is such that an enhancement of the standard Maxwellian control variate by post-stratification according to the scheme in eqn. (2.254) is tested in fig. 2.28. More details have to be provided for the Voronoi method, see fig. 2.29. Inspired by [99] we create a periodic boundary condition in spatial direction by adding periodically ghost particles at both sides of the interval. The boundary in the velocity domain is realized by mirroring at the fastest particle, see fig. 2.29a. In general one should mirror at a higher position as the fastest particle such to avoid the deformed Voronoi cell for the fastest particle see fig. 2.29a. Thus the effective number of particles is quadrupled in the calculation of Voronoi cells. MATLAB uses *qhull* from [127] which is actually not the bottleneck here but rather MATLABs way of looping over cells when calculating the Voronoi volumes. The assumption that the particles are uniformly distributed within each stratum is best met when the particles are sampled uniformly at the initialization. The support of the sampling density g changes over time, but since phase space is incompressible the particles stay uniformly distributed. Stratification improves random numbers but not quasi random sequences since they already have a built in uniformity. Therefore the first example only tests the improvements on ordinary random numbers, see fig. 2.30. In this example the length of the domain was chosen larger $k = 0.3$ than in the standard test-case resulting in a weaker damping rate in order to exclude the small amplitude noise effects whilst still retaining the challenging nonlinear behavior. The Voronoi method appears to be better, such that it shall be discussed in detail here, see fig. 2.31. For uniformly distributed random numbers major improvements are made as expected from theory, see fig. 2.31a. But surprisingly also the QMC numbers, designed for uniformity are clearly improved by the post-stratification see fig. 2.31b. The Lagrangian phase space can be seen in fig. 2.32.
We can see from fig. 2.30b that the mass conservation is far from perfect. This also means that the phase space volume in each stratum is not conserved, when stratification is applied.

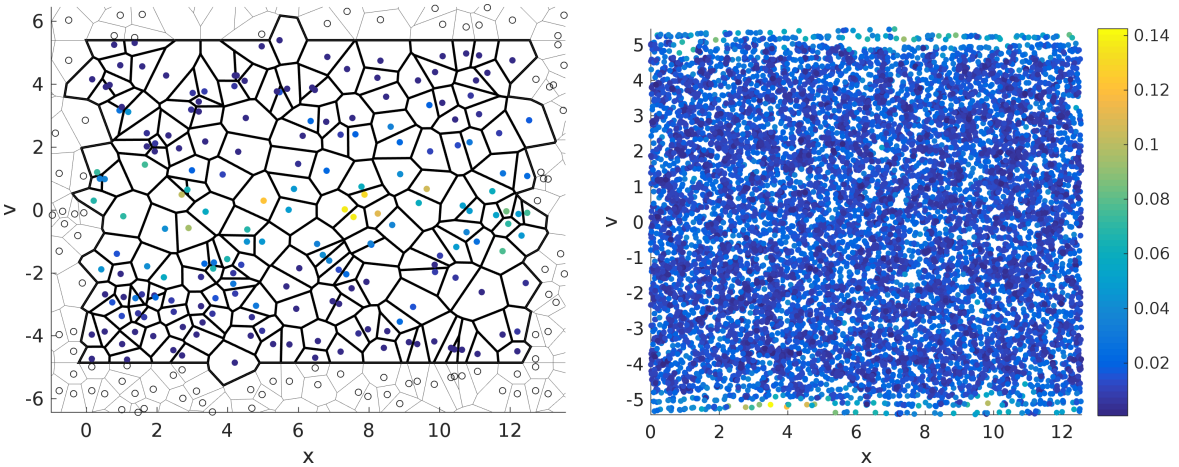


(a) Electrostatic energy for histogram.

(b) Electrostatic energy for quad-tree.

(c) Estimated factor of variance reduction.

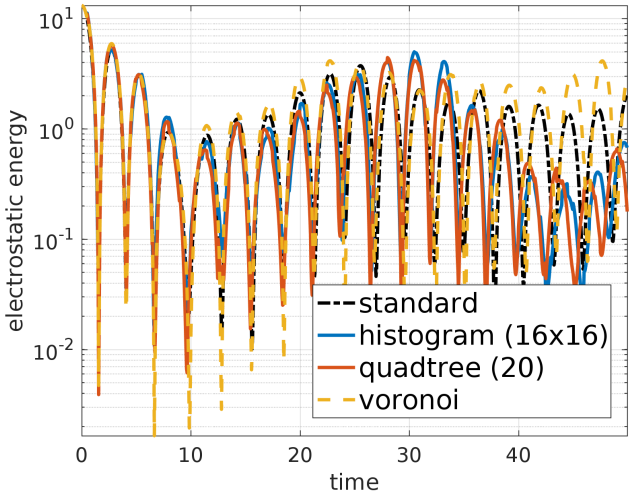
Figure 2.28.: Post-stratification of the standard Maxwellian control variate for nonlinear Landau damping with $N_p = 2 \cdot 10^4$, $\Delta t = 0.1$ with histogram and quadtree based choice of strata. The estimated variance reduction increases significantly in the nonlinear phase, and best results are obtained with the equidistant histogram method while the quad-tree method fails to enhance the simulation. Importance sampling is used, such that the additional positive effect of uniform sampling on the δf method is missing.



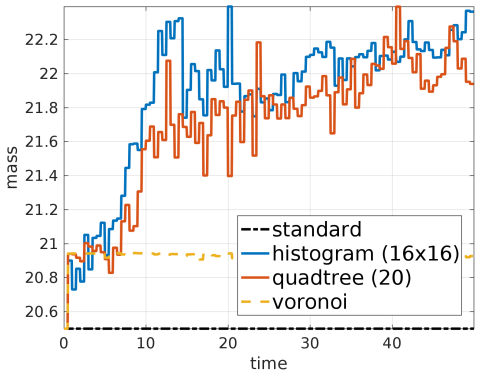
(a) For a selection of 200 markers the white ghost particles enforce periodically and mirrored boundary conditions.

(b) Weights of Voronoi volumes for $N_p = 10^4$ particles at $t = 50$ of nonlinear Landau damping.

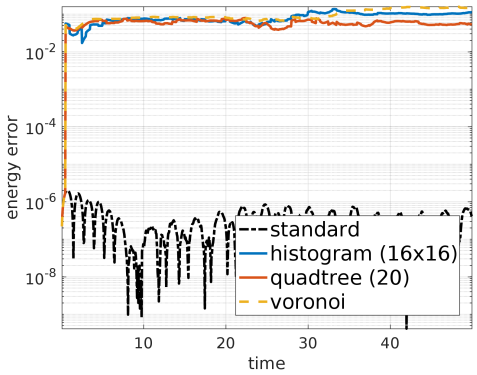
Figure 2.29.: Every particle is assigned the weight of its Voronoi volume, where ghost particles are used to create periodic and reflective boundary conditions in order to phase space volume.



(a) electrostatic energy

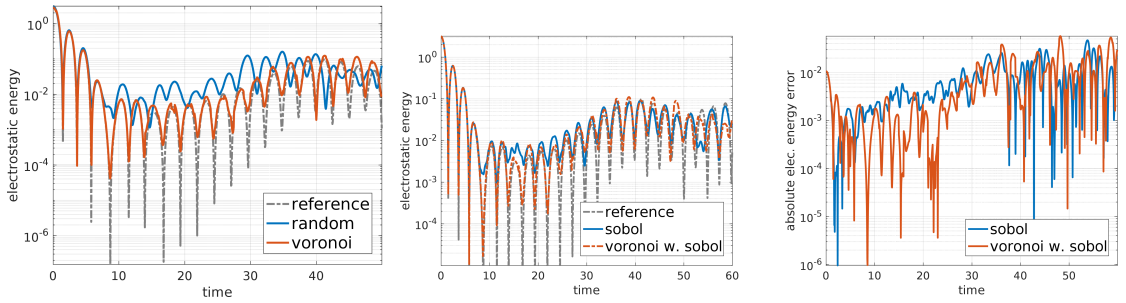


(b) mass



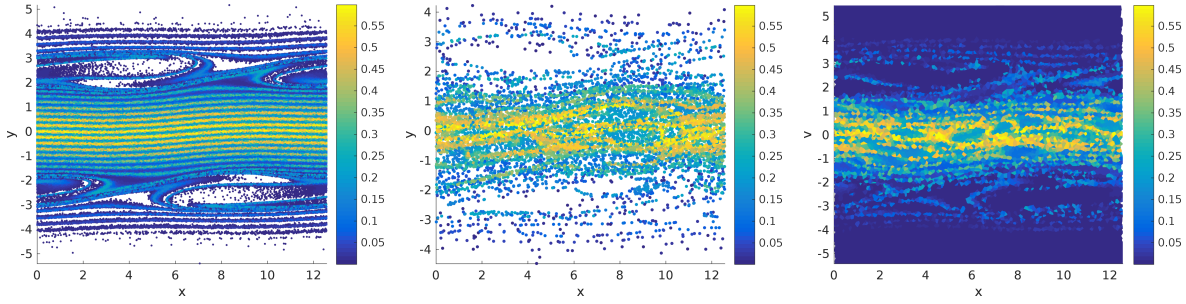
(c) energy error

Figure 2.30.: Post-stratification for nonlinear Landau Damping $\epsilon = 0.5$, $k = 0.3$, $N_p = 10^4$, $\Delta t = 0.05$ on uniformly distributed ($v_{\max} = 5$) random numbers. Re-stratification is done every tenth time step. The best results are achieved by the Voronoi method. The histogram and the Quad-tree method depend highly on the size of the strata, which leads to under and over-smoothing. The standard method uses importance sampling.



(a) MC (b) RQMC (Sobol) (c) RQMC (Sobol)

Figure 2.31.: *Voronoi* Post-stratification for nonlinear Landau Damping $\epsilon = 0.5$, $k = 0.5$, $N_p = 10^4$, $\Delta t = 0.05$.



(a) $N_p = 10^6$ RQMC particles with importance sampling (provided as reference)

(b) $N_p = 10^4$ MC particles with importance sampling

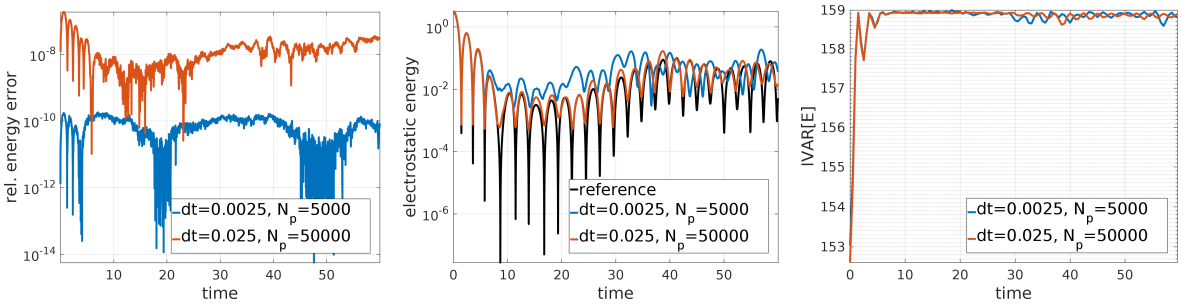
(c) $N_p = 10^4$ Voronoi post-stratified MC particles

Figure 2.32.: Lagrangian particle phase space for nonlinear Landau damping with pseudo-random Monte Carlo markers. The particle trapping effect, seen in the *RQMC* reference, is also retained for the Voronoi method.

Thus, the pure stochastic stratification is too inconsistent, such that in future research one could introduce additional constraints when calculating the new sampling weights g_n in each stratum in oder to conserve mass or even other moments of the Vlasov equation exactly.

Time step control

We have learned that PIC codes are dominated by three errors: the time integration error, the Monte Carlo noise and the finite element approximation error, where the last two form the **RMSE**.
First we want to show that the relative energy error can be a misleading diagnostic. We employ a nonlinear Landau damping $full - f$ simulation with the third order integrator **rk3s** and a very small time step, which gives good energy conservation (fig. 2.33a), yet the result does not fit the reference solution (fig. 2.33b). Increasing the number of particles N_p and the time step by a order of magnitude $\mathcal{O}(10)$, increases the relative energy error by roughly three orders of magnitude due to third order integrator (fig. 2.33a). Nevertheless, the solution is now much closer to the reference (fig. 2.33b), which now obviously stems from the number of particles. Here fig. 2.33c already shows a good estimate for the integrated variance of the electric field at low particle numbers. Note that for comparability, the factor $\frac{1}{N_p}$ does not appear here but later in the sample integrated variance. Increasing the number of particles reduced the variance and was here obviously necessary since we want to relate information about the variance and the time integrator error in order to balance the time step dt and



(a) Relative energy error for small and large time step (dt)

(b) Electrostatic energy

(c) Estimates for the integrated variance of E (without $\frac{1}{N_p}$).

Figure 2.33.: The relative energy error can be a misleading measure of solution quality (here for strong Landau damping). Decreasing the number of particles and the time step by a factor of ten yields the same costs and leads to a smaller energy error (a) but a worse solution of the problem (b). Apart from the particle number the integrated variance depends on the problem itself such that it is clear that independent of the time step a certain number of particles is needed to resolve the small amplitudes (c).

number of particles N_p.
The time integration error can be estimated for every particle by Runge Kutta methods with integrated error estimates [34][pp.181]. Here we use Heun's second order scheme 2.351 - the second order explicit trapezoidal rule, containing already the first order explicit Euler 2.350. An option for higher order is Fehlbergs Runge Kutta 4(5) [35][pp. 171], which for our testing purposes converges too fast thus unnecessarily increasing the costs of this study.
Let $(x_k^t, v_k^t), \quad k = 1, \dots, N_p$ be the random particles at time $t > 0$. The second order explicit trapezoidal rule reads

$$\begin{cases} \tilde{x}_k^{t+\triangle t} := x_k^t + \triangle t \; v_k^t \\ \tilde{v}_k^{t+\triangle t} := v_k^t + \triangle t \; \hat{E}(x_k^t, t)\frac{q}{m} \end{cases} \qquad \text{explicit Euler} \qquad (2.350)$$

$$\begin{cases} x_k^{t+\triangle t} := x_k^t + \triangle t \; \frac{1}{2}\left(v_k^t + \tilde{v}_k^{t+\triangle t}\right) \\ v_k^{t+\triangle t} := v_k^t + \triangle t \; \frac{1}{2}\left(\hat{E}(x_k^t, t) + \hat{E}(\tilde{x}_k^{t+\triangle t}, t+\triangle t)\right)\frac{q}{m} \end{cases} . \qquad (2.351)$$

Here a first order approximation of the local time discretization error ξ at time $t + \Delta t$ is obtained for every particle by

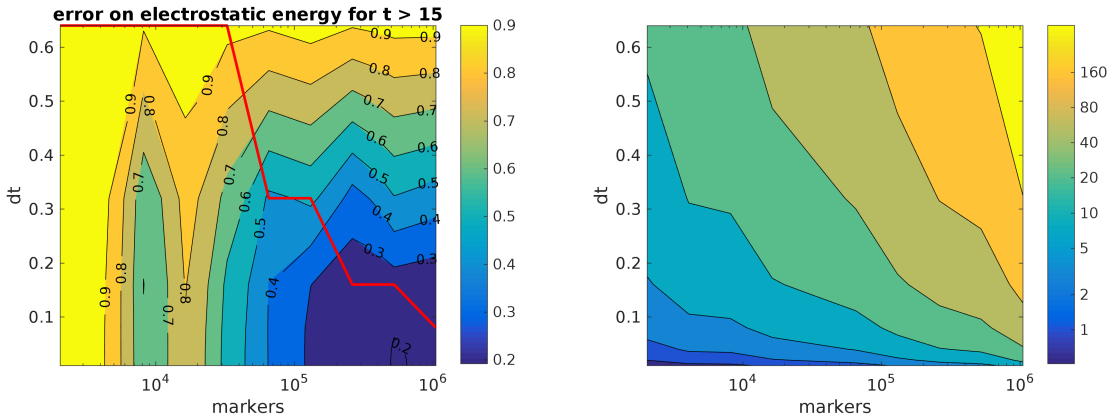
$$\xi_k = \|(\xi_k^x, \xi_k^v)\|_2 = \|\left(x_k^{t+\triangle t} - \tilde{x}_k^{t+\triangle t},\; v_k^{t+\triangle t} - \tilde{v}_k^{t+\triangle t}\right)\|_2. \qquad (2.352)$$

Already in the explicit Euler scheme we changed the notation of from the electric field $E(x,t)$ to its stochastic estimator $\hat{E}(x,t)$. Eventually the local stochastic error for a particle x_k is given by $\mathbb{V}\left[\hat{E}(x_k)\right]$, which we average over all particles yielding the integrated variance

$$\frac{1}{N_p}\sum_{k=1}^{N_p}\frac{\mathbb{V}\left[\hat{E}(x_k,t)\right]}{g_k} \approx \int_0^L \mathbb{V}\left[\hat{E}(x_k,t)\right] = \mathrm{IVAR}[\hat{E}(X,t)]. \qquad (2.353)$$

In the same way we integrate the local time discretization error

$$\bar{\xi} := \frac{1}{N_p}\sum_{k=1}^{N_p}\frac{\xi_k}{g_k}. \qquad (2.354)$$



(a) Error on the electrostatic energy for $t > 15$ for different combinations of Δt and N_p. The red line denotes a ratio $\frac{\mathrm{IVAR}[E]}{\bar{\xi}} = 80$.

(b) Ratio between $\mathrm{IVAR}\,[E]$ and mean time integration error $\bar{\xi}$.

Figure 2.34.: Balancing time discretization error and particle noise.

The electric field estimator is used at two different sub-steps in eqn. (2.351) so we assume $\hat{E}(X(t), t+) \approx \hat{E}(\tilde{X}(t+\Delta t), t+\Delta t)$. The stochastic error should be smaller than the time discretization error, which gives us the following rule of thumb using the standard deviation

$$\frac{1}{2}\triangle t\sqrt{\frac{\mathrm{IVAR}[2\hat{E}]}{N_p}} = \triangle t\sqrt{\frac{\mathrm{IVAR}[\hat{E}]}{N_p}} \quad \ll \quad \bar{\xi}. \tag{2.355}$$

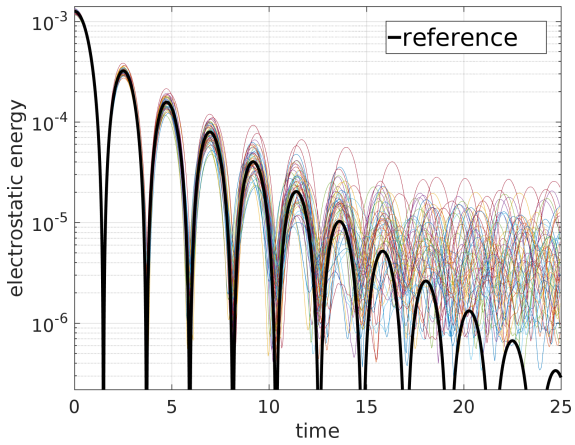
Note that the integrated variance is not accurate enough since the underlying distribution is heavily tailed. Therefore we choose the integrated variance to be at least one or two orders of magnitude smaller than the time discretization error. In a convergence study we experimentally check the relation between ℓ^2-error on the electrostatic energy, the variance and the time discretization error. The contour plot fig. 2.34a shows a nested L-shaped structure. The red line shows the ratio at which particle number and time resolution should be increased to actually decrease the ℓ^2-error. In fig. 2.34b we found the ratio $\frac{\mathrm{IVAR}[E]}{\bar{\xi}} = 80$ to fit best to fig. 2.34a.

Multiple simulations variance

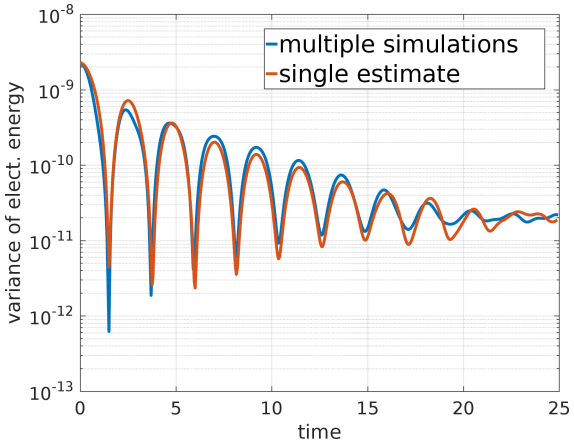
To test our variance estimators, we take a brute force approach. We run the exact same simulation for $N = 1000$ times yet with different independent initial random numbers, by each time choosing a unique seed for the pseudo random number generator. If the problem is resolved by the fixed number of particles, here $N_p = 10^4$ the result should not change. Any fluctuation on the different results gives automatically a hint where the solution can be trusted an where not. Estimating the variance over these different results gives a variance on the solution, which already incorporates already effects due to propagation of the error. This variance, which is rather expensive to obtain can be compared a local variance estimate, by using a single simulation and our standard techniques.
We consider linear Landau damping, where the errors should only propagate linear and therefore are maybe easier to estimate. The parameters are

$$L = \frac{2\pi}{k}, k = 0.5, \epsilon = 0.01, \frac{q}{m} = -1, \Delta t = 0.05, \mathit{rk3s}, N_{\mathrm{fem}} = 32, \mathit{cubic}, N_p = 10^4. \tag{2.356}$$

Fig. 2.35a, showing only a subset of all the runs, indicates clearly that the solution quality is unacceptable for $t > 10$. Using all runs as independent samples, yields an estimate of



(a) Electrostatic energy for 50 stochastically independent runs



(b) Variance of the electrostatic energy estimated from multiple runs and using error propagation for a single simulation.

Figure 2.35.: Error estimation using multiple stochastically independent simulations of linear Landau damping (a). The variance of the electrostatic energy is obtained by taking the variance over 1000 independent runs (b). A comparable result can also be obtained much cheaper using error propagation for a single simulation.

the variance of the electrostatic energy. Fig. 2.35b compares this estimate with the local *self* estimate of a single simulation, where we see fairly good agreement. Note that this uses the sample covariance, where one divides by N_p. Since the single simulation error does not account for error accumulation over time, we expect it to under estimate the variance for larger times. Fig. 2.35b exhibits the phase from over estimation to under estimation, looking at the maxima for $t \in [2.5, 17.5]$, where the general dynamic seem to be intact. It is unaffordable to run a large scale Particle-In-Cell simulations thousands of times, we seek for a simpler diagnostic. When dominated by the stochastic noise, the electric field in the time integration is the culprit. Then we can integrate the integrated variance of the electric field up to a certain time and compare it against it's $L^2_{[0,L]}$-norm to get a relative error. Then Figure 2.36 indicates that for $t > 10$ the solution cannot be trusted any more.

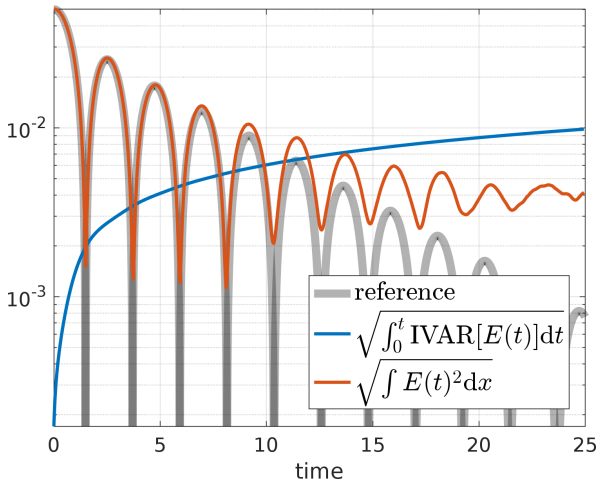


Figure 2.36.: Estimating the accumulated noise by a time integral over the integrated variance of the electric field compared to the L^2 norm of the electric field suggests that the electric field cannot be trusted anymore for $t > 10$, which is definitely true for $t > 12$ when comparing to the reference.

2.7.2. Re-sampling

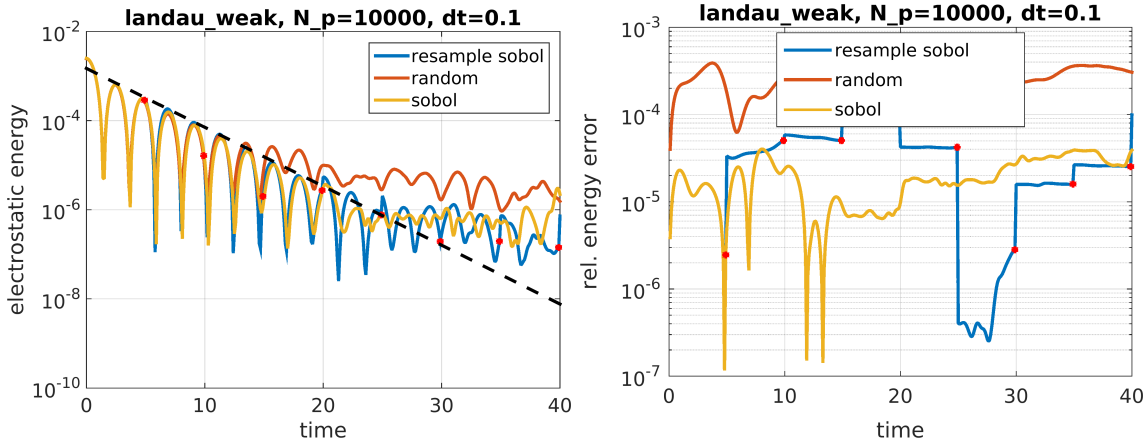
In particle methods re-sampling is done for various reasons in different ways. Sometimes one suspects the distribution function to have degenerated [128], or one wants to neglect small oscillations, perform collisions on a grid or just change the number of markers [129]. An extreme case of re-sampling is the Semi-Lagrangian method, where the particles transport the value of the distribution function along the characteristics only one over time step. Then their contribution is immediately interpolated onto the nodes of a grid, where they start again from for the next time step. Here we have no such a grid, but we always have the initial condition, therefore, we start with a backward Lagrangian method.

Backward characteristics

Suppose we have calculated the electric field up to a time $t > 0$. This contains all necessary information for describing the characteristics. Then the weight for a new randomly drawn particle at time $t > 0$ can be determined by following the characteristics backward in time to $t = 0$ and evaluating the initial condition there. This corresponds to the backward semi-Lagrangian method except that we can not step back just one time step, where we find the distribution function on a grid, but have to run backward the entire elapsed time. In this example the entire ensemble of particles is purged after a certain period and replaced by a new set of particles uniformly distributed according to

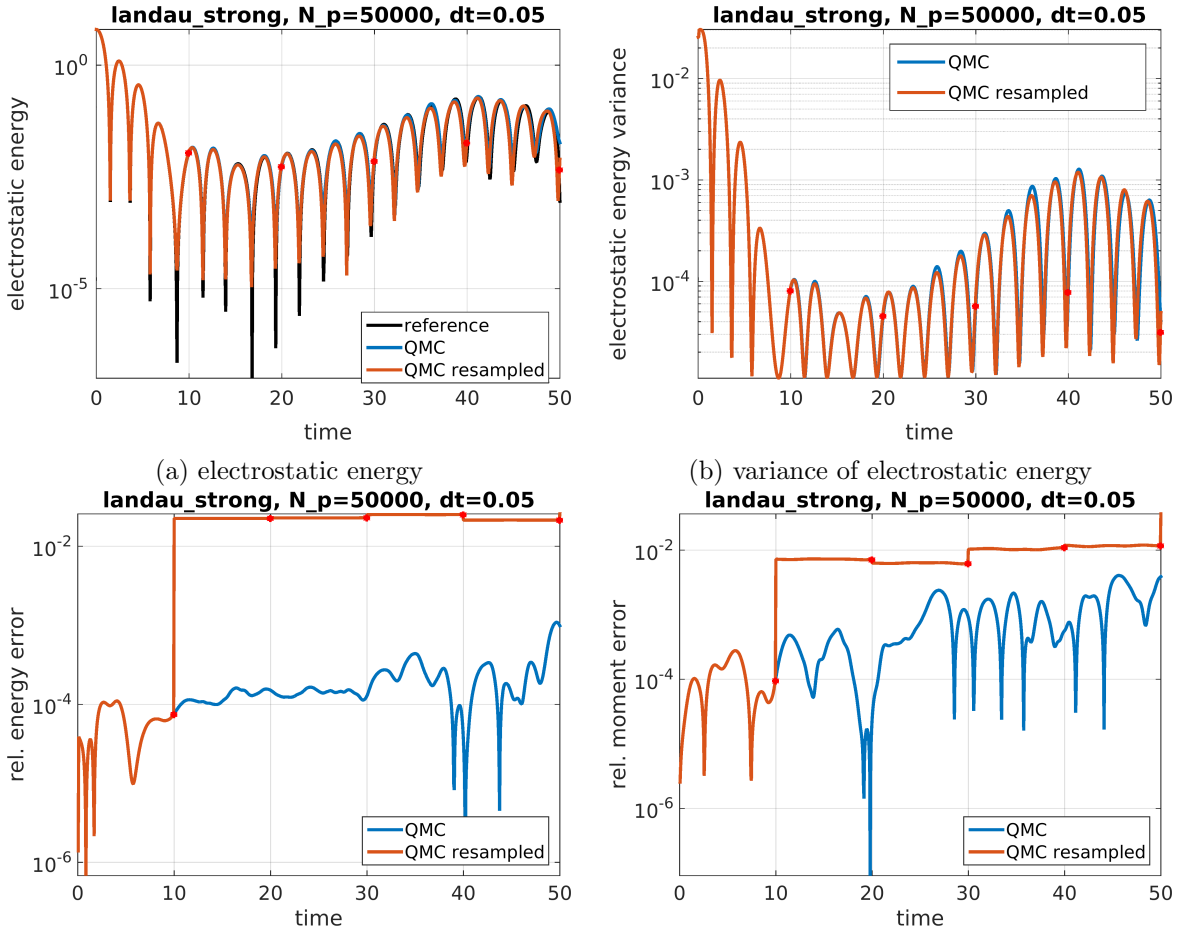
$$g(x, v, t) := \frac{1}{v_{\max} - v_{\min}} \mathbb{1}_{v \in [v_{vmin}, v_{\max}]}. \tag{2.357}$$

Fig. 2.37 and fig. 2.38 indicate that despite its exponentially growing costs this method works in the linear and the nonlinear case. By this uniform re-sampling the low discrepancy of the RQMC sequence is restored periodically, but this appears not to be relevant. Nevertheless this method can be used as a uncertainty quantification, since the extent of discontinuities provides an lower bound on the error in the simulation. Especially the energy error obtained from geometric methods is misleading as energy is discretely conserved, such that the energy error we see here for the re-sampled simulation has more significance.



(a) electrostatic energy (b) relative energy error

Figure 2.37.: Weak Landau damping (δf) with a backward Lagrangian re-sampling by uniformly distributed RQMC(sobol) particles. The red dots mark the re-sampling events. A standard Monte Carlo (random) simulation is given as reference.



(a) electrostatic energy (b) variance of electrostatic energy

(c) relative energy error (d) momentum error

Figure 2.38.: Nonlinear Landau damping with a backward Lagrangian re-sampling by uniformly distributed RQMC particles. The red dots mark the re-sampling events.

SIR for Fokker–Planck collisions

In Landau damping, the initial electrostatic energy contained in the initial perturbation wanders off into a perturbation of the Maxwellian velocity distribution. For linear Landau damping, and linearized strong Landau damping this happens immediately, whereas by actual nonlinear strong Landau damping there are nonlinear effects in between. Nevertheless, it ends with a perturbed Maxwellian, such that the local Maxwellian control variate should give us a good control variate. Except it does not for a large amount of field energy $\epsilon > 0.1$. The perturbation of the Maxwellian takes higher frequency over time, such that at some point it cannot be resolved anymore. Yet every particle carries completely free of diffusion the values of the distribution function along, whereas a lack of resolution should be healed with diffusion. One can introduce Fokker–Planck collisions but still the control variate does not work again, see 2.39a. The likelihoods f_k start to peak, giving an unnatural representation of the density, see fig. 2.39c. We can try sequential importance re-sampling SIR with the weight

$$\delta W = \frac{f(X,V) - f_{equ}(V)}{h(X,V)} \tag{2.358}$$

using the Fokker–Planck equilibrium f_{equ}, which is by definition the local Maxwellian. This also smoothens the likelihoods (fig. 2.39d) and modifies the sampling distribution such that the control variate correlates again, see fig. 2.39a. Already without collisions it is questionable, whether particles with small δW should be neglected, because at later times they can be relevant again, see fig. 2.40. We are not satisfied yet, since there are discontinuities in 2.39b. It is obvious that the re-sampling should be done continuously in time also and not only at certain points. This also avoids the up and down of the control variate correlation in the equilibrium under collisions. The answer is a slight modification, which stochastic sequential importance re-sampling.

2.7.3. Collisions and coarse graining

Here we focus on the Vlasov–Poisson–Fokker–Planck system which incorporates collisions. Depending on the collision frequency the particle distribution approaches the Maxwellian equilibrium such that a Maxwellian control variate should gain efficiency again. Yet this does not work, such that we have to introduce a coarse graining technique. In order to understand the general problem we start with the stochastic counterpart of a one dimensional Fokker–Planck equation the Ornstein–Uhlenbeck process.

Ornstein–Uhlenbeck

In this one dimensional example we start with an initial shifted two stream density

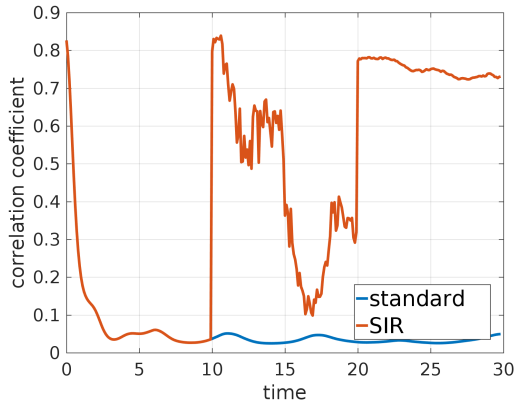
$$f(v, t=0) = \frac{1}{\sqrt{4\pi}} \left(\mathrm{e}^{(v+1)^2} + \mathrm{e}^{(v-3)^2} \right). \tag{2.359}$$

This density shall follow the Fokker–Planck eqn. (2.360) with collision frequency $\theta = 0.01$, diffusivity $D = 0.04$ and drift $\mu = 0$.

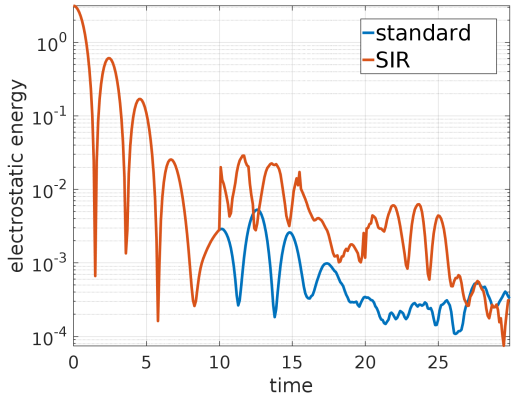
$$\partial_t f(v,t) = \theta \frac{\partial}{\partial v} \left[(v-\mu) f(v,t) \right] + D \frac{\partial^2 f(v,t)}{\partial v^2} \tag{2.360}$$

For long time the density f approaches the equilibrium in form of a single Gaussian f_{equ} which reads

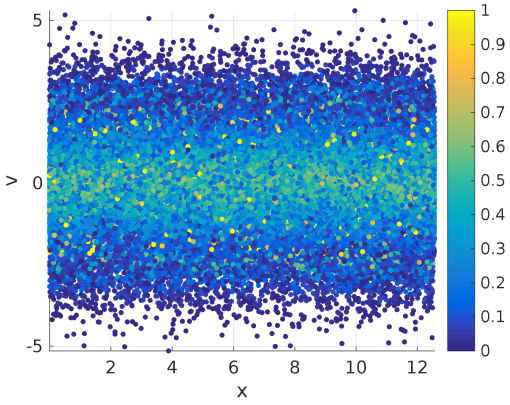
$$f_{equ}(v) = \frac{\theta}{\sqrt{2\pi D}} \mathrm{e}^{-\theta \frac{(v-\mu)^2}{2D}}. \tag{2.361}$$



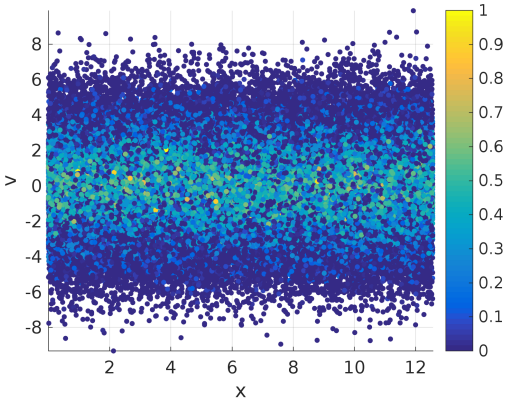
(a) Correlation of the local Maxwellian control variate.



(b) Electrostatic energy with discontinuities at filtering times.

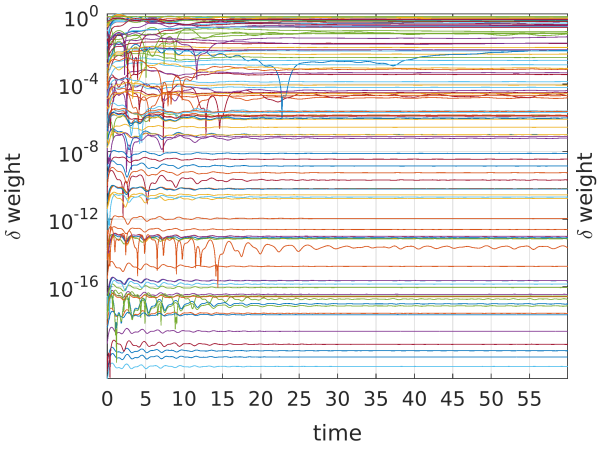


(c) Likelihood f_k for every particle at $t = 30$. Strong peaking of the likelihoods.

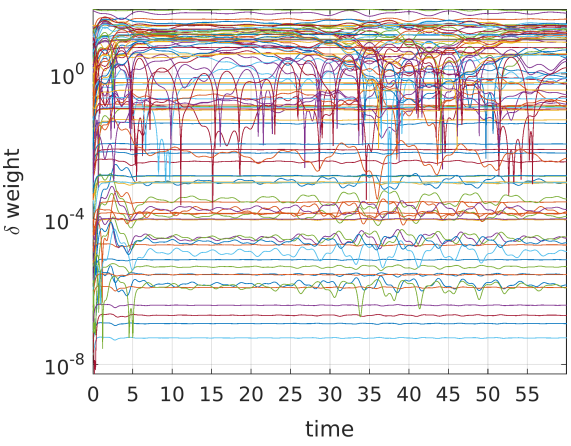


(d) Likelihood f_k for every particle at $t = 30$ under SIR. Peaking likelihoods have been split.

Figure 2.39.: Effects of the brute force particle filtering (SIR) on nonlinear Landau damping under moderate $\theta = 0.01$ Fokker–Planck collisions. The control variate gains efficiency (a), because the peaking likelihoods (c) are split (d), but unfortunately the solution is destroyed (b).



(a) weak Landau damping



(b) strong Landau damping

Figure 2.40.: Development of the δw weights over time for some randomly selected particles. In the linear simulation the weights are basically unperturbed, but in the nonlinear examples unimportant particles with small weights develop large weight at different times.

Let V_0 be a random deviate distributed according to the initial condition $f(v, t = 0)$. Then the random deviate V_t obtained by following the Ornstein–Uhlenbeck process,

$$V_t = V_0 \mathrm{e}^{-\theta t} + \mu \left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} W_{1-\mathrm{e}^{-2\theta t}} \tag{2.362}$$

is distributed according to $f(v, t)$ as solution of eqn. (2.360). In the semi-discretization we obtain a jump over time t by use of another independent deviate $U \sim \mathcal{N}(0, 1)$.

$$V_t = V_0 \mathrm{e}^{-\theta t} + \mu \left(1 - \mathrm{e}^{-\theta t}\right) + \sqrt{\frac{D}{\theta}} \sqrt{1 - \mathrm{e}^{-2\theta t}} U \tag{2.363}$$

In order to cover a time distance t the interval $[0, t]$ can also be divided in N_t multiple jumps. Now, there are two options of sampling from the equilibrium density. One can either directly sample a normally distributed random deviate $V_\infty \sim \mathcal{N}(\mu, \frac{D}{\theta})$ from the equilibrium distribution or use V_t for a very large t. In the following we want to calculate the integral of the function $h(v)$ by Monte Carlo integration.

$$h(v) = \mathrm{e}^{-0.3v^2} \frac{(v+3)v(v-2)+1}{15}, \quad \mathcal{I} = \int_{-\infty}^{\infty} h(v)\mathrm{d}v \approx 0.575296566683170 \tag{2.364}$$

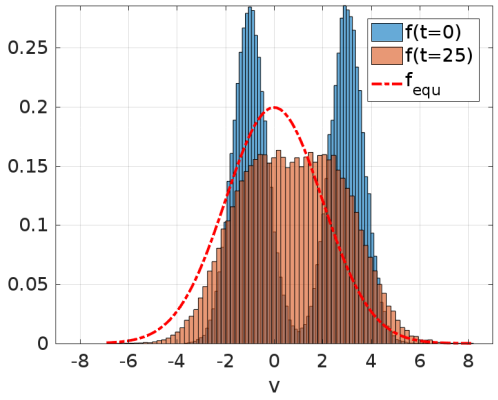
The integral is obtained as expected value using markers obtained from V_0, V_t and V_∞.

$$\int_{-\infty}^{\infty} h(v)\mathrm{dv} = \mathbb{E}\left[\frac{h(V_0)}{f(V_0, 0)}\right] = \mathbb{E}\left[\frac{h(V_t)}{f(V_t, t)}\right] = \mathbb{E}\left[\frac{h(V_\infty)}{f_{equ}(V_\infty)}\right] \tag{2.365}$$

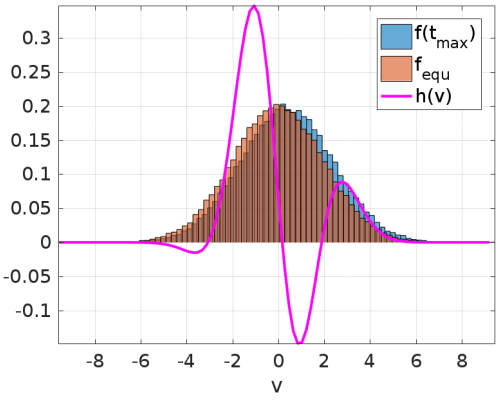
The knowledge of the exact value of the integral $\mathcal{I}$ can then be used via the control variate method in order to reduce the variance of other estimators using the random deviates V_0, V_t and V_∞. The slight problem here is that $f(V_t, t)$ is not known directly such that we have to use the likelihood η_t corresponding to the transition probability in the Ornstein–Uhlenbeck process. The purpose of the example here is to show that we can find a converging Monte Carlo estimator for the integral $\mathcal{I}$ in order to use the function h as a control variate. Although V_t follows a stochastic process it is only the combination of two random deviates V_0 and U. Since both distributions are known Monte Carlo integration with V_t, given the correct likelihoods, is possible. The Monte Carlo estimator combining importance sampling and the Markov chain likelihood η_t is then based upon

$$\mathbb{E}\left[\frac{h(V_t)}{\underbrace{f(V_t, t)}_{unknown}}\right] = \mathbb{E}\left[\frac{h(V_t)}{f(V_0, t)\eta_t}\right]. \tag{2.366}$$

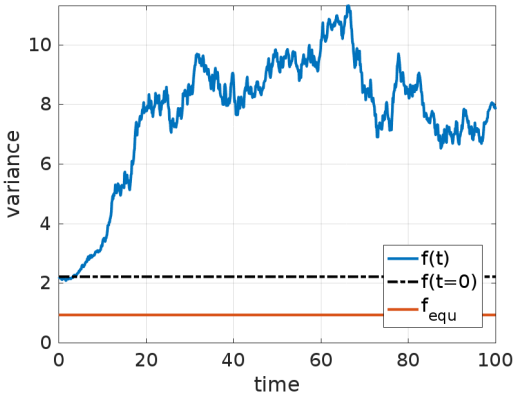
The results are found in fig. 2.41. There it becomes clear, that although the samples V_t approach V_∞ the variance on the standard Monte Carlo estimator increases due to the incorporated likelihood propagation. The samples for f_{equ} are drawn independently an yield a much smaller variance, since the Gaussian sits right on top of the integrand. Understandably the two streams $f(t = 0)$ are not suited for integrating h, yielding a higher variance. In order to reduce the variance f_{equ} is not a suitable control variate. But if one defines several control variates $f_{equ,j} = f_{equ}(v)\mathbb{1}_{v\in\Omega_j}$ as the restrictions of f_{equ} onto a stratum $\Omega_j \subset \mathbb{R}$ a significant variance reduction is achieved for every estimator. We call this a stratified control variate. Now that we can calculate Monte Carlo integrals and apply control variates under an Ornstein–Uhlenbeck process, we can proceed with the more complicated Vlasov equation.



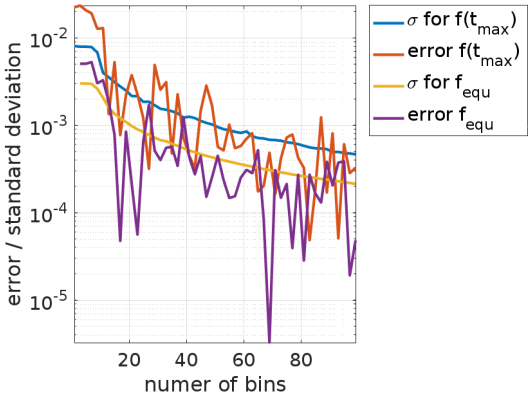
(a) Densities obtained by histogram estimates. Over the two Gaussians approach the equilibrium f_{equ}.



(b) For $t_{\max} = 100$ the density f is very close to the equilibrium, such that both markers distributions can be used to integrate h.



(c) Estimating $\int h dv$ by Monte Carlo integration using V_0 and V_∞ as well as V_t with likelihood propagation and $N_t = 10^3$ time-steps.



(d) Because f_{equ} is an ineffective control variate it is stratified into multiple control variates onto several boxes.

Figure 2.41.: Monte Carlo integration with $N_p = 10^5$ random samples subject to an Ornstein–Uhlenbeck process. Due to the likelihood propagation the variance increases, but can be reduced by a stratified control variate.

Vlasov–Poisson–Fokker–Planck

Coarse graining for Fokker–Planck collisions is a special case of conditional Monte Carlo. The sampling likelihoods are averaged in every stratum Ω while the ratio between f and g is retained for every particle, yielding the temporary likelihoods f_n^* and g_n^* according to

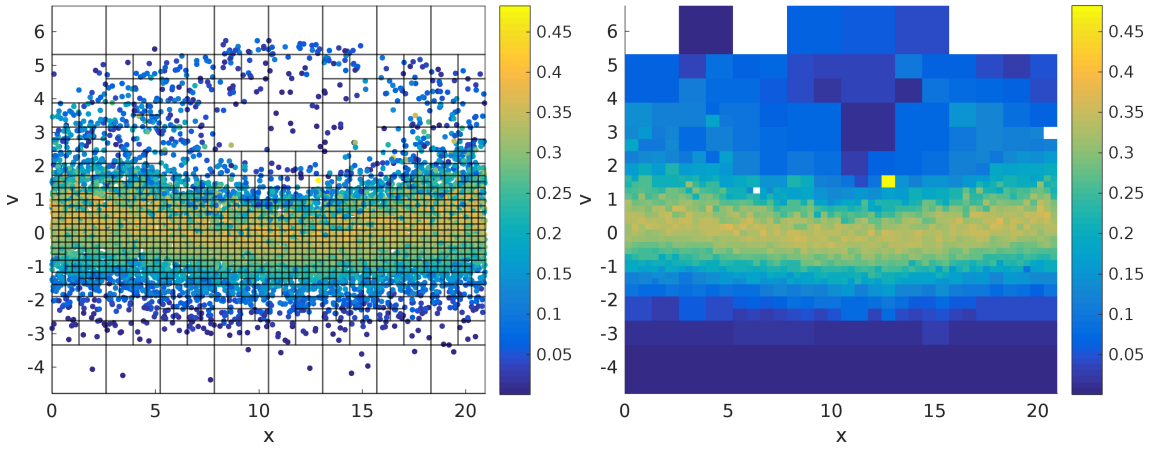
$$\begin{aligned} g_n^* &= \frac{1}{\#\{z|z \in \Omega_j\}} \sum_{z_m \in \Omega_j} g_n \qquad \text{for all } z_n \in \Omega_j \\ f_n^* &= \frac{f_n}{g_n} g_n^*. \end{aligned} \tag{2.367}$$

The original likelihoods are saved, respectively advanced for the collisions, and reused at each coarse graining step. Therefore no information is lost, which is more in the sense of conditional Monte Carlo and mitigates degenerate effects of the coarse graining such that result is less sensitive to the coarse graining frequency. In the examples here the strata Ω_j are obtained by the quad-tree sorting algorithm with a maximum number of particles per box of 20. The basic principle behind the quad-tree sorting algorithm can be seen in fig. 2.42. As another option we use the N^{th}-nearest neighbor method in MATLAB to find for every particle a set Ω_j with the N^{th} (here $N = 10$) nearest neighbors in phase space and coarse grain again according to eqn. (2.367). The first test case is the Bump-on-Tail instability [63][p.140] with initial conditions given in eqn. (2.368).

$$f(x, v, t = 0) := (1 - \epsilon \cos(kx)) \frac{1}{\sqrt{2\pi}} \frac{1}{(1 + a)} \left(\mathrm{e}^{-\frac{v^2}{2}} + \frac{a}{\sigma} \mathrm{e}^{-\frac{(v - v_0)^2}{2\sigma^2}} \right) \tag{2.368}$$

$$L = \frac{2\pi}{k}, \quad \sigma = 0.5, a = \frac{2}{9}, \quad k = 0.3, v_0 = 4.5, \frac{q}{m} = -1, \ \epsilon = 0.03 \tag{2.369}$$

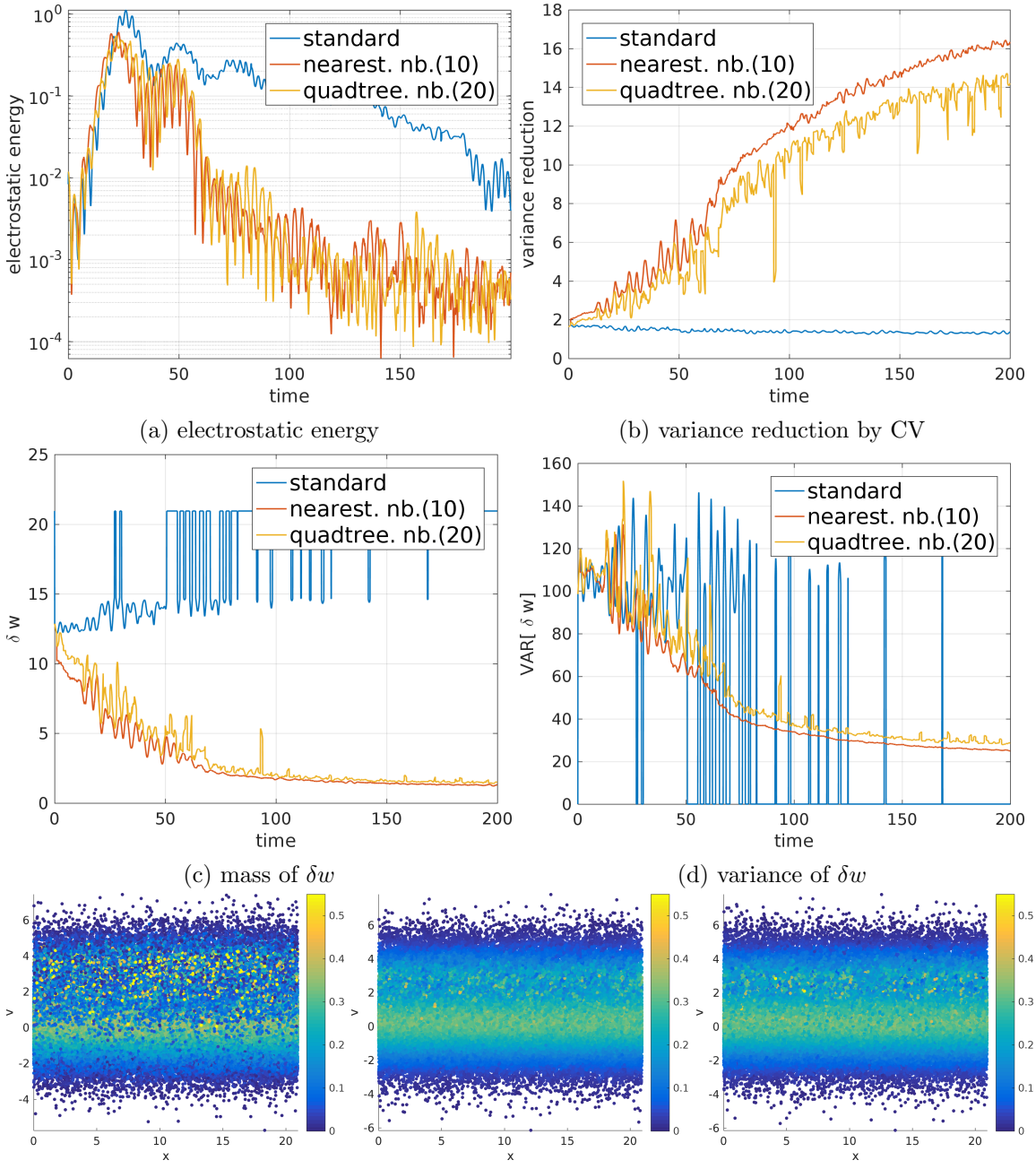
The demonstrations include the Bump-on-tail instability with weak (fig. 2.43) and strong (fig. 2.44) collisions as well as nonlinear Landau damping with strong collisions, see fig. 2.45. In all cases the coarse graining yields a better correlation for the control variate thus increasing the variance reduction when the equilibrium is reached.



(a) Quad-tree structure with particles.

(b) Averaged likelihood for every box in the Quad-tree.

Figure 2.42.: The quad-tree algorithm reduces recursively the size of the boxes such that each box contains less than a certain number of particles. The likelihoods are averaged in each box yielding a coarser representation of phase space.



(a) electrostatic energy

(b) variance reduction by CV

(c) mass of δw

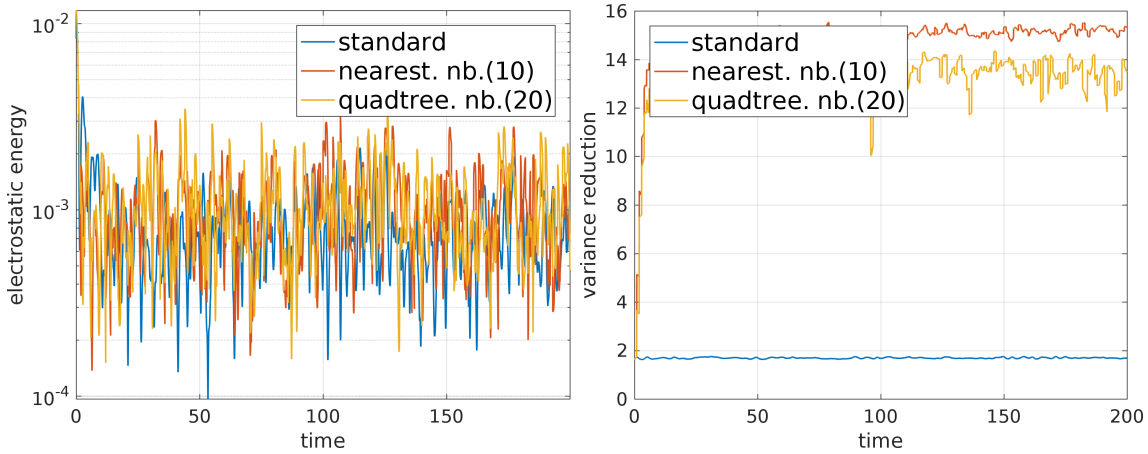
(d) variance of δw

(e) standard

(f) nearest neighbor

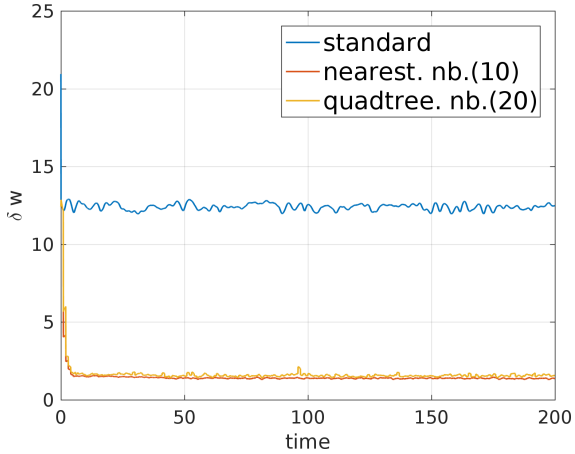
(g) quad-tree

Figure 2.43.: Coarse graining the Bump-on-tail instability under weak collisions $\theta = 0.001$ and $N_p = 10^5$. The coarse graining reduces the peaking of the likelihoods (e,f,g) and enhances the variance reduction by the control variate (b), but it also introduces too much diffusivity leading to a strong damping of the instability (a).

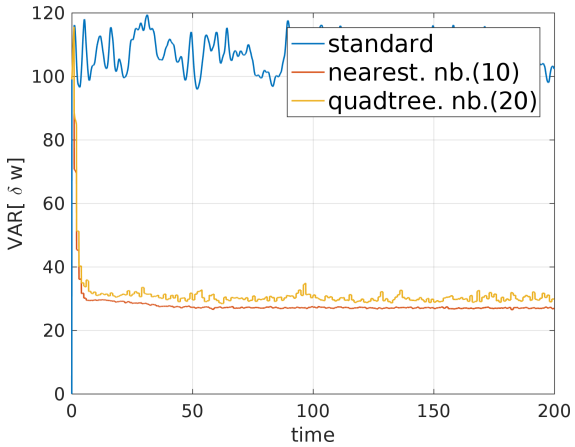


(a) electrostatic energy

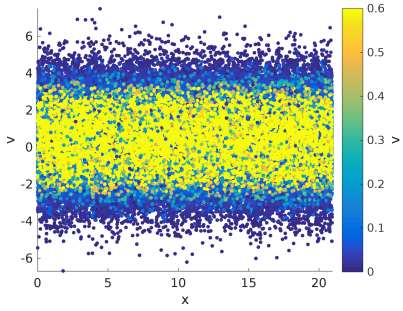
(b) variance reduction by CV



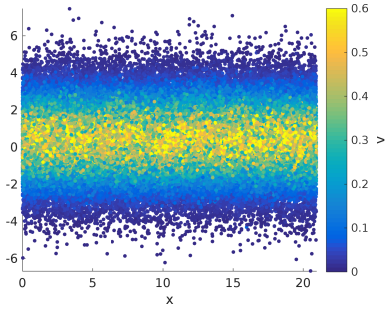
(c) mass of δw



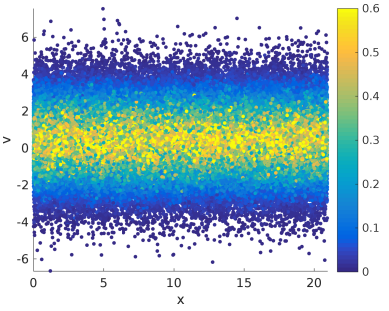
(d) variance of δw



(e) standard

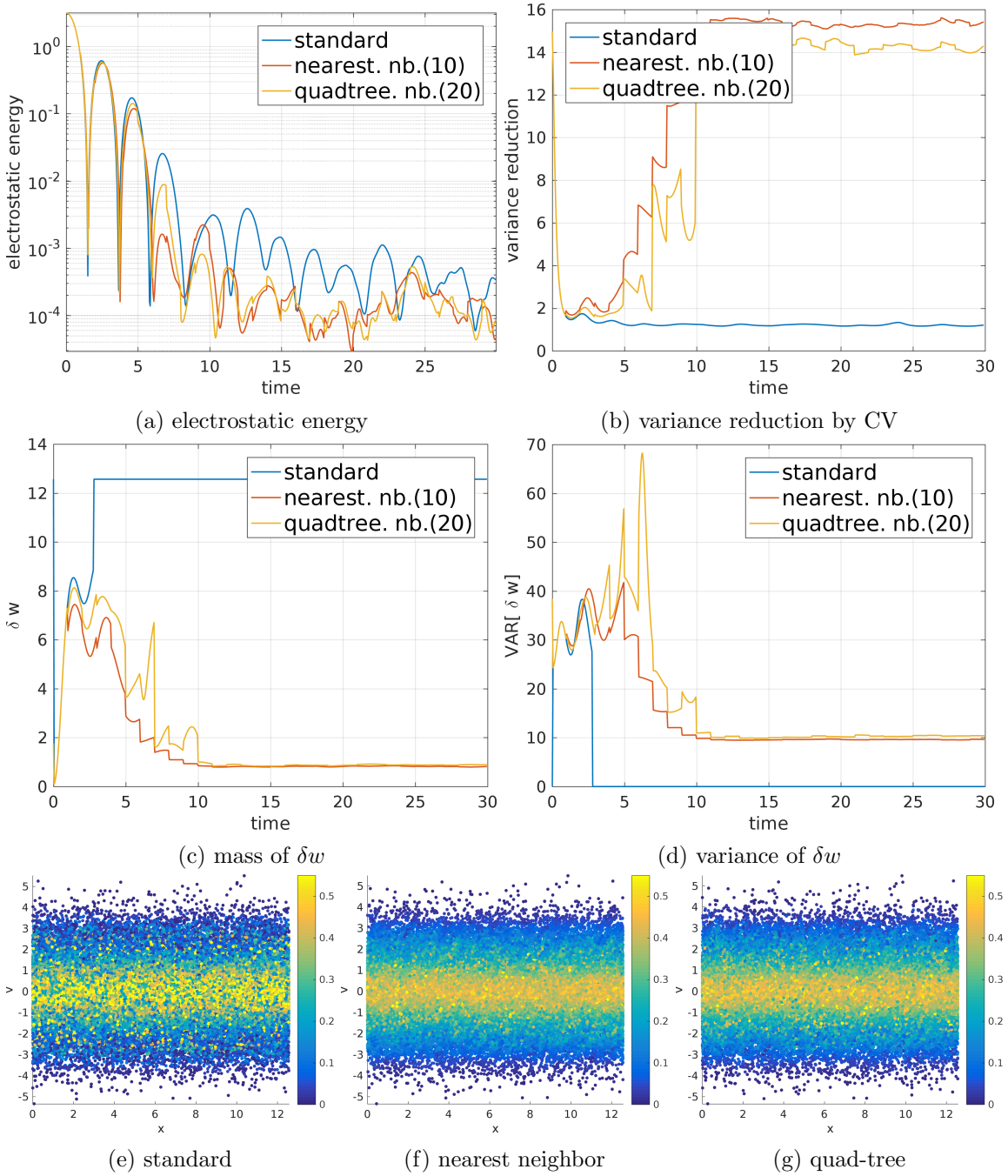


(f) nearest neighbor



(g) quad-tree

Figure 2.44.: Coarse graining the Bump-on-tail instability under strong collisions $\theta = 0.05$ and $N_p = 10^5$. The overshooting of the likelihoods (e) is successfully damped by the coarse graining for both methods (f),(g), such that the control variates gains efficiency in the equilibrium. The high diffusivity does not seem to matter here (a).



(a) electrostatic energy

(b) variance reduction by CV

(c) mass of δw

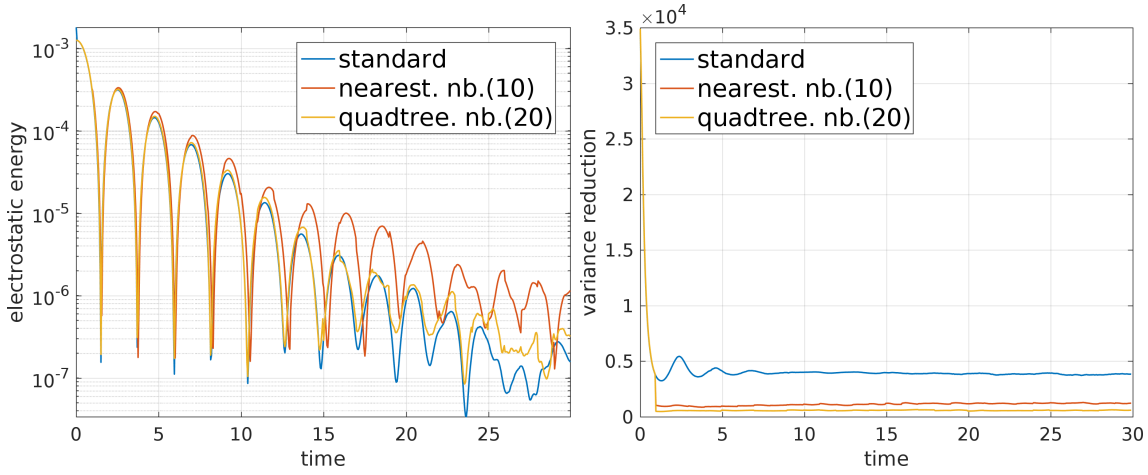
(d) variance of δw

(e) standard

(f) nearest neighbor

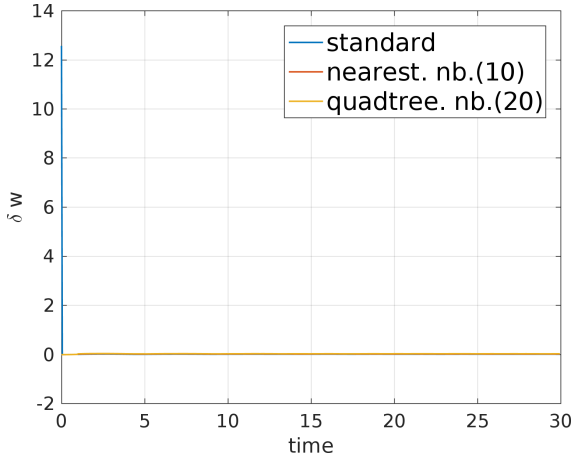
(g) quad-tree

Figure 2.45.: Coarse graining strong Landau damping under moderate collisions $\theta = 0.01$ and $N_p = 10^5$. The diffusivity of the coarse graining can be seen in (a) as the Langmuir wave gets stronger damped. Once the equilibrium is reached it helps the control variate (b).

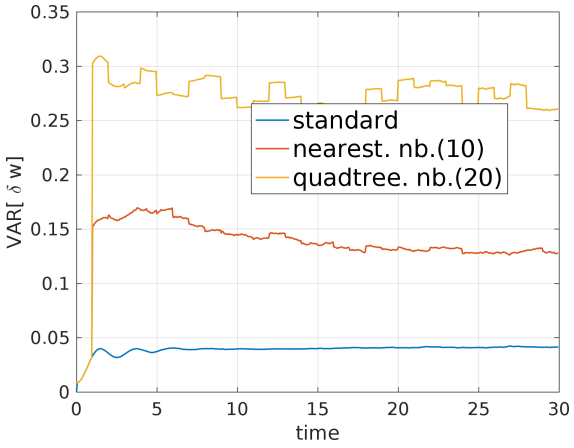


(a) electrostatic energy

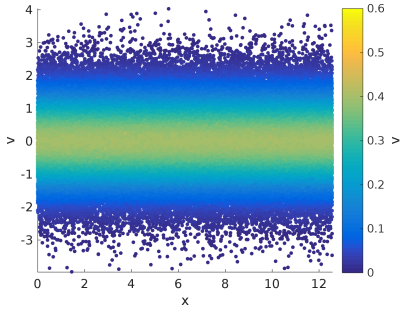
(b) variance reduction by CV



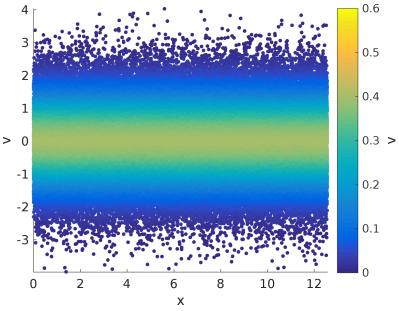
(c) mass of δw



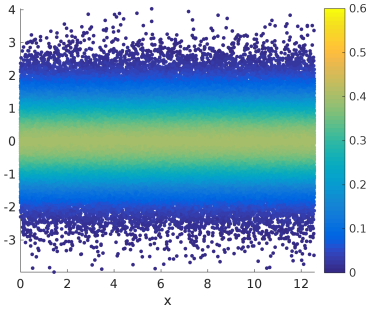
(d) variance of δw



(e) standard



(f) nearest neighbor



(g) quad-tree

Figure 2.46.: Coarse graining weak Landau damping under weak collisions $\theta = 0.01$ and $N_p = 10^5$. Here the coarse graining is definitely not recommended since the likelihoods do not peak(e), it falsifies the solution (a) and brings no additional variance reduction (b).

2.7.4. Principal component analysis

With the covariance matrix $\Sigma_{b(t)}$ we obtain much information about the noise, which should be analyzed. The zeroth Fourier mode is already understood to be the source of noise, for small amplitudes, e.g. in the case of $\epsilon = 0.03$. By principal component analysis we can [130] quantify the noise, using estimates of $\Sigma_{b(t)}$ and the corresponding propagation for the field solver. We can see the noise reduction for the linear phase due to the control variate, and also find the source of noise for the higher modes. It provides also a natural way of filtering. Let $\Psi \in \mathbb{R}^{N_{\text{fem}} \times N_p}$ the sparse matrix containing the evaluation of all basis functions for all particles.

$$(\Psi)_{n,m} = \psi_n(x_m)w_m - \frac{1}{N_p}\sum_{k=1}^{N_p}\psi_n(x_k)w_k, \qquad n = 1, \dots, N_{\text{fem}}, \quad m = 1, \dots, N_p \tag{2.370}$$

Then the covariance matrix Σ_b can also be obtained by

$$\Sigma_b = \frac{1}{N_p - 1}\Psi^t\Psi. \tag{2.371}$$

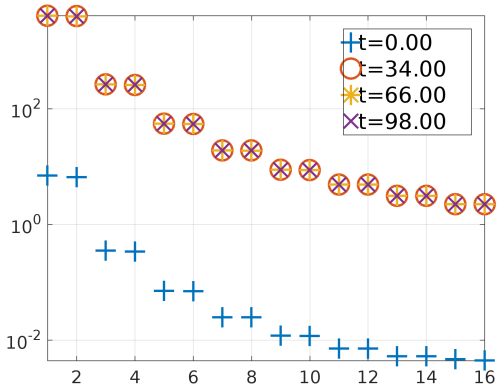
Here the eigenvector to the largest eigenvalues of the covariance matrix corresponds to the direction of the largest variance. Since only the potential is used for the electric field, we can calculate the first N_{pc} principal components of the potential covariance matrix Σ_Φ. The spectrum of Σ_Φ reveals several spectral gaps, grouping always two eigenvalues, see fig. 2.47a. Grouping the pairs of two eigenvectors in v, we plot the corresponding function $x \mapsto v^t\psi(x)$, which reveals the dominant modes of the simulation, see figs. 2.47b, 2.47c and 2.47d. Let V denote the matrix containing the first N_{pc} normalized eigenvectors of Σ_Φ, with the corresponding eigenvalues in d. It then is possible to filter the coefficient vector of the electric potential $a(t)$ by $a(t)_{filterd} := VV^t a(t)$. This yields a similar electrostatic energy as the standard simulation, c.f. fig. 2.48a. When the control variate looses effect in fig. 2.48c the spectrum of variances in the PCA analysis fig. 2.47a raises, by orders of magnitude, yet the overall structure remains. A quite interesting diagnostic is the signal to noise ratio 2.48b, where we plot the ratio

$$SNR = \frac{\sum_{k=1}^{N_{pca}} d_k}{tr\left(\Sigma_\Phi\right) - \sum_{k=1}^{N_{pca}} d_k}. \tag{2.372}$$

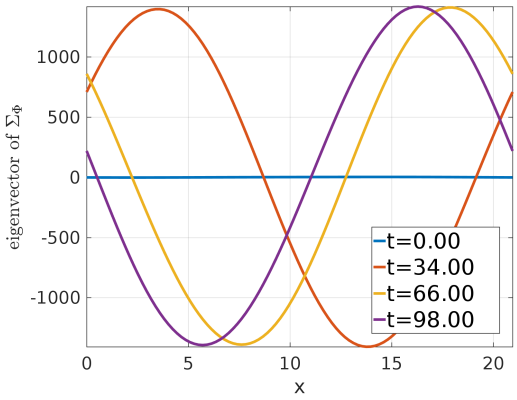
This corresponds to the ratio of the variance that is kept, v.s. the variance in the other components that is neglected. Although for this one dimensional example filtering of the Laplacians eigenmodes is obvious, this general approach can be extended to cases where there is no pre-known mode structure e.g. complex geometries.
Singular value decomposition (SVD), closely related to PCA [130], can give us information for every sample. We would like to identify the particles, with the greatest contribution to the electric potential. To get from the pure contribution to the basis functions, stored in Ψ the Poisson equation is solved for every particle resulting in Ψ_Φ. We perform a SVD on Ψ_Φ, see (2.373), where U corresponds to the eigenvectors of the covariance matrix Σ_Φ.

$$\Psi_\Phi := \mathcal{K}\Phi, \quad \Psi_\Phi = USV^t, \tag{2.373}$$

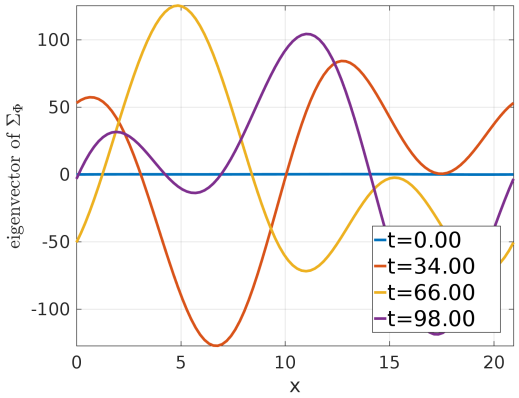
The columns of V are vectors which hold the contribution of every particle to the respective singular value in the diagonal matrix S. It is also the contribution of every particle to the corresponding eigenvector of the potential covariance matrix Σ_Φ. Here, we only use the particle positions x_k. This spatial information should be connected with the velocity space, which is done by considering the phase space position of every particle. By coloring



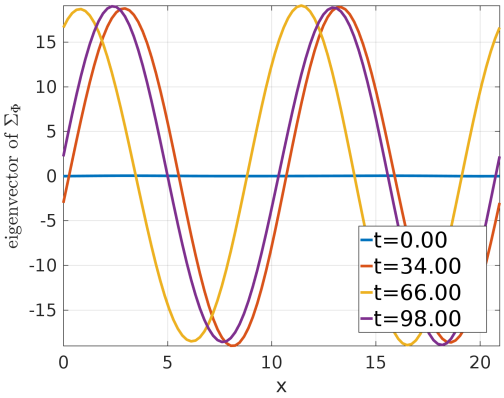
(a) Eigenvalues of the electric potential covariance matrix Σ_Φ.



(b) First pair of eigenvectors of Σ_Φ.

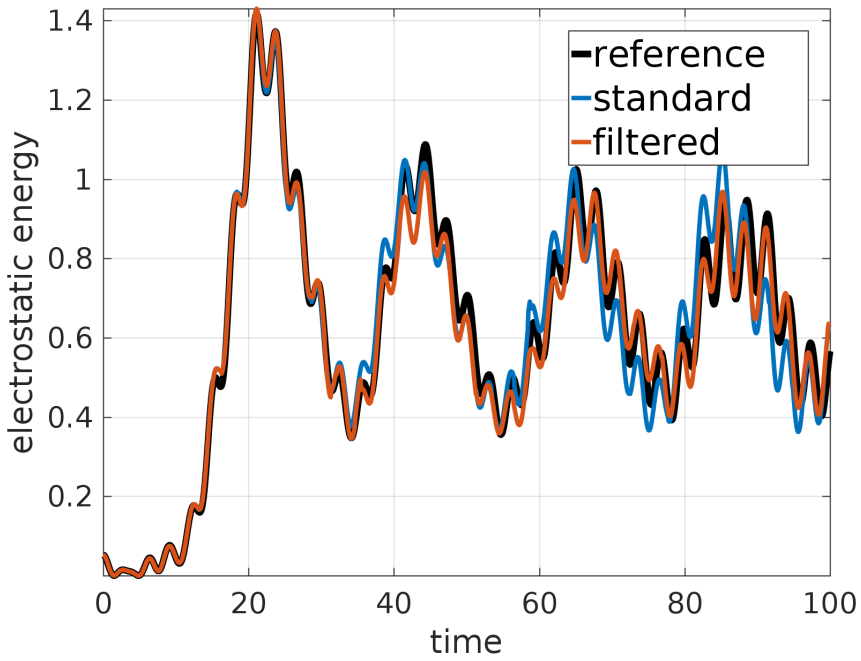


(c) Second pair of eigenvectors of Σ_Φ.

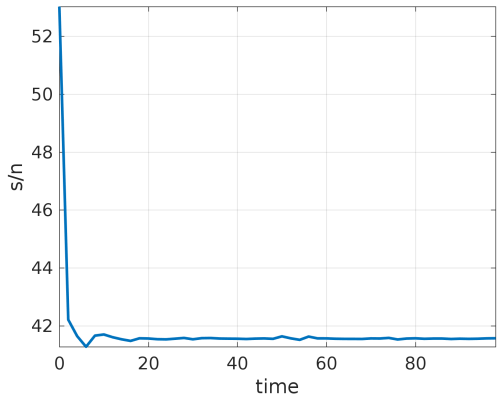


(d) Third pair of eigenvectors of Σ_Φ.

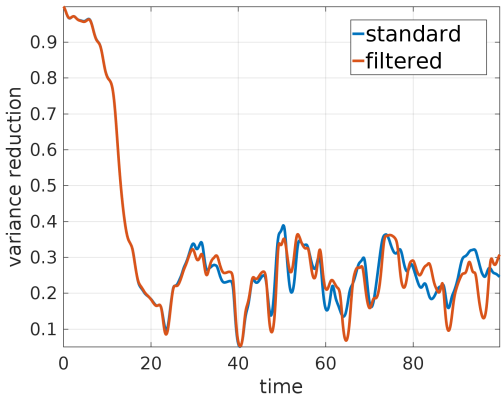
Figure 2.47.: Principal component analysis on the finite element coefficients of the electrostatic potential Φ. The obtained components resemble Fourier modes, which is expected in a periodic domain.



(a) Electrostatic energy



(b) Signal to noise ratio for principal component filter.



(c) Variance reduction by the control variate f_0.

Figure 2.48.: Principal component filtering of first four components $N_{pc} = 4$ for the bump-on-tail instability. The filter reduces successfully the background noise level, yielding a better agreement with the reference solution (a). When the control variate looses its efficiency the signal to noise ratio drops (b). Furthermore the filtering does not influence the (in)efficiency of the control variate (c).

every particle (x_k, v_k) with the absolute value of its contribution $|V_(k, j)|$ to the jth principal component, gives an overview in phase space. In a good signal we expect spectral clustering, therefore it is naturally to take the sum over these contributions weighted with the respective singular value, e.g. the first two components.
For the bump-on-tail instability, see fig. 2.49 the particles contribute to the mode, independent of their velocity. Yet more interesting for a KEEN wave [131], the major contribution to the potential comes from the particles around the forming vortex. For parameters and highly resolved phase space plots, see [132].
Both simulations were performed with the δf method $N_p = 10^5, N_{\text{fem}} = 32$ and the initial velocity distribution as a control variate. Although all the simulation particles were used, it is also possible to select a subset of particles and do the necessary calculations only for them, which should massively speed up the SVD.

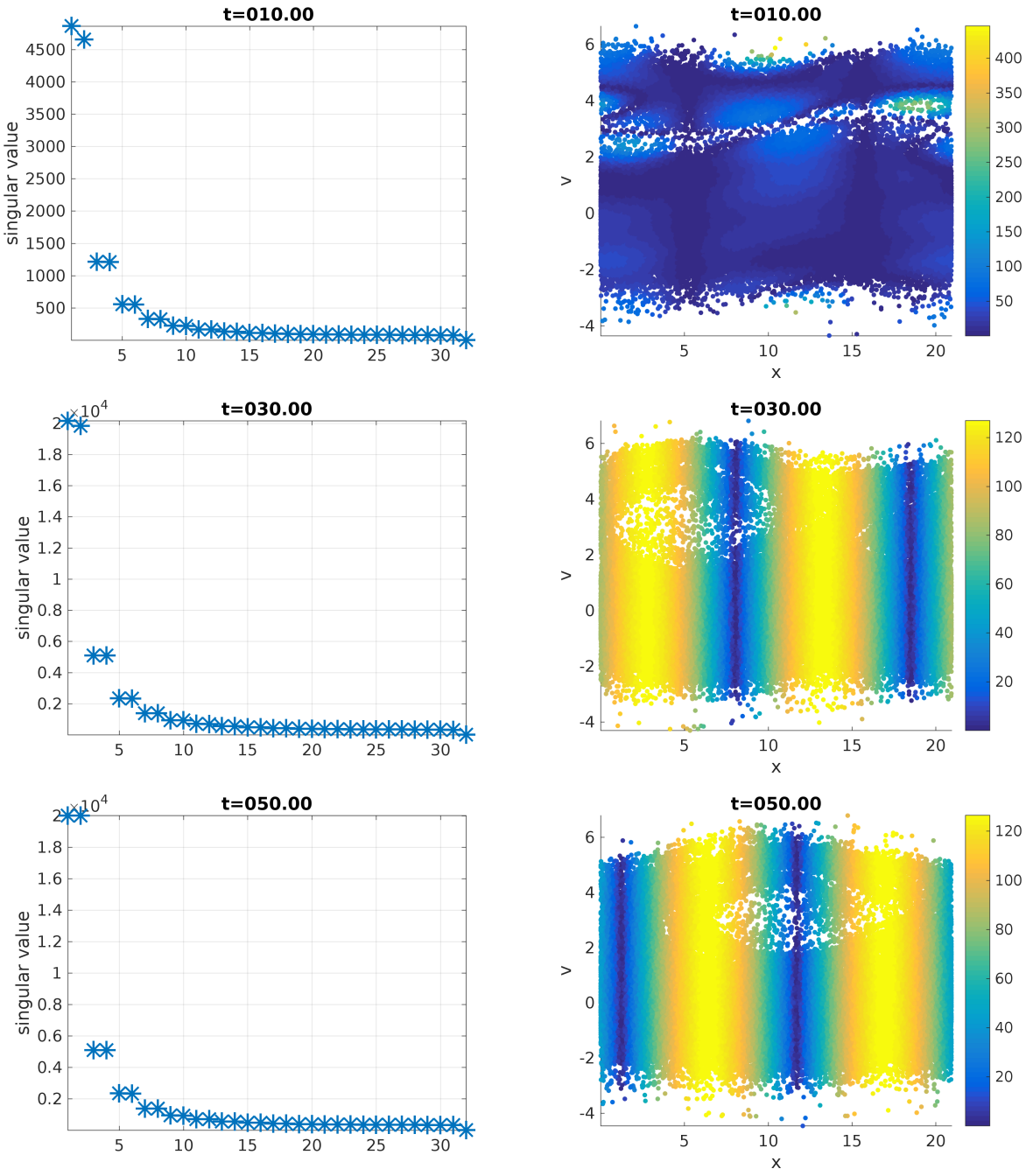


Figure 2.49.: First two leading right eigenvectors of Ψ_Φ for the bump-on-tail instability.

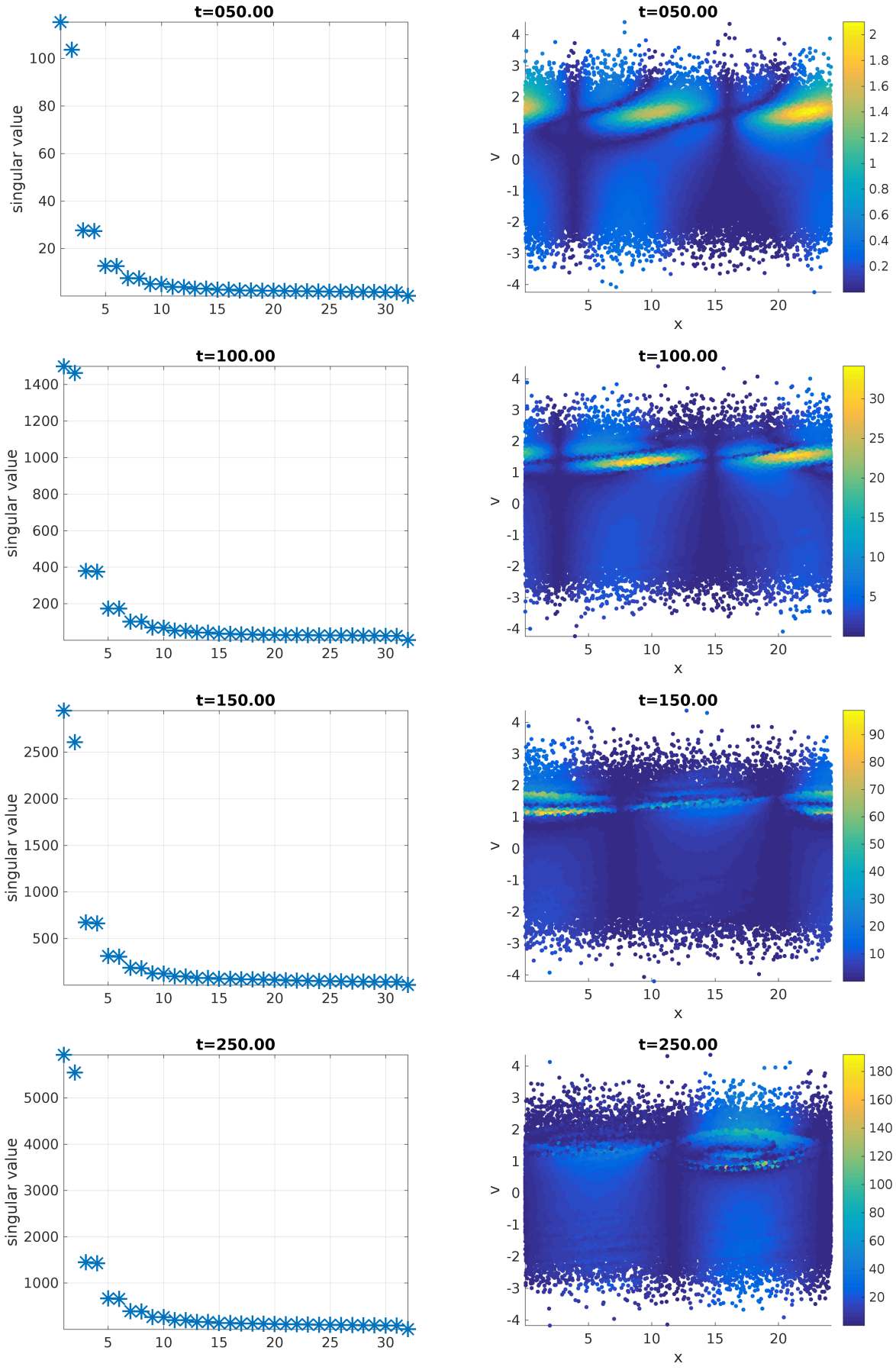


Figure 2.50.: First two leading right eigenvectors of Ψ_Φ for the KEEN Wave.

2.7.5. Unstructured finite elements and multi-scale methods

With a magnetic field of strength B_0 in z-direction the six dimensional Vlasov equation can be reduced to four dimensions.

$$v \times B(x,t) = v \times \begin{pmatrix} 0 \\ 0 \\ B_0(x,t) \end{pmatrix} = \begin{pmatrix} v_2 \\ -v_1 \\ 0 \end{pmatrix} B_0(x,t) \tag{2.374}$$

Because there is no cross product in two dimensions, we denote

$$v \wedge B(x,t) = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \wedge B(x,t) = \begin{pmatrix} v_2 \\ -v_1 \end{pmatrix} B_0(x,t). \tag{2.375}$$

Following [24] we use a scaled version of the four dimensional Vlasov equation given in eqn. (2.376).

$$\epsilon\, \partial_t f + \begin{pmatrix} v_x \\ v_y \end{pmatrix} \cdot \begin{pmatrix} \partial_x f \\ \partial_y f \end{pmatrix} + \left[\begin{pmatrix} E_x(x,y,t) \\ E_y(x,y,t) \end{pmatrix} + \frac{1}{\epsilon} \begin{pmatrix} v_y \\ v_x \end{pmatrix} B_0(x,t) \right] \cdot \begin{pmatrix} \partial_{v_x} f \\ \partial_{v_y} f \end{pmatrix} = 0 \tag{2.376}$$

For $\epsilon = 1$, eqn. (2.376) becomes the standard Vlasov equation. Here we consider non-neutral electrons, which means the right hand side of the Poisson equations consists only of the electron charge density.

$$E(x,y,t) = -\nabla\Phi(x,y,t), \quad -\Delta\Phi(x,y,t) = \iint_{\mathbb{R}^2} f(x,y,v_y,v_y)\ \mathrm{d}(v_x,v_y) \tag{2.377}$$

This non-neutral configuration yields a Kelvin-Helmholtz instability, which exhibits a more turbulent behavior than our standard two dimensional Vlasov phase-space. Hence in combination with unstructured finite elements it is particularly interesting to investigate the particle noise in such a situation. But for a strong magnetic field the gyromotion at the gyrofrequency $\omega = B$ becomes a very small time scale, which is expensive to resolve. Therefore, a limiting model for large homogeneous B is considered.

The two dimensional guiding center model

The two dimensional vorticity eqn. (2.378), often referred to as a guiding center model, emerges as the asymptotic limit $\epsilon \to 0$ of the four dimensional Vlasov–Poisson system under a strong magnetic field [24, 133]. This is often referred to as a reduced fluid model of the kinetic Vlasov–Poisson system. A spatial plasma density f develops under the electric field E which arises as the gradient of the electric potential Φ. The potential Φ is the solution to a Poisson equation coupling the density f to the fields.

$$\partial_t f(x,y,t) - E_y(x,y,t)\partial_x f(x,y,t) + E_x(x,y,t)\partial_y f(x,y,t) = 0 \tag{2.378}$$

$$E(x,y,t) = (E_x(x,y,t), E_y(x,y,t)) = -\left(\partial_x\Phi(x,y,t), \partial_y\Phi(x,y,t)\right) \tag{2.379}$$

$$-\triangle\,\Phi(x,y,t) = f(x,y,t) \tag{2.380}$$

The corresponding characteristics are given in eqn. (2.381).

$$\frac{\mathrm{d}}{\mathrm{d}t}X(t) = -E_y(X(t),V(t),t) \text{ and } \frac{\mathrm{d}}{\mathrm{d}t}Y(t) = E_x(X(t),V(t),t) \tag{2.381}$$

Due to the divergence form of eqn. (2.378) the mass $\int f(x,y,t)\mathrm{d}(x,y)$ is conserved. We aim for the most simple discretization thus linear Lagrange finite elements defined on triangles are chosen for the discretization of the Poisson equation. The computational domain is the

unit disc and we use the MATLABs PDE toolbox to generate an unstructured triangulation. The particle mesh coupling needs the nodal basis functions on every triangle which are for the linear Lagrange finite elements the restriction of the barycentric coordinates onto the respective triangle. For a triangle given by three points $(x_i, y_i) \in \mathbb{R}^2,\ i = 1, 2, 3$ and a point $(x, y) \in \mathbb{R}^2$ the three barycentric coordinates $\lambda_i(x),\ i = 1, 2, 3$ are defined in eqn. (2.382). Here A denotes the area of the triangle.

$$\lambda_1(x, y) := \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{2A} \tag{2.382}$$

$$\lambda_2(x, y) := \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{2A} \tag{2.383}$$

$$\lambda_3(x, y) := \frac{(y_1 - y_2)(x - x_1) + (x_2 - x_1)(y - y_1)}{2A} \tag{2.384}$$

$$A := \frac{1}{2}\left[(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)\right] \tag{2.385}$$

Note that the third coordinate is directly obtained by $\lambda_3 = 1 - \lambda_2 - \lambda_1$ because it holds that

$$1 = \lambda_1(x, y) + \lambda_2(x, y) + \lambda_3(x, y) \quad \text{for all } (x, y) \in \mathbb{R}^2. \tag{2.386}$$

If all barycentric coordinates are non-negative the position x lies within the triangle. The negative barycentric coordinate Later we need to evaluate the variance of the Potential where at every point the product $\psi_i(x, y)\psi_j(x, y)$ of basis functions is needed for all i, j. Fortunately here the barycentric coordinates fulfill the relation (2.387) such that the variance of any quantity at the mesh nodes equals merely the diagonal entries on the corresponding covariance matrix.

$$\lambda_i(x_j, y_j) = \delta_{i,j} \tag{2.387}$$

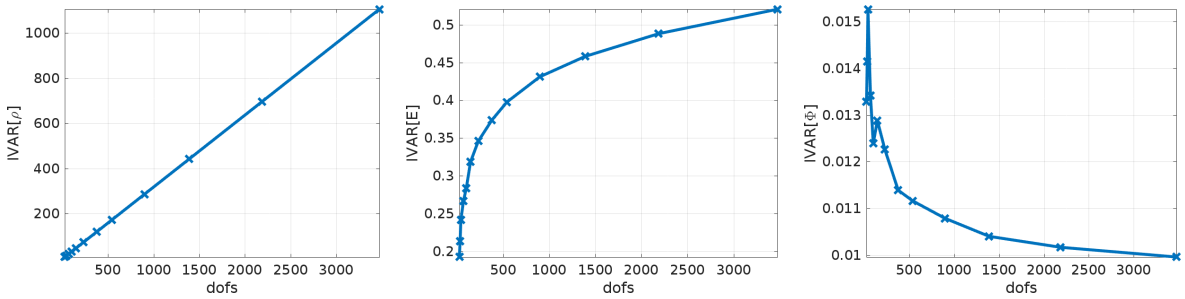
For the electric field the piecewise derivatives of the potential are needed thus the piecewise gradients of the barycentric coordinates are defined in (2.388). These gradients are constant and can therefore be stored during the simulation.

$$\nabla\lambda_1(x, y) := \frac{1}{2A}\begin{pmatrix} y_2 - y_3 \\ x_3 - x_2 \end{pmatrix} \tag{2.388}$$

$$\nabla\lambda_2(x, y) := \frac{1}{2A}\begin{pmatrix} (y_3 - y_1) \\ (x_1 - x_3) \end{pmatrix} \tag{2.389}$$

$$\nabla\lambda_3(x, y) := \frac{1}{2A}\begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \end{pmatrix} \tag{2.390}$$

We see that given a standard library for the meshing and matrix assembly the particle mesh coupling is implemented in very few steps. Except that the particles have to be located in the triangles, which is difficult. Elegant solutions can use structured hexagons [134] or field aligned triangles [135][p.606]. On unstructured grids a particle can be located in a triangle by subsequently following the most negative barycentric coordinate of the current triangle [136]. Since this algorithm uses the last known position it is quite efficient [137] and hence used here, although more robust variants are available nowadays [138] including better treatment of boundary conditions. See [139] for a comprehensive overview.
As the only test-case we consider the Diocotron instability in the unit disc, which also can be observed directly in nature [140] and has been extensively simulated in the community [133, 141, 134, 142, 24, 23].
The unit disc is best described in polar coordinates, which is used in eqn. (2.391) to describe



(a) charge density (b) electric field (c) electric potential

Figure 2.51.: Integrated variances for a constant charge density $\rho(r,\theta) = 1$ in the unit disc under increasing degrees of freedom for linear Lagrange finite elements.

the initial condition.

$$f(r,\theta) = \begin{cases} 1 + \epsilon\cos(k\theta) & \text{for } r \in [r_-, r_+] \\ 0 & \text{else} \end{cases} \tag{2.391}$$

$$r_- = 0.4,\ r_+ = 0.5,\ \epsilon = 0.05,\ k = 5 \tag{2.392}$$

A thin ring of electrons is initialized with a small ϵ perturbation of the kth mode. This simulation uses the fourth order Runge Kutta scheme with time-step $\Delta t = 0.25$, $N_p = 50,000$ RQMC particles and 4252 elements (triangles) resulting in $N_f = 2191$ degrees of freedom. In Cartesian coordinates, this means with respect to the Lebesgue measure, the sampling density g is proportional to f up to a constant.
But first we consider only the Poisson equation with a completely constant right hand side $f(r,\theta) = \rho(r,\theta) = 1$. By calculating the integrated variances for the L^2 projection the constant increase in variance with the degrees of freedom can be observed in fig. 2.51a. Given the fact that we only use first order elements, this is devastating since a lot of elements are required to resolve the stiff Laplace operator. But the Laplace operator damps the higher modes, refer to fig. 2.51b, which yields a much better behavior for the electric field E which is actually needed in the characteristics, eqn. (2.381). Surprisingly the integrated variance for the potential seems to decrease asymptotically to a constant value with increasing degrees of freedom, see (2.51c). So we can conclude that the Laplacian performs quite a strong regularization on the noisy right hand side. Figure 2.52 shows the evolution of the Diocotron instability, where the fives vortices emerge in the nonlinear phase. Here the difference between the noisy density estimation in is remarkable. We observe in fig. 2.53 and fig. 2.54b that the variance on the potential is much lower than on the charge density and that there is only a minor increase after the linear phase. Also the integrated variances in fig. 2.54a and fig. 2.54b seem to be bounded and only oscillate with the rotation of the vortices. This means that over time we do not require more particles as the density evolution undergoes many more non-linearities.

Multilevel Monte Carlo for asymptotically preserving schemes

We preferred the two dimensional guiding center model over a four dimensional Vlasov–Poisson equation because it is costly to resolve the fast gyromotion in the fully kinetic model. But if kinetic effects are of interest the reduced fluid model is not an option. Of course, fluid-kinetic hybrid models can recover effects of both scales [143]. It was also tried to improve kinetic simulations by fluid models [144]. Thus the first stochastic thought is to use the solution of the guiding center eqn. (2.378) as a control variate for eqn. (2.376) via

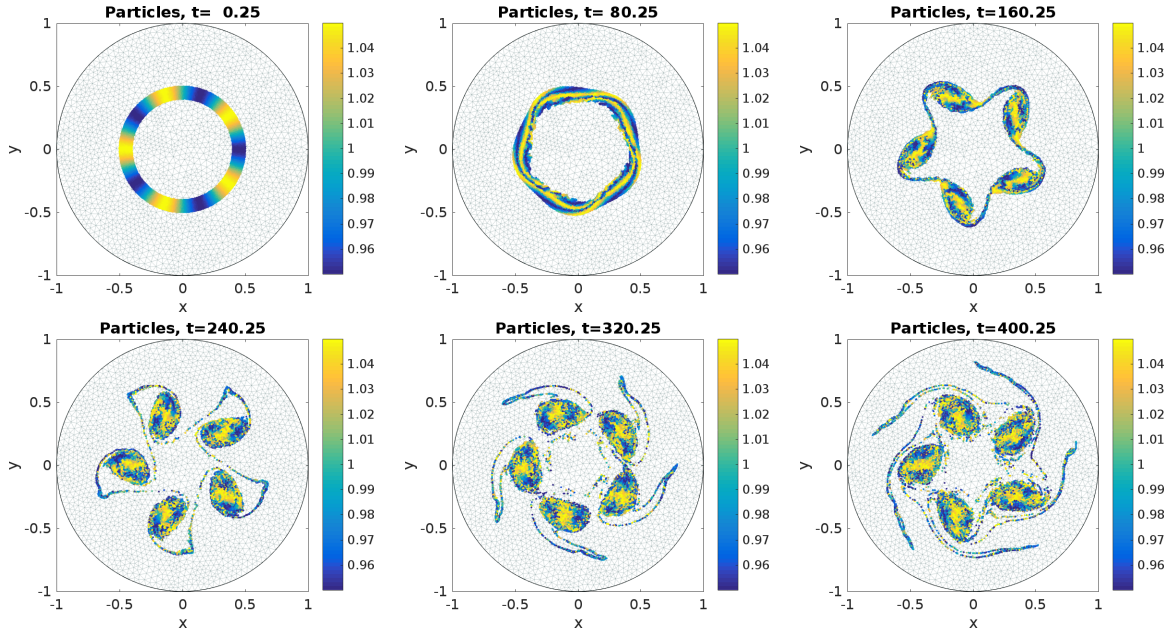


Figure 2.52.: Lagrangian particles carrying the initial value of the distribution function approximating the Diocotron instability.

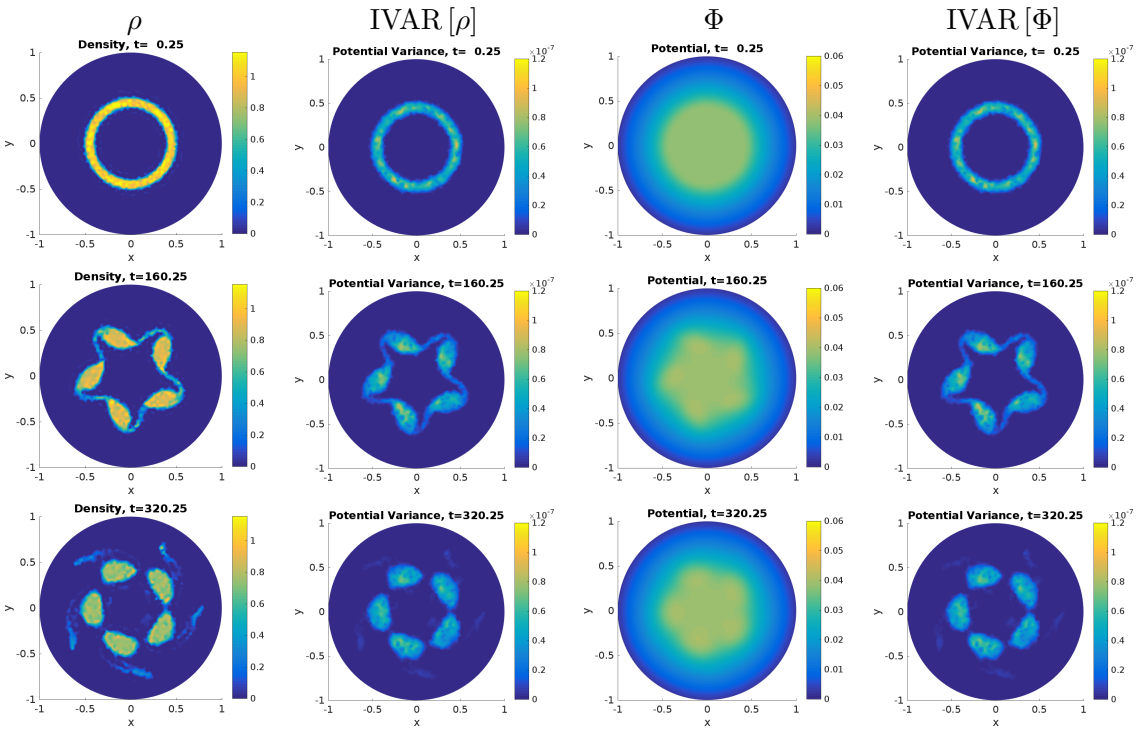
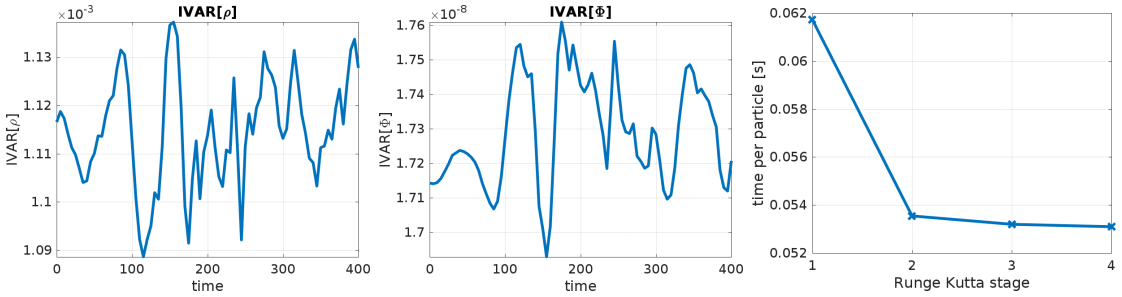


Figure 2.53.: Variances of the finite element estimator for the charge density and the potential next to the estimators themselves.



(a) IVAR of charge density (b) IVAR of potential (c) Duration of particle location per particle for every stage of the Runge Kutta integrator

Figure 2.54.: Integrated variances of the charge density and the potential over time for the nonlinear Diocotron instability.

the δf approach. This has several severe difficulties. The two dimensional density from eqn. (2.378) has to be brought up to a four dimensional one in order to be used as a control variate. This can be done by multiplication with a local Maxwellian. But even if the entire solution of eqn. (2.378) is available on a grid, the local Maxwellian might just not be a good approximation because if it would one does not need the kinetic model after all. Additionally the exact support $\mathrm{supp} g$ of the sampling density on the grid has to be known, which is not the case for the isolated vorticies of the Diocotron instability. Another problem is that the kinetic model has to influence the fluid model because over long time they might just drift apart. Therefore we cannot use the δf method with what we gained in the previous section. The control variate requires the exact knowledge of an expectation but for biased estimators it might be just enough to estimate the expectation with a cheaper but more biased estimator. This method is called multilevel Monte Carlo, see [16] for a very quick but sufficient introduction. Multiple levels are obtained by using different discretizations yielding a different bias. For density estimation on a grid the bias and the variance depend on the cell size h, which is used in [18] in combination with sparse grids and multiple levels. In general this can be treated and optimized like the normal control variates [145]. Here our dominating bias is the time step because of the fluid and kinetic time scale. For Vlasov equations involving collisions [17] provides a suitable multi-grid scheme in time. But here the time scales are so vastly different, that we need an integrator that is capable of solving the kinetic model for a small time step and the fluid model for a large time step, which we found in asymptotically preserving schemes [24]. For small $\epsilon \ll 1$ eqn. (2.376) approaches the to the fluid model, but the $\frac{B}{\epsilon}$ term becomes stiff. The first order integrator given in eqn. (2.393). can solve eqn. (2.376) including the stiff term for a large time-step whilst recovering the asymptotic model [24]. On the other hand the original Vlasov equation is solved for $\epsilon = 1$ but then a smaller time-step is needed.

$$\begin{cases} x^{n+1} &= x^n + \frac{\Delta t}{\epsilon} v^{n+1} \\ v^{n+1} &= v^n + \frac{\Delta t}{\epsilon} \left(\frac{v^{n+1}}{\epsilon} \wedge B(x_n, t_n) + E(x_n, t_n) \right) \end{cases} \tag{2.393}$$

Thus the discretization error is denoted by ϵ which makes our model much cheaper and allows us to use more particles. Suppose we have decided on a suitable scale separation by the choice of two parameters $\epsilon_0 \ll \epsilon_1 = 1$ and $\Delta t_0 \gg \Delta t_1$. Here the first level is the original Vlasov equation and the zeroth level the fluid model. Let $E_{\epsilon,\Delta t}$ denote the electric field obtained by using particles advanced by the scheme (2.393) using the scale ϵ and the time-step Δt. Combining the large time-step Δt_0 with the full Vlasov equation $\epsilon_1 = 1$ is prohibitive, yet

the small time-step Δt_1 can be used when integrating the fluid model. We suppose the fluid scheme is converged on the time-step Δt_0 such that it is safe to assume

$$E_{\epsilon_0,\Delta t_0} = E_{\epsilon_0,\Delta t} \text{ for } \Delta t < \Delta t_0. \tag{2.394}$$

By introducing Monte Carlo estimators we can use the fluid model as a control variate for obtaining the electric field $\hat{E}_{\epsilon_1,\Delta t_1}$ of the full Vlasov equation. The two level estimator lacking the optimization coefficient α reads

$$\mathbb{E}\left[\hat{E}_{\epsilon_1,\Delta t_1}\right] = \mathbb{E}\left[\hat{E}_{\epsilon_0,\Delta t_0}\right] + \mathbb{E}\left[\hat{E}_{\epsilon_1,\Delta t_1} - \hat{E}_{\epsilon_0,\Delta t_1}\right]. \tag{2.395}$$

Due to the larger time-step much more particles can be used on $\hat{E}_{\epsilon_0,\Delta t_0}$ while still being cheaper than $\hat{E}_{\epsilon_1,\Delta t_1}$. The key ingredient is the representation of the fluid solution on the scale of the kinetic model $\hat{E}_{\epsilon_0,\Delta t_1}$. This allows us to use the fluid solution f_{ϵ_1} without going through the δf approach by subtracting actual values of the distribution function according to $\delta w = \frac{f_{\epsilon_1} - f_{\epsilon_0}}{g_{\epsilon_1}}$. In practice the fluid model is solved in order to obtain $E_{\epsilon_0,\Delta t_0}$. It can be calculated by a non-particle method as long as there is a corresponding particle discretization available in order to get $\hat{E}_{\epsilon_0,\Delta t_1}$. Then a set of markers on the Vlasov level is duplicated and both sets are advanced by eqn. (2.393). The original uses $\epsilon_1, \Delta t_1$ giving $\hat{E}_{\epsilon_1,\Delta t_1}$ while the cloned markers are advanced with the same time step but for another level $\epsilon_0, \Delta t_1$ yielding $\hat{E}_{\epsilon_0,\Delta t_1}$. Finally, eqn. (2.395) is calculated by projecting $\mathbb{E}\left[\hat{E}_{\epsilon_0,\Delta t_0}\right]$ onto the finer time grid via linear interpolation. The feedback to the fluid model is provided by substituting the predictor $\hat{E}_{\epsilon_0,\Delta t_0}$ with the corrector $\hat{E}_{\epsilon_1,\Delta t_1}$ on intersecting time points. Another approach is to estimate the entire fluid distribution function from the kinetic particles which is very costly and therefore, not feasible. Especially asymptotic integrator allow for multiple levels using a telescope sum

$$\mathbb{E}\left[\hat{E}_{\epsilon_N,\Delta t_N}\right] = \mathbb{E}\left[\hat{E}_{\epsilon_0,\Delta t_0}\right] + \sum_{n=1}^{N} \mathbb{E}\left[\hat{E}_{\epsilon_n,\Delta t_n} - \hat{E}_{\epsilon_{n-1},\Delta t_n}\right]. \tag{2.396}$$

To use multilevel Monte Carlo efficiently a precise relation between ϵ and suitable Δt has to be known, such that we skip this part. Multilevel time grid notation is cumbersome, such that we refer to [17] for a detailed write-up with proof of convergence. Let φ_t^ϵ denote the discrete flux corresponding to a discrete time integration of the characteristics (X^t, V^t) on the level ϵ over a time step Δt using the field E.

$$\left(X^{t+\Delta t}, V^{t+\Delta t}\right) = \varphi_{\Delta t}^{\epsilon_0}(X^t, V^t; E) \tag{2.397}$$

Then we define to stochastic processes for the characteristics, the fluid particles (X_0^t, V_0^t) and the kinetic particles (X_1^t, V_1^t). For both levels an electric field $E(x)$ as a function of x is obtained by a Monte Carlo estimator. We denote such an estimator using i.i.d. replicates of the random deviate X as

$$E(x) = \mathbb{E}\left[F(x, X)\right] \text{ and } E = \mathbb{E}\left[F(X)\right]. \tag{2.398}$$

This means $F(X_1^t)$ estimates $E(t)$ using the kinetic particles while $F(X_0^t)$ uses the fluid particles. The overall first order algorithm consists then of two steps. First the fluid particles are advanced on the fluid level with a large time step using a given field $E(t)$.

$$(X_0^{t+\Delta t_0}, V_0^{t+\Delta t_0}) = \varphi_{\Delta t_0}^{\epsilon}(X_0^t, V_0^t; E(t)) \tag{2.399}$$

This yields a first estimate for the field at $t + \Delta t_0$ using the temporary fluid particles

$$\hat{E}(t + \Delta t_0) = \mathbb{E}\left[F(\hat{X}_0^{t+\Delta t_0})\right]. \tag{2.400}$$

By linear interpolation between $\hat{E}(t+\Delta t_0)$ and $E(t)$ we define $\hat{E}(t+\Delta t)$ for any $\Delta t > 0$ by

$$\hat{E}(t+\Delta t) := E(t)\left(1 - \frac{\Delta t}{\Delta t_0}\right) + \frac{\Delta t}{\Delta t_0}\hat{E}(t+\Delta t_0). \tag{2.401}$$

In order to sample the fluid level onto the kinetic level, copies $\left(\hat{X}_1^t, \hat{V}_1^t\right)$ of the kinetic particles at time t, called hybrid particles, are independently evolved using not the self consistent field, but the interpolated projection from the fluid level $\hat{E}(t)$.

$$(\hat{X}_1^{t+n\Delta t_1}, \hat{V}_1^{t+n\Delta t_1}) = \varphi_{\Delta t_1}^{\epsilon_0}\left(\hat{X}_0^{t+(n-1)\Delta t_1}, \hat{V}_0^{t+(n-1)\Delta t_1}; \hat{E}(t+(n-1)\Delta t_1)\right), \quad n = 0, \dots, \frac{\Delta t_0}{\Delta t_1} \tag{2.402}$$

With the hybrid particles it is possible to sample the difference between the fluid and the kinetic model such that the discrete analog of the rather vague multilevel Monte Carlo eqn. (2.395) reads

$$E(t+n\Delta t_1) = \hat{E}(t+n\Delta t_1) - \mathbb{E}\left[F(X_1^{t+n\Delta t_1}) - F(\hat{X}_1^{t+n\Delta t_1})\right]. \tag{2.403}$$

The kinetic particles are then advanced using the right field $E(t+n\Delta t_1)$, which is then also fed back into the fluid model at the start of our algorithm in eqn. (2.399).

$$(X_1^{t+n\Delta t_1}, V_1^{t+n\Delta t_1}) = \varphi_{\Delta t_1}^{\epsilon_0}\left(X_0^{t+(n-1)\Delta t_1}, V_0^{t+(n-1)\Delta t_1}; E(t+(n-1)\Delta t_1)\right), \quad n = 0, \dots, \frac{\Delta t_0}{\Delta t_1} \tag{2.404}$$

In order to provide the correct feedback to the fluid level for higher order methods, the fluid particles have to be advanced again using the corrected field $E(t+\Delta t), \Delta t \in [0, \Delta t_0]$ leaving eqn. (2.399) only to be a predictor step.
The explicit version of the first order scheme in eqn. (2.393) for our discretization is given in eqn. (2.405).

$$\begin{cases} x^{n+1} &= x^n + \frac{\Delta t}{\epsilon} v_x^{n+1} \\ y^{n+1} &= y^n + \frac{\Delta t}{\epsilon} v_y^{n+1} \\ v_x^{n+1} &= \left[-\frac{B_0(x^n,y^n,t^n)}{\epsilon}\frac{\Delta t^2}{\epsilon^2}E_x(x^n,y^n,t^n) + \frac{\Delta t}{\epsilon}E_y(x^n,y^n,t^n) + \frac{\Delta t}{\epsilon}\frac{B_0(x^n,y^n,t^n)}{\epsilon}v_y^n + \frac{v_x^n}{\epsilon}\right] \\ &\quad \cdot \left[\frac{B_0(x^n,y^n,t^n)^2\Delta t^2}{\epsilon^4} + 1\right]^{-1} \\ v_y^{n+1} &= \left[-\frac{B_0(x^n,y^n,t^n)}{\epsilon}\frac{\Delta t^2}{\epsilon^2}E_x(x^n,y^n,t^n) + \frac{\Delta t}{\epsilon}E_y(x^n,y^n,t^n) - \frac{\Delta t}{\epsilon}\frac{B_0(x^n,y^n,t^n)}{\epsilon}v_x^n + \frac{v_y^n}{\epsilon}\right] \\ &\quad \cdot \left[\frac{B_0(x^n,y^n,t^n)^2\Delta t^2}{\epsilon^4} + 1\right]^{-1} \end{cases} \tag{2.405}$$

Examining the Jacobi determinant (2.406) of the discrete flux $\varphi(x, y, v_x, v_y, \Delta t)$ of the scheme (2.405) reveals that not only is the scheme extremely dissipative for small ϵ but also for $\epsilon = 1$.

$$\frac{1}{\frac{B(x,y)^2\Delta t^2}{\epsilon^4} + 1} < 1 \text{ for } B \neq 0 \tag{2.406}$$

Thus phase-space volume is lost causing trouble for the likelihood propagation and application of the standard δf method. The standard Boris algorithm is not symplectic but actually preserves phase space volume, see [31], such that we would prefer an asymptotic preserving scheme based on such an integrator. In the following computations the second order L-stable scheme from [24] was used, since the first order scheme is just too dissipative for $\epsilon = 1$. The fluid initial condition eqn. (2.391) is extended by an additional Maxwellian with temperature $T_e = 0.05$.

$$f(x, y, v_x, v_y, t = 0) = \rho(x, y, t = 0)\frac{1}{2\pi T_e}\mathrm{e}^{-\frac{v_x^2+v_y^2}{2T}}. \tag{2.407}$$

The parameters used here are $B = 10,\ \epsilon_1 = 1,\ \Delta t_1 = 0.01,\ N_{p,1} = 10^3, \epsilon_0 = 0.05, \Delta t_0 = 2,\ N_{p,0} = 10^5$ and $\epsilon = 0.3$. Because of the magnetic field we expect an oscillation in the energies caused by the upper hybrid oscillations at frequency

$$\omega = \sqrt{1 + B^2 + 3k^2 T_e}. \tag{2.408}$$

These oscillations can be seen in the kinetic energy and the electrostatic energy in fig. 2.56. The variance reduction in the kinetic part is also at an satisfactory high level. These values denote the lowest variance reduction in the last kinetic time step, when (X_1, V_1) and their fluid replicates $(\hat{X}_1, \hat{V}_1)$ are the most apart, such that even higher values can be expected. Although the method works, we are not fully satisfied because the schemes are so dissipative such that actual multiple levels seem not to be feasible. The obvious alternative is to use Boris for Vlasov–Poisson and some Runge Kutta scheme for the fluid model. Here we considered only Monte Carlo method but in general a coupling between Eulerian and Lagrangian particles works the same way circumventing the conventional δf approach. Therefore, the three ingredients needed for a general composition are

- Eulerian fluid model
- Lagrangian fluid model (using Eulerian fields)
- Lagrangian kinetic model.

The next step is application to an extension with non-homogeneous magnetic field [23] or directly Lagrangian or Eulerian four dimensional drift-kinetic for six dimensional Lagrangian kinetic model.

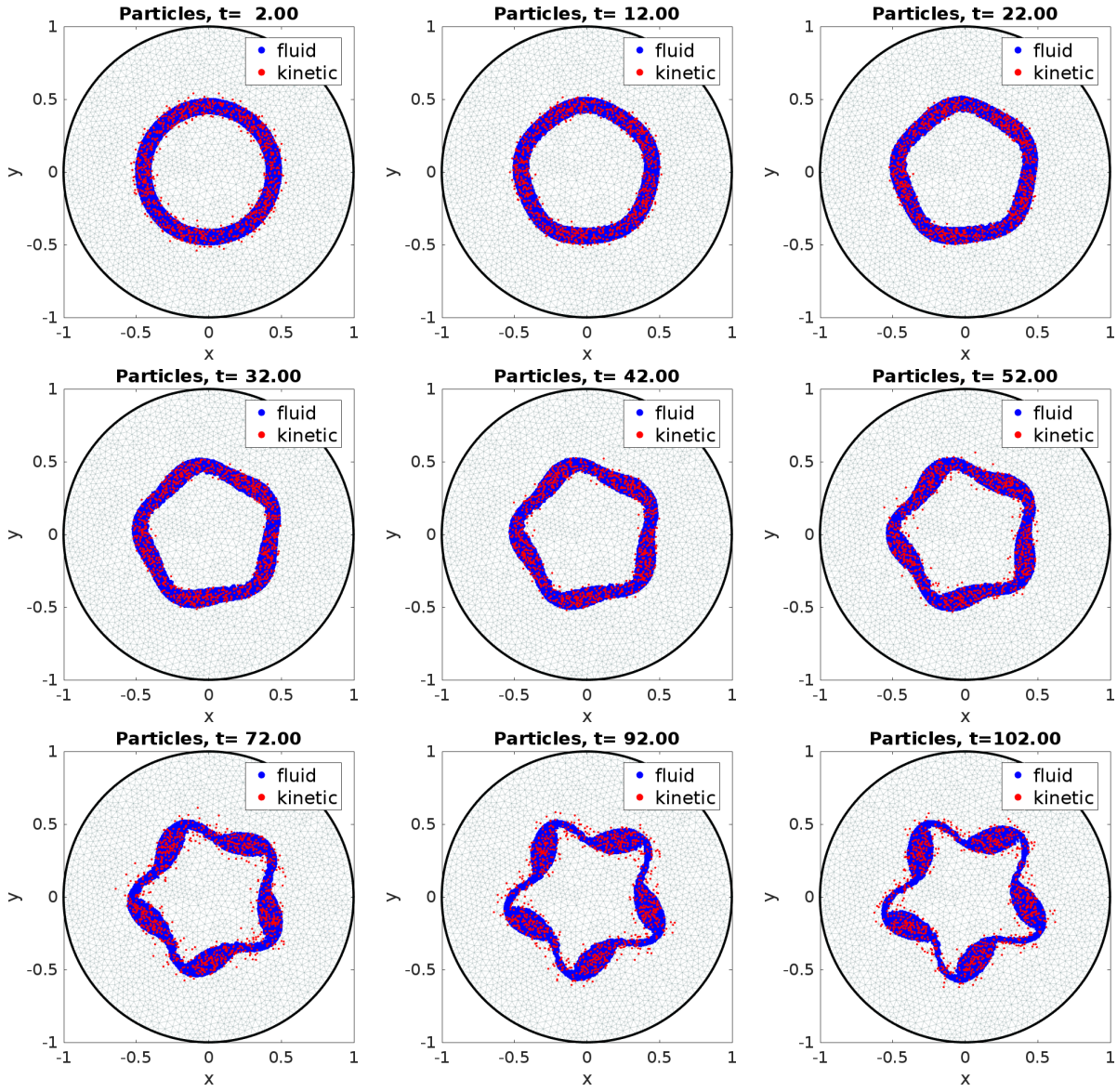
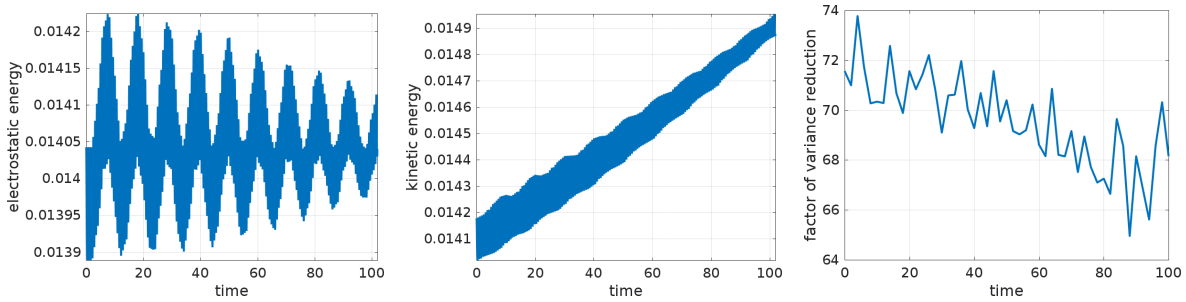


Figure 2.55.: Lagrangian particles in two level Monte Carlo for magnetized four dimensional Vlasov–Poisson using the same asymptotic preserving scheme with different scaling and time-step. Much more particles on the fluid level provide an effective control variate for few kinetic particles. The kinetic particles following the cyclotron motion blur the initial condition while the fluid particles are strongly confined.



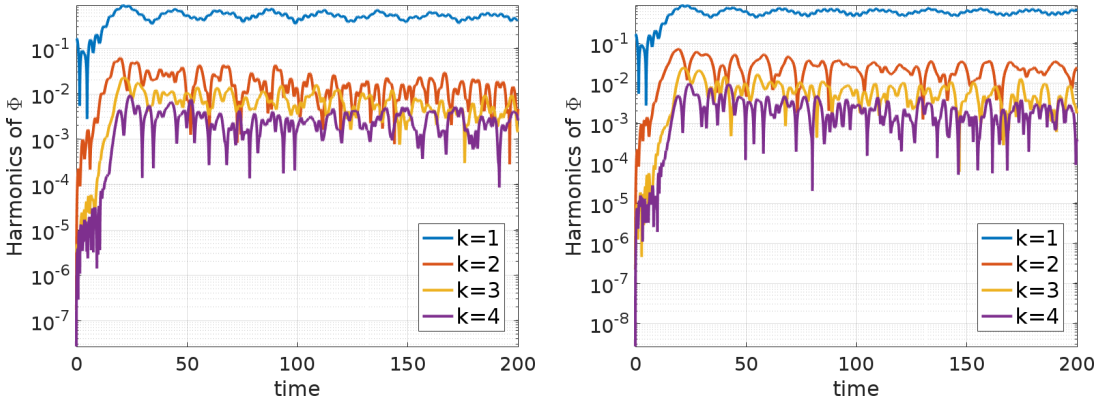
(a) Electrostatic energy (b) Kinetic energy (c) Factor of variance reduction

Figure 2.56.: Two level Monte Carlo for the Diocotron instability using a second order asymptotic preserving scheme.

Chapter 3.

Spectral particles

Spectral discretizations for the field solver of Particle-In-Cell codes are quite common, see [146, 147, 148]. Particle-In-Fourier (PIF), which uses Fourier modes instead of finite elements as basis functions, was already used to couple kinetic and MHD simulations in the HMGC code [149, 150], where the particles still have an unnecessary spatial shape. For periodic and especially periodic systems in beam physics hybrids between PIC and PIF have already been applied for slab and cylindrical geometry [151, 152]. Such an hybrid has also been used to evaluate the Fourier filtered fields in a gyrokinetic PIC simulation on GPU and CPU [153]. The first variational framework for PIF is provided in [10], where it also becomes clear that PIF conserves both energy and momentum where the latter one is not conserved in the standard finite element PIC. - Decreasing the field discretization error in PIC, thus, increasing its spectral fidelity avoids known unphysical instabilities [55]. Such schemes are commonly referred to as *dispersion free* and are becoming more common [154].
In the following, Particle-In-Fourier (PIF) is introduced, compared and combined with PIC in order to discretize various Vlasov systems. PIF uses a Fourier representation for the fields, which is efficient when the number of physically relevant Fourier modes remains small [149]. *Physically relevant* is a vague criterion, since it requires an a-priori deeper understanding of the solution. Additionally the Vlasov–Poisson system generates many small scales and filaments that have to be resolved such that Fourier filtering is mostly linked to some form of filamentation filtration [155, 156]. The smallest filament a grid based solver can resolve is a priori determined by its resolution, but Lagrangian particles can resolve large and small scales separately and are not subject to such limitations. Of course, the grid based field equations should resolve small scales in the fields, but it is absolutely not necessary. In many cases restricting the fields on few Fourier modes also results in much smaller structures, see fig. 3.1 and fig. 3.2, because the coupling between Vlasov and Poisson equation is of nonlinear nature. Although turbulence requires high resolution, the diffusive behavior rising from the averaging over the gyromotion in strongly magnetized plasmas is perfectly captured by the particle discretization of the Vlasov equation and does not depend on the resolution of the fields. It is known that PIC has issues resolving the high Fourier modes [75, 157, 158], such that our stochastic framework can help us to determine which Fourier modes are obscured by the Monte Carlo noise and which ones are still well captured, see fig. 3.3. In the following, PIF and other orthogonal methods provide an intuitive access in order to resolve the variance-bias balancing problem.



$N_f = 1$ $N_f = 16$

Figure 3.1.: The electric potential Φ in PIC simulations can be Fourier filtered such that only the first N_f Fourier modes remain in the electric field used for the advection of the particles. Although for $N_f = 1$ only the first Fourier mode contributes to the advection of the particles, the neglected modes can still be kept for diagnostic purposes. Here for a Bump-on-tail instability the remaining Fourier modes are also growing although they are not present in the discrete system itself. There is little difference between a simulation with one ($N_f = 1$) and many ($N_f = 16$) Fourier modes for $N_p = 10^6$, $N_{\text{fem}} = 32, \partial t = 0.05$ and cubic B-splines. This means that also small spatial structures in the density emerge although they are not resolved by the field discretization. This ultimately motivates the use of PIF.

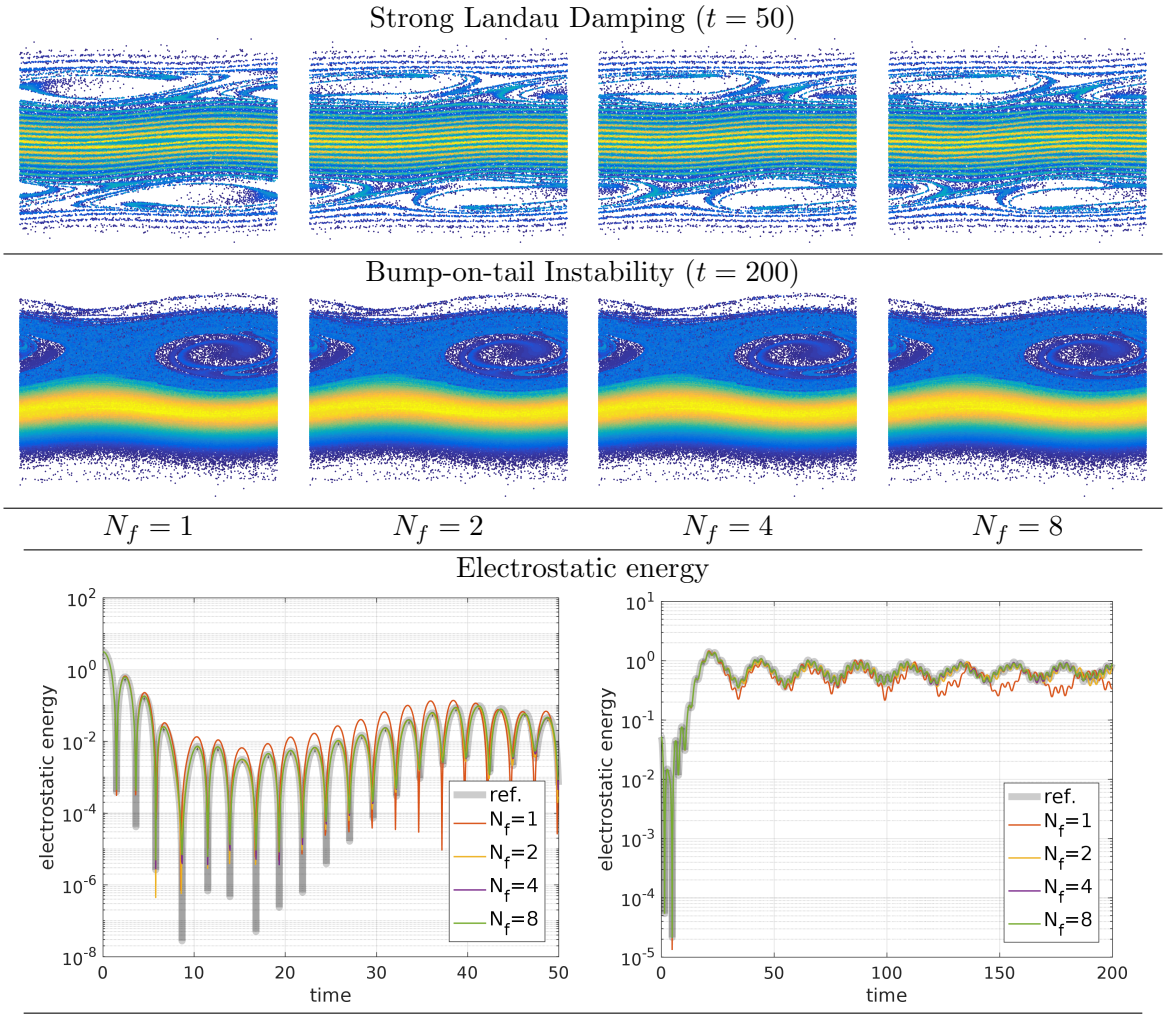
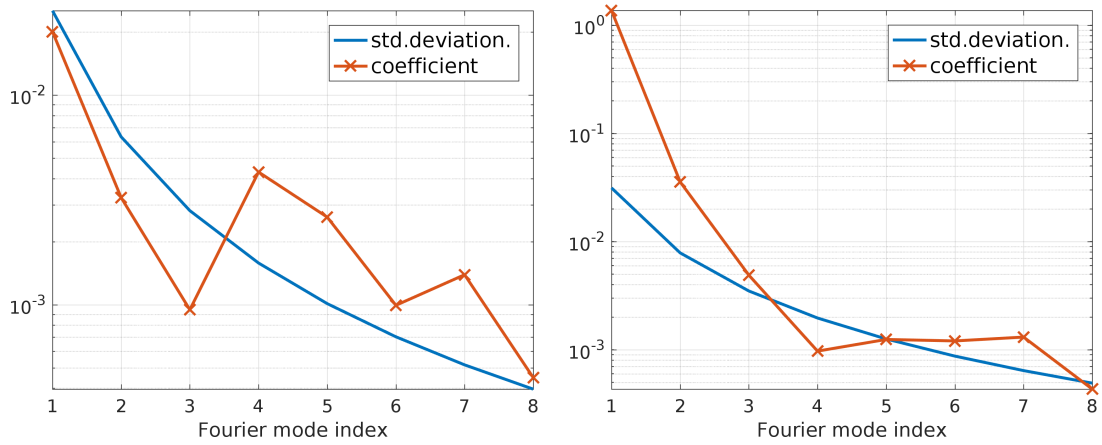


Figure 3.2.: Particle-In-Fourier (PIF) of the Vlasov–Poisson system for increasing number of Fourier modes N_f. The particle phase space exhibits small structures independent of the number of Fourier modes motivating the use of PIF. There remains also no visible difference in the electrostatic energy $N_f > 2$ compared to the reference solution. Note that in the linear phase of the Bump-on-tail instability ($t \leq 20$) the solution for $N_f = 1$ coincides with the reference. This is expected since only that single mode is excited and there is no nonlinear mode interaction at the beginning such that the second Fourier mode becomes only relevant at later times.



Strong Landau Damping ($t = 50$) Bump-on-tail Instability ($t = 200$)

Figure 3.3.: In PIF, the variance of each Fourier coefficient $\hat{\Phi}(k)$ of the potential Φ can be estimated straightforward compared to PIC, since the Fourier modes are orthogonal. Because of this orthogonality the variance itself is meaningful and contrary to PIC covariances do not have to be taken into account. If the absolute value of a coefficient $\hat{\Phi}_k$ is at the order of the sample standard deviation, the true value is obscured by noise such that either the number of particles N_p is massively increased or the respective Fourier mode can be neglected yielding a speed-up when using PIF. This can be generalized to other orthogonal series, which form eigenfunctions of the respective field equation.

3.1. Electrostatic electron model — Vlasov–Poisson/Ampère

We consider again the Vlasov equation with external magnetic field B, $\operatorname{div}(B) = 0$

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f - (E + v \times B) \cdot \nabla_v f = 0. \tag{3.1}$$

It can be coupled with the Poisson equation for the electric potential Φ. With the charge density $\rho = \int f \, \mathrm{d}v$ the Poisson equation is defined as

$$-\Delta\Phi = \rho - \rho_{\text{ion}}, \qquad E := -\nabla\Phi, \tag{3.2}$$

In the same way one can define the current density $j(x,t) = \int v f(x,v,t) \, \mathrm{d}v$ and solve instead of eqn. (3.2) the Ampère equation

$$\partial_t E(x,t) = j(x,t) - j_{\text{ion}}(x) \tag{3.3}$$

in order to obtain the evolution of the electric field. If not specified otherwise we set $j_{\text{ion}} = 1$, which makes Vlasov–Poisson and Vlasov–Ampère equivalent in one dimension. We already know from the previous introduction that eqn. (3.1) describes a conservation law, which is solved by the methods of characteristics.

3.1.1. Density estimation by Fourier transform

The natural way of solving the Poisson equation in a periodic domain is by Fourier transform, but first we have to Fourier transform random particle densities. In a domain of length L the wave vector k corresponding to the nth Fourier mode is defined as $k = \frac{2\pi}{L} n$ for $n \in \mathbb{Z}$. For the canonical case $L = 2\pi$ the wave vector corresponds to the nth Fourier mode, such that for the sake of notation k is used to denote the k^{th} Fourier mode. For every mode k the Fourier coefficients $\tilde{\rho}(k,t)$ of the charge density $\rho(x,t)$ are then obtained by the Fourier transform:

$$\tilde{\rho}(k,t) := \frac{1}{L} \int_{\mathbb{R}} \int_0^L \mathrm{e}^{-\mathrm{i}kx} f(x,v,t) \, \mathrm{d}x\mathrm{d}v. \tag{3.4}$$

The charge density itself can be expressed as a Fourier series, see eqn. (3.5). In a numerical simulation this Fourier series is truncated such that just a finite number of Fourier modes k is regarded.

$$\rho(x,t) = \sum_k \tilde{\rho}(k,t) \mathrm{e}^{\mathrm{i}kx} \tag{3.5}$$

In L^2 the Fourier modes $\left(x \mapsto \mathrm{e}^{\mathrm{i}kx}\right)_{k\in\mathbb{Z}}$ form an orthogonal series, which means that they are orthogonal with respect to the L^2 scalar product

$$\frac{1}{L} \int_0^L (\mathrm{e}^{ikx})^\dagger \mathrm{e}^{\mathrm{i}lx} \, \mathrm{d}x = \frac{1}{L} \int_0^L \mathrm{e}^{\mathrm{i}(l-k)x} \, \mathrm{d}x = \delta_{l,k}. \tag{3.6}$$

The plasma density f can be described by a stochastic process $Z(t) = (X(t), V(t))$ combined with a weight $W = \frac{f(X(t),V(t),t)}{g(X(t),V(t),t)}$, such that the Fourier transform of the charge density in eqn. (3.5) is rewritten as

$$\begin{aligned} \tilde{\rho}(k,t) &= \frac{1}{L} \int_{\mathbb{R}} \int_0^L \mathrm{e}^{-\mathrm{i}kx} f(x,v,t) \, \mathrm{d}x\mathrm{d}v \\ &= \frac{1}{L} \int_{\mathbb{R}} \int_0^L \mathrm{e}^{-\mathrm{i}kx} \frac{f(x,v,t)}{g(x,v,t)} g(x,v,t) \, \mathrm{d}x\mathrm{d}v = \frac{1}{L} \mathbb{E}\left[\mathrm{e}^{-\mathrm{i}kX(t)} W\right]. \end{aligned} \tag{3.7}$$

Fourier transforming random deviates is actually a major tool in stochastic theory, where the characteristic function of a random deviate X is defined as

$$\varphi_X(t) = \mathbb{E}\left[\mathrm{e}^{\mathrm{i}tX}\right]. \tag{3.8}$$

Given a number of samples the Monte Carlo estimator for the expectation in eqn. (3.7) can also be obtained by inserting the Klimontovich density f_p into the original integral in eqn. (3.4).

$$\begin{aligned}\tilde{\rho}(k,t) \approx \hat{\tilde{\rho}}(k,t) &= \frac{1}{L}\int_{\mathbb{R}}\int_0^L \mathrm{e}^{-\mathrm{i}kx} f_p(x,v,t)\ \mathrm{d}x\mathrm{d}v \\ &= \frac{1}{L}\int_{\mathbb{R}}\int_0^L \mathrm{e}^{-\mathrm{i}kx}\frac{1}{N_p}\sum_{n=1}^{N_p}\delta(x-X_n(t))\delta(v-V_n(t))w_n\ \mathrm{d}x\mathrm{d}v \\ &= \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n \mathrm{e}^{-\mathrm{i}kX_n(t)}\end{aligned} \tag{3.9}$$

Given the weighted samples $(X_n, V_n)_{n=1,\ldots,N_p}$, eqn. (3.9) provides a Monte Carlo estimator $\hat{\tilde{\rho}}(k,t)$ for every Fourier mode $\tilde{\rho}(k,t)$ of the charge density ρ. Inserting these estimators into the Fourier series for ρ, see eqn. (3.5), yields the unbiased density estimator $\hat{\rho}(x,t)$

$$\rho(x,t) \approx \hat{\rho}(x,t) = \sum_k \hat{\tilde{\rho}}(k,t)\mathrm{e}^{\mathrm{i}kx}. \tag{3.10}$$

Unbiased, because the Fourier modes were not truncated, and using the exact identity (3.5), we obtain

$$\mathbb{E}\left[\hat{\rho}(x,t)\right] = \sum_k \mathbb{E}\left[\hat{\tilde{\rho}}(k,t)\right]\mathrm{e}^{\mathrm{i}kx} = \sum_k \tilde{\rho}(k,t)\mathrm{e}^{\mathrm{i}kx} = \rho(x,t). \tag{3.11}$$

But in reality we have to truncate somewhere resulting in the Fourier series approximation

$$\rho(x,t) \approx \sum_{k=-N_f}^{N_f} \tilde{\rho}(k,t)\mathrm{e}^{\mathrm{i}kx} \tag{3.12}$$

and the biased density estimator $\hat{\rho}$ given in a single expression as

$$\rho(x,t) \approx \hat{\rho}(x,t) = \sum_{k=-N_f}^{N_f} \hat{\tilde{\rho}}(k,t)\mathrm{e}^{\mathrm{i}kx} = \frac{1}{L}\frac{1}{N_p}\sum_{k=-N_f}^{N_f}\sum_{n=1}^{N_p} w_n \mathrm{e}^{\mathrm{i}k[x-X_n(t)]}. \tag{3.13}$$

Biased, because the Fourier modes are truncated and the bias that is, the difference between the expectation of the estimator and the quantity it is supposed to approximate, is nonzero and reads

$$\left|\rho(x,t) - \mathbb{E}\left[\hat{\rho}(x,t)\right]\right| = \left|\sum_k \tilde{\rho}(k,t)\mathrm{e}^{ikx} - \sum_{k=-N_f}^{N_f}\tilde{\rho}(k,t)\mathrm{e}^{\mathrm{i}kx}\right| = \sum_{\substack{k>N_f \\ k<-N_f}} \tilde{\rho}(k,t)\mathrm{e}^{\mathrm{i}kx}. \tag{3.14}$$

It is, however, more meaningful to consider the integrated squared bias

$$\int_0^L \left|\rho(x,t) - \mathbb{E}\left[\hat{\rho}(x,t)\right]\right|^2\ \mathrm{d}x = L\sum_{k<-N_f, k>N_f} |\tilde{\rho}(k,t)|^2. \tag{3.15}$$

Thus, the bias consists only of the truncated Fourier modes. Orthogonal spectral series like Fourier, Chebyshev and Legendre series have the nice property that the absolute value of

the coefficients $\tilde{\rho}(k,t)$ decreases with increasing k. Boyd [54][pp. 50] uses this geometric convergence to conclude that the truncation error is in the order of the last coefficient. Applying this rule of thumb here means the bias can be estimated using the last coefficient. But since we couple stochastic and deterministic methods the mean squared error (MSE) is better suited to describe the error of the approximation $\hat{\rho}$.

$$\begin{aligned}\mathrm{MSE}\left[\hat{\rho}(x,t)\right] &= \mathbb{E}\left[\left|\hat{\rho}(x,t)-\rho(x,t)\right|^2\right] = \frac{\mathbb{V}\left[\hat{\rho}(x,t)\right]}{N_p} + \left|\rho(x,t)-\mathbb{E}\left[\hat{\rho}(x,t)\right]\right|^2 \\ &= \underbrace{\frac{\mathbb{V}\left[\hat{\rho}(x,t)\right]}{N_p}}_{variance} + \underbrace{\left|\sum_{\substack{k>N_f \\ k<-N_f}} \tilde{\rho}(k,t)\mathrm{e}^{ikx}\right|^2}_{bias^2}\end{aligned} \tag{3.16}$$

Instead of this point-wise description we integrate again, such that the mean integrated squared error (MISE) reads

$$\begin{aligned}\mathrm{MISE}\left[\hat{\rho}\right] &= \int_0^L \mathbb{E}\left[\left|\hat{\rho}(x,t)-\rho(x,t)\right|^2\right]\,\mathrm{d}x = \frac{\mathbb{V}\left[\hat{\rho}(x,t)\right]}{N_p} + \left|\rho(x,t)-\mathbb{E}\left[\hat{\rho}(x,t)\right]\right|^2 \\ &= \underbrace{\frac{\mathrm{IVAR}\left[\hat{\rho}(t)\right]}{N_p}}_{integrated\ variance} + \underbrace{L\sum_{\substack{k>N_f \\ k<-N_f}} \left|\tilde{\rho}(k,t)\right|^2}_{integrated\ bias^2}.\end{aligned} \tag{3.17}$$

The integrated variance can be estimated by covariance propagation in the same manner as it was done for Particle-In-Cell. The ultimate goal is of course to balance bias and variance, where another rule of thumb emerges from the orthogonality of the Fourier modes. For any estimated Fourier series, Fourier coefficients which are smaller than their variance can be truncated. We will consider this in more detail later, but be reminded that the strength of the orthogonal series density estimation lies within the accessible control over the variance bias relation. Now that we can project from a marker density onto a spectral k-grid and back, we can continue with the solution of the field equations.

3.1.2. Fourier transform of Ampère and Poisson equation

The Poisson equation for electrons with a constant ion background reads

$$-\Delta\Phi(x,t) = 1-\rho(x,t), \tag{3.18}$$

and its Fourier transform is given as

$$-(\mathrm{i}k)^2\tilde{\Phi}(k,t) = -\tilde{\rho}(k,t). \tag{3.19}$$

The constant ion background cancels with the average electron density, which was a notational hassle for finite elements. But in Fourier space this just means that the $k=0$ Fourier mode is dropped, such that the solution to eqn. (3.19) is merely a scalar multiplication according to eqn. (3.20).

$$\tilde{\Phi}(k,t) = \frac{\tilde{\rho}(k,t)}{(\mathrm{i}k)^2} \quad \Rightarrow \quad \Phi(x,t) = \sum_{k\neq 0}\frac{\tilde{\rho}(k,t)}{(\mathrm{i}k)^2}\mathrm{e}^{\mathrm{i}kx} \tag{3.20}$$

Something quite commonly known is that: *Fourier methods do not scale.*
Although there have been great and successful efforts to implement scalable Fourier transforms [159], the $\mathcal{O}(N\log(N))$ transform of function values on a grid is never going to be so

embarrassingly parallel as the standard Monte Carlo estimator. Or is it? The combination of both Fourier transform and Monte Carlo estimation according to eqn. (3.7) scales better for sure. Also the $\mathcal{O}(N)$ Poisson solve is then merely a negligible scalar multiplication. Thus, maybe we should refine our statement to: *Fourier transforms do not scale.*
The equations of motions require the electric field E, which is obtained as

$$E(x,t) = -\nabla\Phi(x,t) \;\Rightarrow\; \tilde{E}(k,t) = -\mathrm{i}k\;\tilde{\Phi}(k,t) \;\Rightarrow\; E(x,t) = -\sum_{k\neq 0}\frac{\tilde{\rho}(k,t)}{\mathrm{i}k}\mathrm{e}^{\mathrm{i}kx}. \tag{3.21}$$

The electrostatic energy as the L^2 norm of the electric field is

$$\mathcal{H}_E = \frac{1}{2}\int_0^L |E(x,t)|^2\,\mathrm{d}x = \frac{L}{2}\sum_{k\neq 0}|\tilde{E}(k,t)|^2 = \frac{L}{2}\sum_{k\neq 0}\left|\frac{\tilde{\rho}(k,t)}{k}\right|^2. \tag{3.22}$$

Inserting the particles yields an estimator for the electric field, such that the complete field solver can be expressed and implemented in one single equation.

$$\begin{aligned} E(x,t) \approx \hat{E}(x,t) &= -\sum_{k\neq 0}\frac{\hat{\tilde{\rho}}(k,t)}{\mathrm{i}k}\mathrm{e}^{\mathrm{i}kx} \\ &= -\frac{1}{L}\frac{1}{N_p}\sum_{k\neq 0}\sum_{n=1}^{N_p}\frac{w_n}{\mathrm{i}k}\mathrm{e}^{\mathrm{i}k[x-X_n(t)]} \end{aligned} \tag{3.23}$$

The compactness and simplicity of eqn. (3.23) is an enormous strength of the Particle-In-Fourier method in many ways. Complex derivations of new numerical methods can incorporate the Poisson solve in just one expression, which allows for much quicker development and testing of new schemes in contrast to Particle-In-Cell codes, where particle sorting, mass and stiffness matrices are additional factors of complexity. Let us proceed with the counterpart, the Vlasov Ampère system for which we need the Fourier transformed electron current density j with

$$j(x,t) = \int_{\mathbb{R}}\int_0^L v f(x,v,t)\;\mathrm{d}x\mathrm{d}v \quad\text{and}\quad \tilde{j}(k,t) = \int_{\mathbb{R}}\int_0^L v f(x,v,t)\mathrm{e}^{-\mathrm{i}kx}\;\mathrm{d}x\mathrm{d}v. \tag{3.24}$$

The Fourier transform of the Ampère equation does also incorporate an ion background, which is, without loss of generality, set to $j_{\mathrm{ion}}(x) = \frac{1}{L}\int_{\mathbb{R}}\int_0^L v f(x,v,t=0)\;\mathrm{d}x\mathrm{d}v$ such that the zeroth Fourier mode always drops out. In this very special case Vlasov–Poisson and Vlasov–Ampère are equivalent in a single dimension.

$$\partial_t\tilde{E}(k,t) = \tilde{j}(k,t) - \tilde{j}_{\mathrm{ion}}(k) \tag{3.25}$$

Since eqn. (3.25) depends on time, we have to deploy a time discretization, before we split the Vlasov and the Poisson equation from each other. In the Vlasov–Poisson particle discretization this means that first the given particles are used to obtain the electric field E and then this field is used to advance the particles according to their equations of motion. Thus, the naive approach is to do the same for the Ampère equation yielding

$$\tilde{E}(k,t+\Delta t) = \tilde{E}(k,t) + \int_t^{t+\Delta t}\tilde{j}(k,\tau)\mathrm{d}\tau = \tilde{E}(k,t) + \Delta t\;\tilde{j}(k,t). \tag{3.26}$$

But an important property is lost in eqn. (3.26). The factor $\frac{1}{k}$ in eqn. (3.21) damps the high modes, which is a crucial physical feature that cannot be found in eqn. (3.26). Since the high

modes are not damped, particle noise (variance) increases affecting the solution[1]. The correct Hamiltonian splitting of the Vlasov–Ampère system solves the problem, see eqn. (3.27).

$$\begin{cases} \partial_t f(x,v,t) + E(x,t)\partial_v f(x,v,t) = 0 \\ \partial_t E(x,t) = 0 \end{cases} \\ \begin{cases} \partial_t f(x,v,t) + v\partial_x f(x,v,t) = 0 \\ \partial_t E(x,t) = \int_{\mathbb{R}} v f(x,v,t)\ \mathrm{d}v - j_{\mathrm{ion}}(x) \end{cases} \tag{3.27}$$

The corresponding particle splitting using a stochastic process and the expectation reads

$$\begin{cases} \dot{X}(t) = 0, \\ \dot{V}(t) = E(X(t),t), \\ \partial_t \tilde{E}(k,t) = 0, \end{cases} \tag{3.28}$$

$$\begin{cases} \dot{X}(t) = V(t), \\ \dot{V}(t) = 0, \\ \partial_t \tilde{E}(k,t) = \mathbb{E}\left[V(t)\mathrm{e}^{-\mathrm{i}kX(t)}\right]. \end{cases} \tag{3.29}$$

The idea of the splitting is that the corresponding split steps are so simple that they can be solved exactly. In eqn. (3.28) the particle position and the field coefficients and therefore, the field itself stay constant over time, such that the ODE for the velocity $V(t)$ can be solved exactly according to

$$\begin{aligned} V(t+\Delta t) &= V(t) + \int_t^{t+\Delta t} \dot{V}(\tau)\ \mathrm{d}\tau \\ &= V(t) + \int_t^{t+\Delta t} \underbrace{E(X(\tau),\tau)}_{=E(X(t),t)\ \text{(constant)}}\ \mathrm{d}\tau \\ &= V(t) + \Delta t\ E(X(t),t). \end{aligned} \tag{3.30}$$

The second split step is a bit more involved. First note that the trajectories $X(t)$ are solved exactly by

$$X(\tau) = X(t) + \int_t^{\tau} \dot{X}(\tau')\ \mathrm{d}\tau' = X(t) + (\tau - t)V(t), \quad \tau \in [t, t+\Delta t]. \tag{3.31}$$

Using the trajectory of X during the split step allows us to solve the Ampère equation correctly by using the line integral and the indefinite integral of the Fourier mode, which

[1] On several occasions the author faced the statement that Vlasov–Ampère has more noise because it does not damp the high modes. Actually the variance can differ depending on the first moment of the particle velocity distribution, see the multi-species Vlasov–Maxwell example.

shows again how easy things are in Fourier space.

$$\begin{aligned}
\tilde{E}(k,t+\Delta t) &= \tilde{E}(k,t) + \int_t^{t+\Delta t} \partial_t \tilde{E}(k,\tau)\ \mathrm{d}\tau \\
&= \tilde{E}(k,t) + \tilde{E}(k,t) + \int_t^{t+\Delta t} \mathbb{E}\left[V(\tau)\mathrm{e}^{-\mathrm{i}kX(\tau)}\right]\ \mathrm{d}\tau \\
&= \tilde{E}(k,t) + \tilde{E}(k,t) + \mathbb{E}\left[\int_t^{t+\Delta t} V(\tau)\mathrm{e}^{-\mathrm{i}kX(\tau)}\ \mathrm{d}\tau\right] \\
&= \tilde{E}(k,t) + \mathbb{E}\left[\int_t^{t+\Delta t} V(\tau)\mathrm{e}^{-\mathrm{i}k[X(t)+(\tau-t)V(t)]}\ \mathrm{d}\tau\right] \\
&= \tilde{E}(k,t) + \mathbb{E}\left[\int_{X(t)}^{X(t)+\Delta t V(t)} \mathrm{e}^{-\mathrm{i}ks}\ \mathrm{d}s\right] \\
&= \tilde{E}(k,t) + \mathbb{E}\left[\left[\frac{1}{-\mathrm{i}k}\mathrm{e}^{-\mathrm{i}ks}\right]_{s=X(t)}^{s=X(t+\Delta t)}\right] \\
&= \tilde{E}(k,t) - \frac{1}{\mathrm{i}k}\mathbb{E}\left[\mathrm{e}^{-\mathrm{i}kX(t+\Delta t)} - \mathrm{e}^{-\mathrm{i}kX(t)}\right]
\end{aligned} \tag{3.32}$$

Now the $\frac{1}{\mathrm{i}k}$ factor, which damps the high modes, again appears in eqn. (3.32).

3.1.3. Variational aspects

PIF was introduced ad-hoc which provided an intuitive access but does not guarantee any conservation laws. Actually the same scheme can be derived by a discrete Euler–Lagrange principle, such that we recall some mechanisms from [10]. For a particle discretization

$$f_p(x,v,t) = \frac{1}{N_p}\sum_{n=1}^{N_p} w_n S(x-x_n)\delta(v-v_n), \tag{3.33}$$

with a spatially smoothing shape function S the discrete particle Lagrangian for Vlasov–Poisson reads

$$\begin{aligned}
&\mathcal{L}(x,\dot{x},v,\dot{v},\Phi,\dot{\Phi}) = \\
&\qquad \frac{1}{N_p}\sum_{n=1}^{N_p} w_n\left[x_n\dot{v}_n - \frac{1}{2}v_n^2 - \int S(\tilde{x}-x_n)\Phi(x)\ \mathrm{d}\tilde{x}\right] + \frac{1}{2}\int \left(\partial_{\tilde{x}}\Phi(\tilde{x})\right)^2 \mathrm{d}\tilde{x}.
\end{aligned} \tag{3.34}$$

The equations of motions with the field equations are obtained by the Euler–Lagrange principle from eqn. (3.34) as

$$\begin{aligned}
\dot{x}_n &= v_n, \\
\dot{v}_n &= \int S\left(x - x_n(t)\right)\partial_{\tilde{x}}\Phi(\tilde{x})\ \mathrm{d}\tilde{x}, \\
-\int \partial_x\Phi(x)(\partial_x\varphi(x))^\dagger\ \mathrm{d}x &= \frac{1}{N_p}\sum_{n=1}^{N_p} w_n \int S(\tilde{x}-x_n)\varphi(x)^\dagger\ \mathrm{d}\tilde{x}, \quad \forall\varphi.
\end{aligned} \tag{3.35}$$

Given a nontrivial shape function S, the Vlasov–Poisson system is mollified such that (3.34) and (3.35) correspond to the system (3.36).

$$\begin{aligned}
\partial_t f(x,v,t) + v\partial_x f(x,v,t) - E(x,t)\partial_v f(x,v,t) &= 0 \\
-\Delta\Phi(x,t) = 1 - \int_0^L\int_{-\infty}^{\infty} f(x-y,v,t)S(y)\ \mathrm{d}v\mathrm{d}y & \\
E(x,t) = -\int_0^L \partial_x\Phi(x-y,t)S(y)\ \mathrm{d}y &
\end{aligned} \tag{3.36}$$

In order to obtain our standard PIC, finite elements ψ_k are chosen as Ansatz for $\Phi = \sum \Phi_k \psi_k$ and the test-functions as $\varphi \in \{\psi_k\}$. Additionally the particle shape is restricted to a delta function $S(x) = \delta(x)$. For the classical PIF the choice of Fourier modes $\psi_k = \mathrm{e}^{\mathrm{i}kx}$ yields the discrete Lagrangian

$$\mathcal{L} = \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \left[x_n \dot{v}_n - \frac{1}{2} v_n^2 - \sum_k \frac{1}{L} \Phi_k \mathrm{e}^{\mathrm{i}kx_n} \right] + \frac{L}{2} \sum_k \frac{1}{k^2} \Phi_k \Phi_k^\dagger, \tag{3.37}$$

along with the equations of motion and the Fourier transformed Poisson equation

$$\dot{x}_n = v_n, \;\; \dot{v}_n = \sum_k \mathrm{i}k\Phi_k \mathrm{e}^{\mathrm{i}kx_n}, \;\; k^2 \Phi_k = \frac{1}{L} \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \mathrm{e}^{-\mathrm{i}kx_n}. \tag{3.38}$$

For a non-trivial particle shape function the most common choice is based on splines. The B-splines S_m of order m with cell size h are obtained by successive convolution of the rectangular function S_0

$$\begin{aligned} S_0(x) &= \begin{cases} \frac{1}{h} & \text{if } x \in \left(-\frac{h}{2}, \frac{h}{2}\right) \\ 0 & \text{else} \end{cases} \\ S_m(x) &= \underbrace{S_0 * \cdots * S_0}_{m+1 \text{ times}}(x) = S_0^{*(m+1)}(x) = \int_{-\infty}^{\infty} S_0(x-y) S_{m-1}(y) \, \mathrm{d}y. \end{aligned} \tag{3.39}$$

Note that by the convolution the support of the m^{th} order spline S_m increases with the degree m,

$$\operatorname{supp}(S_m) = \left[-h\frac{m+1}{2}, h\frac{m+1}{2} \right]. \tag{3.40}$$

Convolution is merely a multiplication in Fourier space, such that the Fourier transform of the B-splines is explicitly given as

$$\int_0^L S_m(x) \mathrm{e}^{-ikx} \, \mathrm{d}x = \left[\operatorname{sinc}\left(\frac{kh}{2}\right) \right]^{m+1}. \tag{3.41}$$

Here the unnormalized definition $\operatorname{sinc}(x) = \frac{\sin(x)}{x}$ is used, but most software uses the normalized convention $\operatorname{sinc}(x) = \frac{\sin(x\pi)}{x\pi}$. Inserting the B-splines and the Fourier modes into eqn. 3.35 yields a variant of PIF with finite particles with spatial extent $h(m+1)$.

$$\begin{aligned} \dot{x}_n &= v_n, \\ \dot{v}_n &= \sum_k \mathrm{i}k\Phi_k \left[\operatorname{sinc}\left(\frac{kh}{2}\right) \right]^{m+1} \mathrm{e}^{\mathrm{i}kx_n} \\ &= \sum_k \frac{1}{-\mathrm{i}k} \left[\operatorname{sinc}\left(\frac{kh}{2}\right) \right]^{2(m+1)} \frac{1}{L} \frac{1}{N_p} \sum_{m=1}^{N_p} w_m \mathrm{e}^{-\mathrm{i}k(x_n - x_m)} \\ k^2 \Phi_k &= \left[\operatorname{sinc}\left(\frac{kh}{2}\right) \right]^{m+1} \frac{1}{L} \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \mathrm{e}^{-\mathrm{i}kx_n} \end{aligned} \tag{3.42}$$

Because of the particle shape the initial condition f_0 is subject to an additional convolution resulting in $\tilde{f}_0$ given by

$$\tilde{f}_0(x,v) = \int_0^L f_0(y,v) S_m(x-y) \, \mathrm{d}y. \tag{3.43}$$

In order to obtain a consistent initial condition the deconvolution has to be incorporated into the initial condition. Observe that

$$1+\epsilon\cos(kx)=\int_0^L \big[1+\underbrace{\operatorname{sinc}\left(\frac{kh}{2}\right)^{-m}\epsilon}_{:=\tilde\epsilon}\cos(ky)\big]S_m(x-y)\,\mathrm{d}y, \tag{3.44}$$

which means it suffices to change the amplitude of the perturbation from ϵ to $\tilde\epsilon$ for the typical initial condition. Although the initial condition is modified correctly the outcome of the simulation depends on the particle shape and does not necessarily coincide with the original Vlasov–Poisson system, see fig. 3.4. For $m=1$ eqn. (3.42) was already derived in [10], where also the conservation of energy and momentum is discussed. Among the conserved quantities of the Vlasov–Poisson system is the momentum, which including the particle discretization reads

$$\int f(x,v,t)v\,\mathrm{d}x\mathrm{d}v=\frac{1}{N_p}\sum_{n=1}^{N_p}w_nv_n^t\int S(x)\,\mathrm{d}x. \tag{3.45}$$

For the classical finite difference PIC the symmetry in the charge projection scheme e.g., cloud in cell, to and from the grid yields the momentum conservation at the discrete level, but the conserved quantities such as energy are in general lost when those schemes are not derived from a variational principle [5]. For the canonical variational particle algorithm with particles shapes S, the condition (3.46) was derived in [10].

$$\int\Phi(x)\partial_x\psi_k(x)^\dagger\,\mathrm{d}x\int S(x-x_n)\psi_k(x)\,\mathrm{d}x=-\int\Phi(x)\psi_k^\dagger\,\mathrm{d}x\int S(x-x_n)\partial_x\psi_k(x)\,\mathrm{d}x \tag{3.46}$$

Here $(\psi_k)_{k=1,\dots}$ are orthogonal basis functions for the potential Φ satisfying the orthogonality

$$\int\psi_k(x)\psi_l(x)^\dagger\,\mathrm{d}x=\delta_{k,l}. \tag{3.47}$$

Equation (3.46) holds true for Fourier modes $\psi_k=\mathrm{e}^{\mathrm{i}kx}$ and, thus, momentum and energy are conserved. Unfortunately we do not know of any other basis for which this also is true. Instead of showing the translation invariance for the Lagrangian which is done in [10], the force a particle exerts on itself is calculated. Recall that given particles $(x_n)_{n=1,\dots,N_p}$ and weights $(w_n)_{n=1,\dots,N_p}$ in PIF, the electric field at the position x reads

$$\begin{aligned}E(x;x_1,\dots,x_{N_p},w_1,\dots,x_{N_p})=\\ 3\sum_{\substack{k=-N_f\\k\neq0}}^{N_f}\mathrm{e}^{\mathrm{i}kx}\frac{1}{\mathrm{i}k}\frac{1}{N_p}\sum_{n=1}^{N_p}w_n\mathrm{e}^{-\mathrm{i}kx_n}=\frac{1}{N_p}\sum_{\substack{k=-N_f\\k\neq0}}^{N_f}\frac{1}{\mathrm{i}k}\sum_{n=1}^{N_p}w_n\mathrm{e}^{\mathrm{i}k(x-x_n)}\\ =\frac{1}{N_p}\sum_{n=1}^{N_p}w_n\sum_{k=1}^{N_f}\left(\frac{1}{\mathrm{i}k}\mathrm{e}^{\mathrm{i}k(x-x_n)}+\frac{1}{-\mathrm{i}k}\mathrm{e}^{-\mathrm{i}k(x-x_n)}\right)=\frac{1}{N_p}\sum_{n=1}^{N_p}w_n\sum_{k=1}^{N_f}\frac{2}{k}\sin\left(k(x-x_n)\right).\end{aligned} \tag{3.48}$$

Evaluating the electric field (3.48) for a particle x_m eqn. (3.49) reveals that there is no particle self force.

$$\begin{aligned}E(x_m;x_1,\dots,x_{N_p},w_1,\dots,x_{N_p})&=\frac{1}{N_p}\sum_{n=1}^{N_p}w_n\sum_{k=1}^{N_f}\frac{2}{k}\sin\left(k(x_m-x_n)\right)\\ &=\underbrace{\sum_{k=1}^{N_f}\frac{2}{k}\frac{1}{N_p}w_m\underbrace{\sin\left(k(x_m-x_m)\right)}_{=0}}_{\textit{self force}}+\frac{1}{N_p}\sum_{\substack{n=1\\n\neq m}}^{N_p}w_n\sum_{k=1}^{N_f}\frac{2}{k}\sin\left(k(x_m-x_n)\right)\end{aligned} \tag{3.49}$$

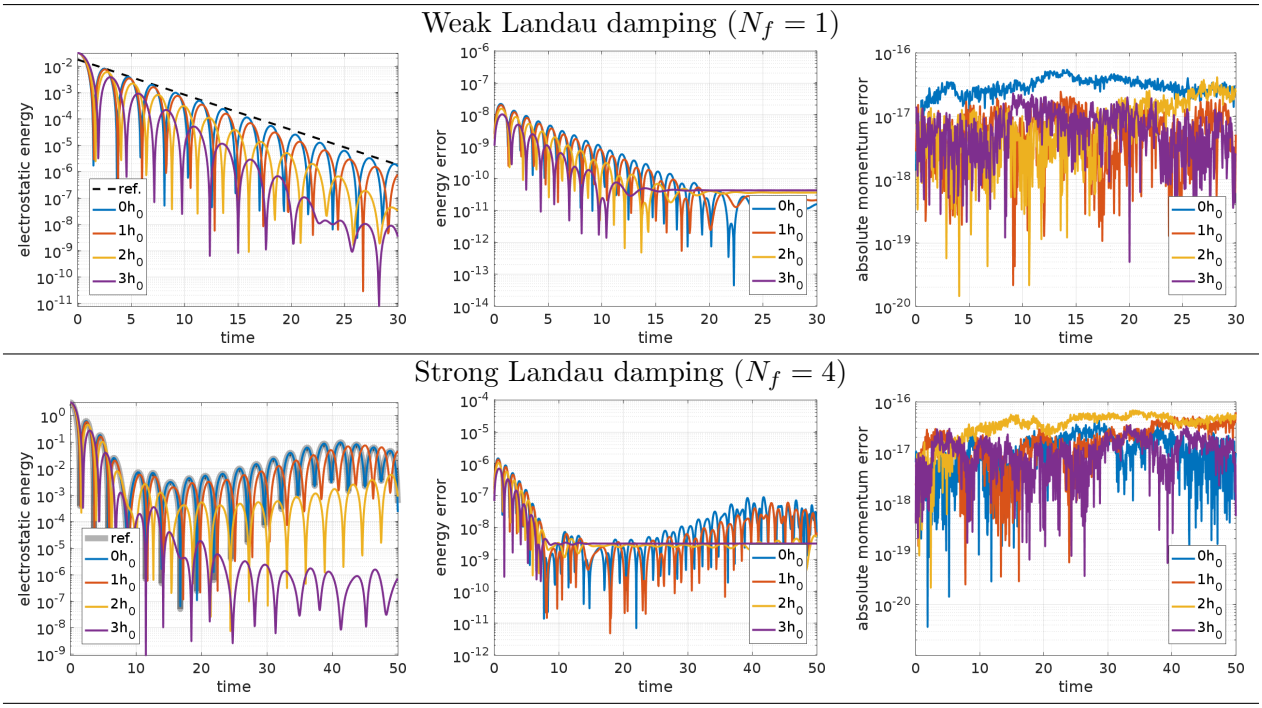


Figure 3.4.: PIF simulations of weak Landau damping using different quadratic B-spline particle shape functions S_2 of varying width h. A null width $h = 0$ results in the standard PIF. With increasing particle width, the solution clearly deviates from the reference although energy and momentum are conserved. ($h_0 = \frac{L}{4(2+1)}$, $N_p = 10^6$, $\Delta t = 0.05$, rk3s)

But this does not necessarily imply total momentum conservation when using control variates. In [160][p.2] it is already noted that there is an inconsistency in momentum conservation stemming from the control variate. For a control variate PIC the constant weights w_n are replaced by time dependent weights δw_n. As long as these weights do not change during a time step, there remains no particle self force although the total momentum conservation is violated by the changing weights.

3.1.4. Variance in PIF

We have already discussed the errors in the particle mesh coupling for PIC, such that all results obtained for PIC apply also for PIF by choosing the Fourier modes as basis functions $\left(\psi_m(x) = \mathrm{e}^{\mathrm{i}m\frac{2\pi}{L}x}\right)_{m=1,\dots,N_f}$ which yields diagonal mass and stiffness matrices

$$\mathcal{M}_{k,l} = \delta_{k,l}\frac{1}{L} \text{ and } \mathcal{K}_{k,l} = \delta_{k,l}\frac{1}{k^2 L}, \quad k,l = 1,\dots,N_f. \tag{3.50}$$

This simplicity allows us to directly state the variance of the Fourier coefficients of charge density, potential and electric field. Using independent identically distributed samples the sample variance for the Fourier coefficients reads

$$\mathbb{V}\left[\hat{\hat{\rho}}(k,t)\right] = \frac{1}{N_p}\frac{1}{L^2}\mathbb{V}\left[\mathrm{e}^{-\mathrm{i}kX(t)}W(t)\right]. \tag{3.51}$$

The variance for the charge density estimator in eqn. (3.13) given in eqn. (3.52) contains covariances of all Fourier modes.

$$\begin{aligned}\mathbb{V}\left[\hat{\rho}(x,t)\right] &= \mathbb{V}\left[\sum_{k=-N_f}^{N_f}\hat{\tilde{\rho}}(k,t)\mathrm{e}^{\mathrm{i}kx}\right] = \frac{1}{L^2N_p}\mathbb{V}\left[\sum_{k=-N_f}^{N_f}\mathrm{e}^{-\mathrm{i}kX(t)}W(t)\mathrm{e}^{\mathrm{i}kx}\right] \\ &= \frac{1}{L^2N_p}\sum_{k_1=-N_f}^{N_f}\sum_{k_2=-N_f}^{N_f}\mathbb{COV}\left[\mathrm{e}^{-\mathrm{i}k_1X(t)}W(t)\mathrm{e}^{ik_1x},\ 3\mathrm{e}^{-\mathrm{i}k_2X(t)}W(t)\mathrm{e}^{\mathrm{i}k_2x}\right] \\ &= \frac{1}{L^2N_p}\sum_{k_1=-N_f}^{N_f}\sum_{k_2=-N_f}^{N_f}\mathbb{COV}\left[\mathrm{e}^{-\mathrm{i}k_1X(t)}W(t),\ \mathrm{e}^{-\mathrm{i}k_2X(t)}W(t)\right]\mathrm{e}^{\mathrm{i}(k_1-k_2)x}\end{aligned} \tag{3.52}$$

Whether PIC or PIF is used, the coefficient covariance matrix is always dense. This matrix is needed in order to calculate more meaningful criterion — the integrated variance. Inserting (3.52) and using the orthogonality of the Fourier modes shows that the integrated variance in PIF is directly obtained by summing up the variances of each Fourier mode.

$$\begin{aligned}\mathrm{IVAR}\left[\hat{\rho}(t)\right] &= \int_0^L \mathbb{V}\left[\hat{\rho}(x,t)\right]\ \mathrm{d}x \\ &= \frac{1}{L^2N_p}\sum_{k_1=-N_f}^{N_f}\sum_{k_2=-N_f}^{N_f}\mathbb{COV}\left[\mathrm{e}^{-\mathrm{i}k_1X(t)}W(t),\ \mathrm{e}^{-\mathrm{i}k_2X(t)}W(t)\right]\underbrace{\int_0^L\mathrm{e}^{\mathrm{i}(k_1-k_2)x}\ \mathrm{d}x}_{L\delta_{k_1,k_2}} \\ &= \frac{1}{LN_p}\sum_{k=-N_f}^{N_f}\mathbb{V}\left[\mathrm{e}^{-\mathrm{i}kX(t)}W(t)\right] = \sum_{k=-N_f}^{N_f}L\mathbb{V}\left[\hat{\tilde{\rho}}(k,t)\right]\end{aligned} \tag{3.53}$$

The variances of the field coefficients are then obtained by scalar multiplication

$$\begin{aligned}\mathbb{V}\left[\hat{\tilde{\Phi}}(k,t)\right] &= \mathbb{V}\left[\frac{\hat{\tilde{\rho}}(k,t)}{\mathrm{i}k^2}\right] = \frac{1}{k^4}\left[\hat{\tilde{\rho}}(k,t)\right], \\ \mathbb{V}\left[\hat{\tilde{E}}(k,t)\right] &= \mathbb{V}\left[\frac{\hat{\tilde{\rho}}(k,t)}{-\mathrm{i}k}\right] = \frac{1}{k^2}\left[\hat{\tilde{\rho}}(k,t)\right],\end{aligned} \tag{3.54}$$

which results in the integrated variances

$$\mathrm{IVAR}\left[\hat{\Phi}\right] = \sum_{\substack{k=-N_f\\k\neq0}}^{N_f}\frac{L}{k^4}\mathbb{V}\left[\hat{\tilde{\rho}}(k,t)\right]\ \text{ and }\ \mathrm{IVAR}\left[\hat{E}\right] = \sum_{\substack{k=-N_f\\k\neq0}}^{N_f}\frac{L}{k^2}\mathbb{V}\left[\hat{\tilde{\rho}}(k,t)\right]. \tag{3.55}$$

3.1.5. Fourier filtering and aliasing in PIC

For the finite element PIC a series of N_f m^{th}-order $h = \frac{L}{N_f}$ wide B-splines was used as basis functions for the fields. Fourier transforming such a periodic series results in

$$\begin{aligned}u(x) &= \sum_{n=1}^{N_f}u_n\underbrace{S_m\left(x-nh-\bar{x}\right)}_{=\psi_n(x)} \\ \tilde{u}_k &= \frac{1}{L}\int_0^L u(x)\mathrm{e}^{-ikx}\ \mathrm{d}x = \frac{1}{L}\sum_{n=1}^{N_f}u_n\left[\mathrm{sinc}\left(\frac{kh}{2}\right)\right]^{m+1}\mathrm{e}^{-\mathrm{i}k(nh+\bar{x})}.\end{aligned} \tag{3.56}$$

For $\bar{x} = 0$ the first spline is centered at $x = 0$, but an equidistant periodic grid will have the first node at $x = 0$ such that the centered spline has to be shifted to $\bar{x} = h\frac{m+1}{2}$. Inserting the canonical choice $k = \frac{2\pi}{L}l,\ l = 0, \dots, N_f - 1$ and $h = \frac{L}{N_f}$ into eqn. (3.56) yields eqn. (3.57).

$$\tilde{u}_l = \mathrm{e}^{-2\pi\mathrm{i}\frac{\bar{x}}{L}l}\left[\operatorname{sinc}\left(\pi\frac{l}{N_f}\right)\right]^{m+1}\frac{1}{L}\underbrace{\sum_{n=1}^{N_f} u_n \mathrm{e}^{-2\pi\mathrm{i}\frac{l}{N_f}}}_{\text{discrete Fourier transform}} \tag{3.57}$$

Here it becomes clear that the Fourier modes $\tilde{u}$ can be obtained from the finite element coefficients (u_n) by a discrete Fourier transform on the coefficient vector and a corresponding scaling with the *sinc* function. This is a beneficial coincidence, since the involved finite element matrices are circulant Toeplitz matrices which can be diagonalized using the discrete Fourier transform (3.58) on the coefficient vectors, see also [161][p. 34].

$$\begin{aligned}(\mathcal{F})_{n,m} &= \mathrm{e}^{-2\pi\mathrm{i}\frac{nm}{N_f}}, \quad n, m = 0, \dots, N_f - 1\\ (\mathcal{F}^{-1})_{n,m} &= \frac{1}{N_f}\mathrm{e}^{-2\pi\mathrm{i}\frac{nm}{N_f}}, \quad n, m = 0, \dots, N_f - 1,\end{aligned} \tag{3.58}$$

If the mass and stiffness matrix are diagonalized by

$$M = F^{-1}D_M F \text{ and } K = F^{-1}D_K F \tag{3.59}$$

a Fourier filter is implemented by setting entries of the diagonal matrices D_M, D_K to zero, which correspond to the desired Fourier modes according to eqn. (3.57). This also applies for the inversion, where the constant Fourier mode is subtracted for neutrality. Thus, field solve and Fourier filter can be incorporated into the same process. Note that this form of filtering does not destroy the time symmetry such that the filtered energy is conserved and essentially nothing changed from the variational perspective expect the choice of basis functions. In the following we are interested up to which extent Fourier filtering reduces particle noise. Since the IVAR of the fields depends mainly on the structure of $f(x, v, t)$, one cannot draw a general conclusion without having any knowledge of f. Therefore, we restrict ourselves to an equilibrium case $\rho(x) = 1$ with a uniform sampling density $g(x) = \frac{1}{L}$. Equation (2.97) already holds a closed expression for the corresponding right hand side covariance, such that for PIC the Fourier filtered mass and stiffness matrices (3.59) can be used for covariance propagation. For PIF the Fourier coefficients and their variance are directly obtained by

$$\mathbb{E}\left[\hat{\tilde{\rho}}(n)\right] = \tilde{\rho}(n) = \frac{1}{L}\int_0^L \frac{\rho(x)}{g(x)}\mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx}g(x)\,\mathrm{d}x = \frac{1}{L}\int_0^L L\mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx}\frac{1}{L} = 0\,\mathrm{d}x \text{ for } n \neq 0 \tag{3.60}$$

and

$$\begin{aligned}\mathbb{V}\left[\hat{\tilde{\rho}}_n(t)\right] &= \frac{1}{N_p}\int_0^L \left(\frac{1}{L}\frac{\rho(x)}{g(x)}\mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx} - \tilde{\rho}(n)\right)\left(\frac{1}{L}\frac{\rho(x)}{g(x)}\mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx} - \tilde{\rho}(n)\right)^{\dagger} g(x)\,\mathrm{d}x\\ &= \frac{1}{N_p}\int_0^L \left(\frac{1}{L}L\mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx} - 0\right)\left(\frac{1}{L}L\mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx} - 0\right)^{\dagger}\frac{1}{L}\,\mathrm{d}x\\ &= \frac{1}{N_p}\int_0^L \mathrm{e}^{-\mathrm{i}\frac{2\pi}{L}nx+\mathrm{i}\frac{2\pi}{L}nx}\frac{1}{L}\,\mathrm{d}x = \frac{1}{N_p}.\end{aligned} \tag{3.61}$$

Note that the zeroth Fourier mode is constant and is not subject to any variance $\mathbb{V}\left[\hat{\tilde{\rho}}\right] = 0$. This only holds true for importance sampling where the variance of the weights is zero $\mathbb{V}[W] =$

0, or when a control variate is applied respectively. Inserting (3.61) into the previously obtained expressions for the integrated variance (3.55) results in

$$\mathrm{IVAR}\left[\hat{\rho}\right] = \sum_{n=1}^{N_f} 2L\mathbb{V}\left[\hat{\hat{\rho}}(n)\right] = \frac{2LN_f}{N_p} \xrightarrow{N_f\to\infty} \infty. \tag{3.62}$$

For an increasing number of Fourier modes, the integrated variance of the density estimate tends to infinity which means that the number of particles per mode have to be held constant in order to balance the background noise. Nevertheless this changes once we include the field solve. With eqn. (3.55) we obtain upper bounds for the integrated variance of the potential

$$\mathrm{IVAR}\left[\hat{\Phi}\right] = \sum_{n=1}^{N_f} 2\frac{L}{\left(\frac{2\pi}{L}n\right)^4}\mathbb{V}\left[\hat{\hat{\rho}}(n)\right] = \frac{2L^5}{(2\pi)^4 N_p} \underbrace{\sum_{n=1}^{N_f} \frac{1}{n^4}}_{\xrightarrow{N_f\to\infty}\frac{\pi^4}{90}} \xrightarrow{N_f\to\infty} \frac{L^5}{720N_p} \tag{3.63}$$

and the electric field used for the advection

$$\mathrm{IVAR}\left[\hat{E}\right] = \sum_{n=1}^{N_f} 2\frac{L}{\left(\frac{2\pi}{L}n\right)^2}\mathbb{V}\left[\hat{\hat{\rho}}(n)\right] = \frac{2L^3}{(2\pi)^2 N_p}\sum_{n=1}^{N_f}\frac{1}{n^2} \xrightarrow{N_f\to\infty} \frac{2L^3}{(2\pi)^2 N_p}\frac{\pi^2}{6} = \frac{L^3}{12N_p}. \tag{3.64}$$

Therefore, contrary to a plain density estimate, the Monte Carlo noise is limited by the field equation. For this case fig. 3.5 contains a comparison of the integrated variance for PIF and PIC for different spline order and number of cells. Compared to the B-splines the PIF has the highest IVAR. Also with increasing support, order of the splines, the IVAR increases slightly. The correct variance propagation allows us to see that IVAR$[E]$ is bounded with respect to the number of basis functions, whereas IVAR$[\rho]$ is not for both PIC and PIF.

Contrary to grid based integration, in Monte Carlo integration the number of samples required for a certain precision on a Fourier coefficient does not depend on the mode number, which is indicated already in eqn. (3.61). To further investigate this, we consider importance sampling of a uniform mode k with amplitude $\epsilon > 0$ according to

$$\rho(x) = 1 + \epsilon\cos(kx), \quad g(x) = \frac{\rho(x)}{L}. \tag{3.65}$$

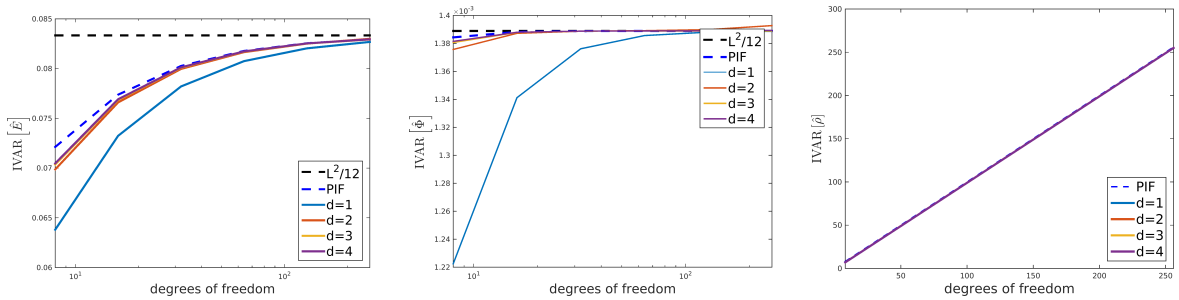
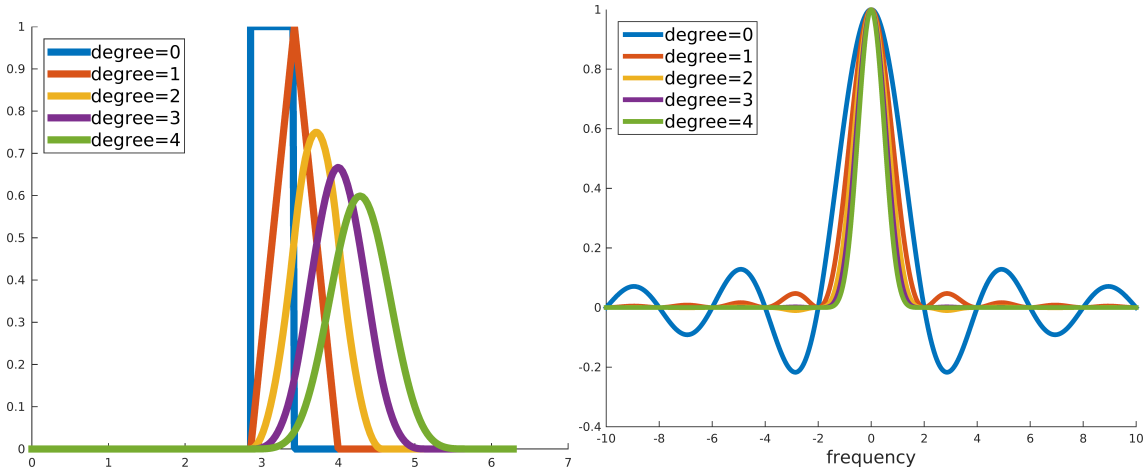


Figure 3.5.: Integrated variance of electric field, potential and charge density for $\rho(x) = 1$, $g(x) = \frac{1}{L}$. Although every additional Fourier mode yields more variance, the damping by the Laplace operator bounds the increase. With varying B-spline degree d PIC approximates the variance of PIF while both have the same asymptotic bound.



(a) Arbitrary degree B-splines $S_m(x)$

(b) Fourier transform $\tilde{S}_m(k)$ of the B-splines $S_m(x)$

Figure 3.6.: B-splines and their Fourier transform. While S_m (a) has compact spatial support the Fourier transform $\tilde{S}_m$ (b) is globally supported in Fourier space. This causes aliasing, since every Fourier mode contributes to one B-spline. Higher order splines decay much faster in Fourier space, such that the aliasing is suppressed.

The expectation and the variance of the k^{th} Fourier coefficient then read

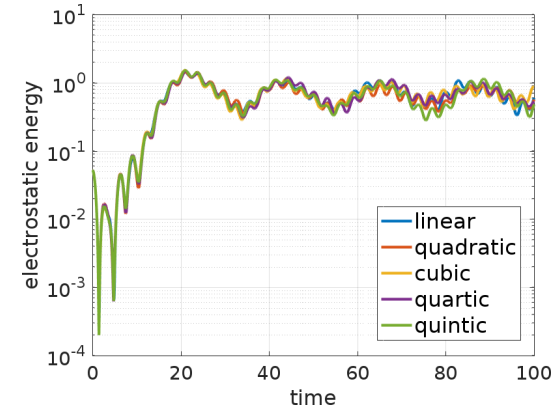
$$\begin{aligned}
\mathbb{E}\left[\hat{\tilde{\rho}}(k)\right] &= \tilde{\rho}(k) = \frac{1}{L}\int_0^L (1+\epsilon\cos(kx))\mathrm{e}^{-\mathrm{i}kx}\,\mathrm{d}x = \frac{\epsilon}{2}, \\
\mathbb{V}\left[\hat{\tilde{\rho}}(k)\right] &= \frac{1}{N_p}\frac{1}{L^2}\int_0^L \left[L\mathrm{e}^{-\mathrm{i}kx} - \frac{\epsilon}{2}\right]\left[L\mathrm{e}^{-\mathrm{i}kx} - \frac{\epsilon}{2}\right]^\dagger \frac{1}{L}(1+\epsilon\cos(kx))\,\mathrm{d}x \\
&= \frac{1}{N_p}\left[\frac{1}{L^2}\int_0^L L^2\frac{1}{L}(1+\epsilon\cos(kx))\,\mathrm{d}x - \frac{\epsilon^2}{4}\right] = \frac{1}{N_p}\left(1-\frac{\epsilon^2}{4}\right).
\end{aligned} \tag{3.66}$$

It seems that especially the grid-less PIF is suited for few very high mode numbers, since the corresponding variance is bounded and the estimator is unbiased.
Finite element PIC codes based on B-splines can only provide a biased estimate of the Fourier modes because of the discretization error depending on the grid size and spline degree. This error causes high frequencies to appear in a low frequency interval, which is called aliasing. Also Fourier methods based on the FFT also suffer from aliasing, but there are filtering techniques such as the 3/2-rule [162][p.30] to remove this effect. Aliasing is independent of the particle number and depends only on the choice of of the basis functions. We utilize Shannons sampling theorem [163] and analyze the high frequency behavior of a m-th degree B-spline S_m by using the Fourier transform obtained in

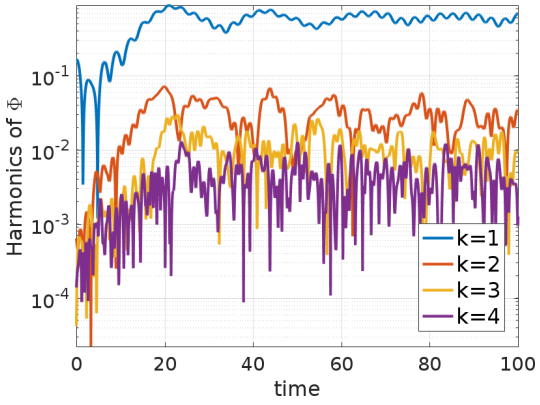
$$\tilde{S}_m(\omega) = \operatorname{sinc}\left(\frac{\omega}{2}\right)^{m+1} = \left(\frac{2\sin\left(\frac{\omega}{2}\right)}{\omega}\right)^{m+1} = \mathcal{O}\left(\frac{1}{\omega^{m+1}}\right).$$

The support of S_m in Fourier space is unbounded such that all frequencies are included which leads to aliasing, see fig. 3.6. The decay rate of $\tilde{S}_m$ depends on the B-spline degree, such that with higher order B-splines the aliasing is suppressed. As aliasing can cause instabilities [55] its extend should be quantified. For diagnostic purposes this has already been done for gyrokinetic simulations [164], such that we restrict ourselves to two dimensional Vlasov–Poisson simulations and extend the analysis to B-splines of varying degree. Estimating the Fourier modes directly — as in the unbiased PIF — yields no aliasing of other frequencies.

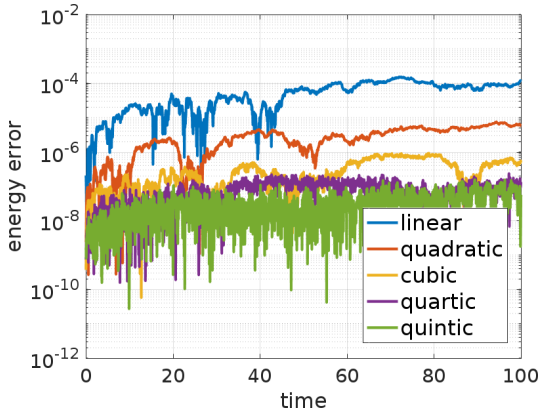
Since PIF is unbiased and the same Fourier modes can be calculated via PIC using eqn. (3.57) the bias in a PIC simulation can be determined by the difference to the PIF estimate for the same simulations. This means that the particles are advanced by the PIC scheme and only for diagnostic reasons the Fourier modes obtained in PIC by eqn. (3.57) are compared to the values obtained with the direct Fourier transform on exactly the same markers, which we call PIF. By increasing the B-spline degree, aliasing is suppressed and the harmonics of the potential estimated by PIC converge to the PIF estimate, see fig. 3.7.



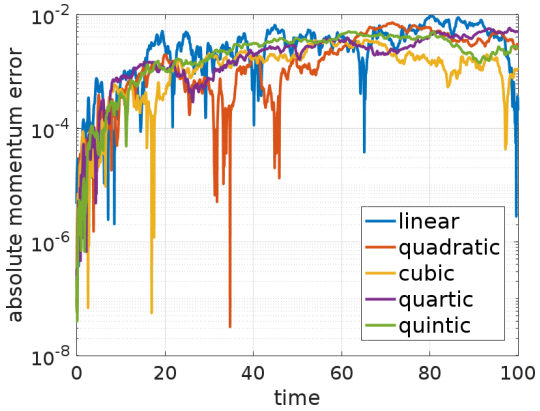
(a) Electrostatic energy



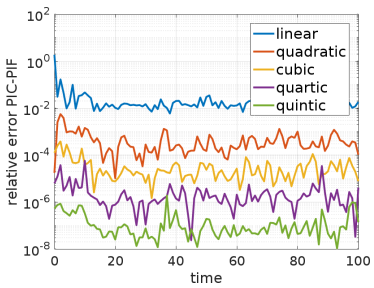
(b) Spatial harmonics $\tilde{\Phi}(k)$



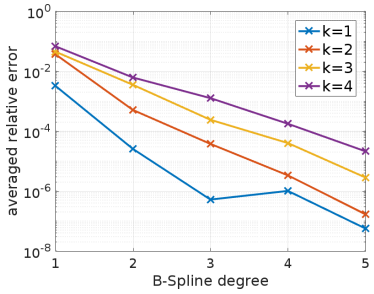
(c) Relative energy error



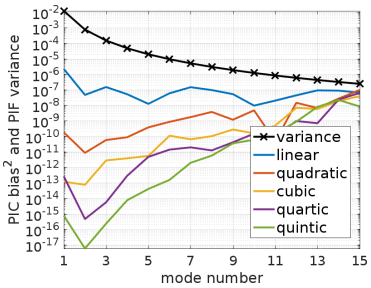
(d) Absolute momentum error



(e) Bias on harmonics $\tilde{\Phi}(k)$



(f) Mean bias on harmonics $\tilde{\Phi}(k)$



(g) Square bias and variance for $\tilde{\Phi}(k)$ at $t = 200$

Figure 3.7.: PIC simulations of the Bump-on-tail instability ($N_{\text{fem}} = 32$, $N_p = 10^4$, $\epsilon = 0.03$, $\Delta t = 0.05$, rk3s, RQMC) with varying B-spline degree. Higher order splines improve the energy conservation, but not the total momentum error. The Bump-on-tail instability is driven by the $k = 1$ mode in the linear phase and excites higher harmonics in the nonlinear regime. The bias on the PIC estimator of $\tilde{\Phi}(k)$ is calculated by using the unbiased PIF estimator for $\tilde{\Phi}(k)$ on the same particles. The variance on $\tilde{\Phi}(k)$ is estimated by PIF. Although the variance is much higher than the square bias, it is still possible to observe the difference because the same particles are used and the PIC and PIF estimate are, therefore, highly correlated. This diagnostic shows that the bias stays constant over time and decreases with increasing spline degree as expected but resides several orders of magnitude below the variance.

3.1.6. Variational Multilevel PIF

In the original Lagrangian (3.34) the particle shape function S can differ for each particle. Upon this fact a multilevel Monte Carlo (MLMC) scheme can be constructed [16]. The general idea is that coarse spatial grid exhibits less noise than a finer grid. Therefore, multiple ensembles of particles can live on coarse and fine grids, such that the coarse result is used as a control variate for the finer grid. Thus MLMC, can be extended into the control variate formalism, see [145]. The exact mean on the coarse grid is not known, but can be estimated by using much more particles or the same number of particles as the overall variance is smaller due to the coarser structure. This is realized by adapting the smoothing window with h of the particle shape function S^h. A similar idea is discussed in [18] using sparse grids but no variational framework. For M levels with smoothing widths $h_0 < \cdots < h_{M-1}$ and independent samples $(x_n^l, v_n^l, w_n^l), \quad n = 1, \ldots, N_l$ for each level $l = 0, \ldots, M-1$ the corresponding multilevel particle Lagrangian reads

$$\begin{aligned}\mathcal{L} = \frac{1}{2}\int (\partial_{\tilde{x}}\Phi(\tilde{x}))^2 \ \mathrm{d}x + \sum_{n=0}^{N_1} w_n^0 \left[x_n^0 \dot{v}_n^0 - \frac{1}{2}(v_n^0)^2 - \int S^{h_0}(\tilde{x} - x_n^0)\Phi(x) \ \mathrm{d}\tilde{x} \right] \\ + \sum_{l=1}^{M-1} \frac{1}{N_l} \sum_{n=1}^{N_l} w_n^l \left[x_n^l \dot{v}_n^l - \frac{1}{2}(v_n^l)^2 - \int \underbrace{\left[S^{h_l}(\tilde{x} - x_n^l) - S^{h_{l-1}}(\tilde{x} - x_n^l) \right]}_{\text{difference to previous level}} \Phi(x) \ \mathrm{d}\tilde{x} \right]. \end{aligned} \tag{3.67}$$

If identical samples are used for each level the Lagrangian (3.67) will collapse to the original one (3.34). For different number of samples the coarser particles act as a control variate for the finer grid, where the mean field interaction is not exactly known but sampled by the particles. Discretizing the fields with PIF and choosing the shape function as m^{th} order B-splines yields the multi-level equations of motions and the discrete Poisson equation in eqn. (3.68).

$$\begin{aligned} \dot{x}_n^l &= v_n^l, \\ \dot{v}_n^0 &= \sum_k \mathrm{i}k\Phi_k \left[\operatorname{sinc}\left(\frac{kh_0}{2}\right) \right]^{m+1} \mathrm{e}^{\mathrm{i}kx_n^0}, \\ \dot{v}_n^l &= \sum_k \mathrm{i}k\Phi_k \left\{ \left[\operatorname{sinc}\left(\frac{kh_l}{2}\right) \right]^{m+1} - \left[\operatorname{sinc}\left(\frac{kh_{l-1}}{2}\right) \right]^{m+1} \right\} \mathrm{e}^{\mathrm{i}kx_n^l}, \\ k^2\Phi_k &= \left[\operatorname{sinc}\left(\frac{kh_0}{2}\right) \right]^{m+1} \frac{1}{L}\frac{1}{N_0} \sum_{n=1}^{N_0} w_n \mathrm{e}^{-\mathrm{i}kx_n} \\ &\quad + \frac{1}{L} \sum_{l=1}^{M-1} \left\{ \left[\operatorname{sinc}\left(\frac{kh_l}{2}\right) \right]^{m+1} - \left[\operatorname{sinc}\left(\frac{kh_{l-1}}{2}\right) \right]^{m+1} \right\} \frac{1}{N_l} \sum_{n=1}^{N_l} w_n^l \mathrm{e}^{-\mathrm{i}kx_n^l} \end{aligned} \tag{3.68}$$

It remains to make a choice concerning the particle width h_l and the number of particles for each level. Given a particle width h_0 the other levels can defined by refinement according to $h_l = \frac{1}{2^l}h_0, \ldots l = 0, \ldots, M-1$. From the smoothed Vlasov–Poisson system (3.36) and eqn. (3.68) it becomes clear that the additional convolution is applied two times, such that the integrated variance of the electric field under spatially uniform sampling reads

$$\mathrm{IVAR}\left[\hat{E}_l\right] = \frac{1}{N_l} \sum_{\substack{k=-N_f \\ k\neq 0}}^{N_f} \frac{L}{k^2} \operatorname{sinc}\left(\frac{kh_l}{2}\right)^{4(m+1)}. \tag{3.69}$$

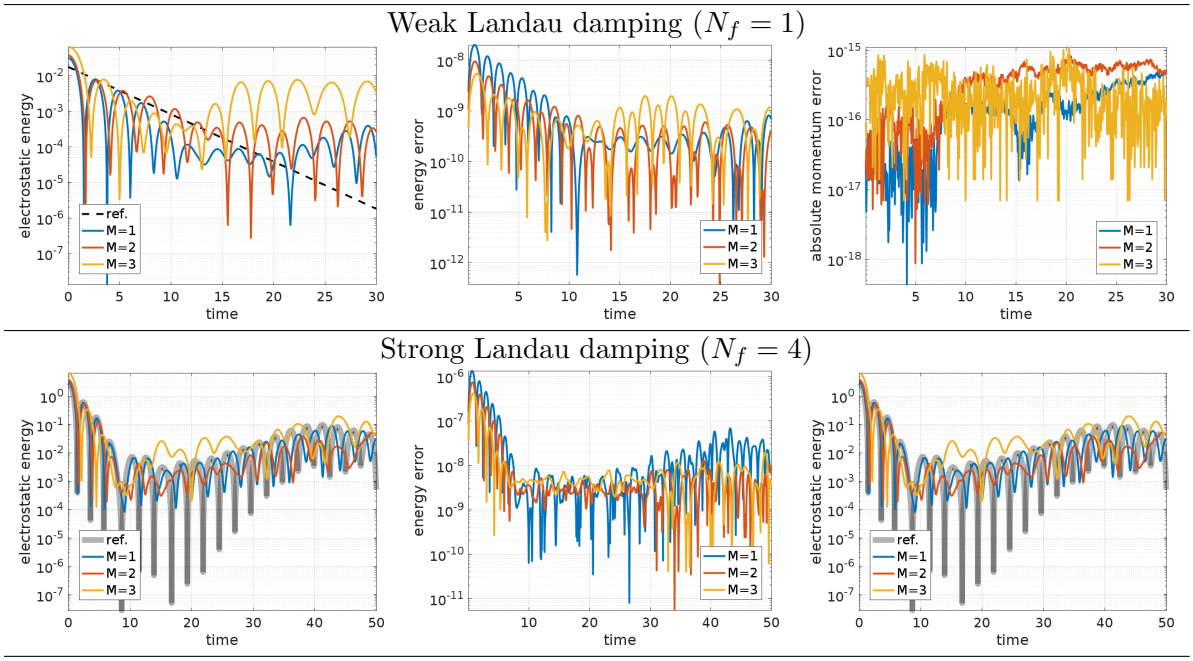


Figure 3.8.: Multilevel Monte-Carlo PIF simulations of weak Landau damping using different quadratic B-spline particle shape functions S_2 with varying number of levels M. With increasing number of levels the solution deviates clearly from the reference. ($h_{M-1} = \frac{L}{4(2+1)}$, $N_p = 10^6$, $\Delta t = 0.05$, rk3s)

For a given total number of markers $N_p = \sum_{l=0}^{M-1} N_l$, the number of markers N_l for each level is then chosen such that the integrated variance is balanced according to

$$N_l \sim \sum_{k=1}^{N_f} \frac{2L}{k^2} \operatorname{sinc}\left(\frac{k h_l}{2}\right)^{4(m+1)} . \tag{3.70}$$

PIC has problems resolving small amplitudes due to the noise, where a control variate that subtracts the background $f(v)$ provided a remedy. This form of noise is the dominant problem in Monte Carlo particle methods, which unfortunately cannot be solved by purely particle based MLMC approach. With increasing number of levels and therefore, decreasing number of N_l particles per level the background noise on each level is larger such that small amplitude effects such as linear and nonlinear Landau damping are recovered worse, see fig. 3.8. Similar results are reported in [18], where the memory consumption was reduced by sparse grids but not the particle noise thus rendering the MLMC mechanism inefficient. We conclude that MLMC might be better suited for multiple levels in the time discretization, which was already successfully applied for general Vlasov–McKean processes in [17].

3.2. Particle in spectral space

We can generalize the Particle-In-Fourier method in order to solve the Poisson equation in a non-periodic domain including Dirichlet and Neumann boundary conditions. The Fourier modes form a global orthogonal series in the periodic domain, which rapidly approximates smooth functions at the expense that every particle contributes to every Fourier mode. Spectral methods for bounded domains are based on other orthogonal series, which are mostly global polynomials [54, 162]. In that context every particle contributes also to every polynomial, but polynomials are much cheaper to evaluate than the trigonometric functions. Instead

of going to deep into the theory of these polynomials we want to point out the Chebyshev identity (3.71), which tells us that the Fourier modes — here $\cos(n\theta)$ and $\sin(n\theta)$ — can be obtained by a two term recurrence relation.

$$\begin{aligned}\cos(n\theta) &= 2\cos(\theta)\cos\left((n-1)\theta\right) - \cos\left((n-2)\theta\right),\\ \sin(n\theta) &= 2\cos(\theta)\sin\left((n-1)\theta\right) - \sin\left((n-2)\theta\right).\end{aligned} \tag{3.71}$$

Those recurrence relations play an essential role in the theory of the spectral methods, such that all the other orthogonal series are also defined by such a relation. The most popular global spectral methods use the Chebyshev polynomials, which are defined as

$$T_n(x) = \cos\left(n\cos^{-1}(x)\right) \quad x \in x \in [-1,1]. \tag{3.72}$$

With the Chebyshev identity (3.71) this definition yields a three term recurrence relation eqn. (3.73), which is the handier definition for a series of polynomials. It provides an efficient, and most important, numerically stable scheme to evaluate all Chebyshev polynomials at a certain position $x \in [-1,1]$.

$$\begin{aligned}T_0(x) &= 1\\ T_1(x) &= x\\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x)\end{aligned} \tag{3.73}$$

More formulas can be found in the appendix C.1.1, such that we can turn to another set of orthogonal polynomials.
The straightforward orthogonal series density estimation (OSDE) uses the orthogonal Legendre polynomials P_n, see [165] for an overview and appendix C.1.3 for useful equations. The use of Legendre polynomials is not very widespread because, contrary to the Chebyshev polynomials, there is no similar fast transform from values on a grid to Legendre coefficients. Yet this poses no obstacle for Lagrangian particles, since there is no grid present. Thus, the Legendre polynomials are perfectly suited for particle methods. They are defined for $x \in [-1,1]$ by the three term recurrence eqn. (3.74).

$$\begin{aligned}P_0(x) &= 1\\ P_1(x) &= x\\ P_{n+1}(x) &= \frac{2n+1}{n+1}xP_n(x) - \frac{n}{n+1}P_{n-1}(x)\end{aligned} \tag{3.74}$$

The Legendre polynomials are very well suited for density estimation because of their strict orthogonality with respect to the Lebesgue measure, see eqn. (3.75).

$$\int_{-1}^{1} P_n(x)P_m(x)\,\mathrm{d}x = \frac{2}{2n+1}\delta_{n,m} \tag{3.75}$$

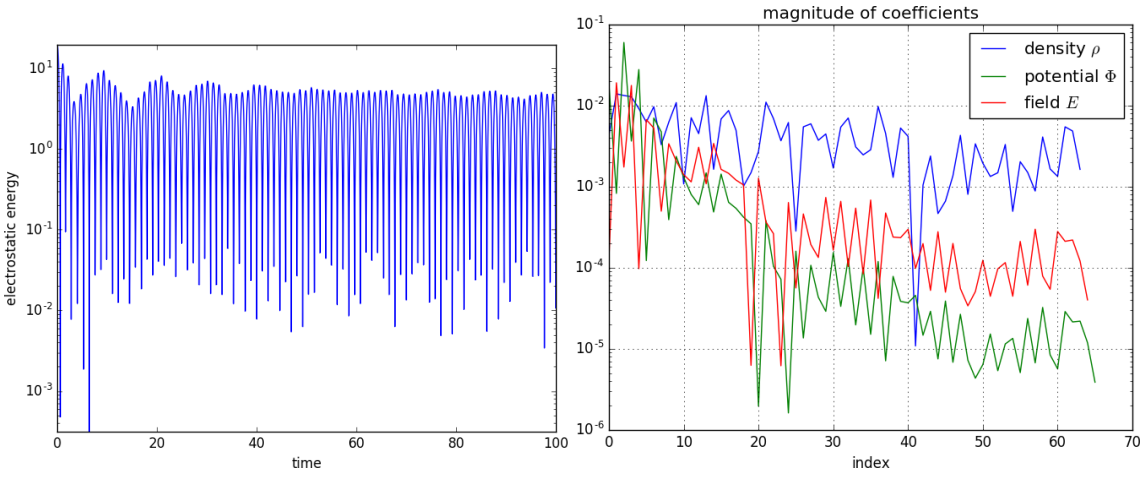
Given Lagrangian particles, the L^2 projection is the most attractive operation, since, in contrast to the Chebyshev polynomials, no additional weighting function is present. But on the other hand, Chebyshev based methods are much more widespread and efficient. Yet any Legendre series can be transformed into a Chebyshev series and vice versa. Algorithms performing the transform in both directions are available in $\mathcal{O}(N)$, see [166] and $\mathcal{O}\left(\frac{N\log(2N)}{\log\log(N)}\right)$ from [167]. The latter one is actually faster and available in *FastTransforms.jl*.

3.2.1. Particle in Legendre and Chebyshev

There is an abundance of systems that can be solved using Particle-In-Cell yet many times finite element solvers are rewritten from scratch although well developed libraries are available. There is development in using particles with deal.II [168], and fenics [169]. For spectral methods in MATLAB *chebfun* [170] is the obvious choice, but since MATLAB does not scale it is not a long term option. *Julia* on the other hand is much faster and better suited for large scale particle methods and there is a freshly emerging spectral library *Approxfun.jl* [171]. Therefore, we implement a particle-spectral-grid coupling in this environment. For over 20 years efficient spectral methods have been derived for simple geometries (cylinder, sphere, see [172, 173, 174, 175]) and we could implement them right away, but then a specific plasma physics problem tailored to the corresponding geometry has to be solved. The Poisson equation with homogeneous Dirichlet boundary conditions can be solved trivially with Legendre polynomials, such that for a clamped mode Linear landau damping us used to verify the feasibility of the scheme in fig. 3.10. In a more general approach *Approxfun.jl*, which is embeeded in julia *approximation*, provides us with a set of tools for any nonlinear PDE. Especially the banded matrix assembly for all types of boundary conditions and the preconditioning by ultraspherical polynomials are taken care of. This has such a generality that we demonstrate something rather odd, namely a periodic Vlasov–Poisson solver that uses Legendre and Chebyshev polynomials as basis functions. The particle mesh coupling takes place at the level of Legendre polynomials, where the obtained coefficients are transformed into the Chebyshev basis using *FastTransforms.jl*. On this level any boundary conditions or equations can be solved efficiently. Since the resulting fields are given in Chebyshev polynomials, the back transform onto the Legendre basis is skipped. The errors made by this basis transform are on the level of machine precision, hence we can safely ignore them. The algorithm is long-term stable and energy is conserved, see fig. 3.9. The more interesting question is how many degrees of freedom are needed given a certain number of particles or where should we truncate the expansion. Given the heavily perturbed density at the end of the simulation, see fig. 3.9b, there are obviously many modes present. But the field coefficients are decreasing, see fig. 3.9b, which means some of them are merely noise and can be neglected. In order to avoid calculation of covariances we turn to the Legendre basis and estimate the variance of the Legendre coefficients. Taking advantage of the Legendre orthogonality this costs as much as another charge assignment. Although the coefficients oscillate, the tail $n > 30$ is at the order of the standard deviation, such that those coefficients are dominated by noise and do not contribute to the solution, see fig. 3.9e. Possible improvements can be made by averaging the coefficient of variance over some time in order to obtain a smoother picture. Given the capabilities of *Approxfun.jl* for tensor structured domains, this particle mesh coupling can be directly used in higher dimensions including other nonlinear PDEs that can also incorporate curvature.

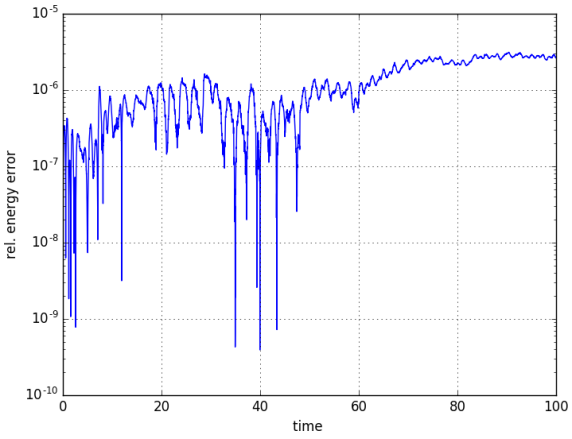
3.2.2. Particle-In-Fourier Hankel

The natural analytic way of solving the Poisson equation in cylindrical domain is by the Fourier Hankel transform [176]. The idea was already applied in a Vlasov–Maxwell PIC code that still has an intermediate grid [147, 148]. Here we focus on an entirely grid-less variant. The expansion in Bessel functions has only algebraic convergence, thus other possible methods based on polynomials are a better choice from a numerical perspective see [54][pp. 385]. Here polar coordinates (r, θ) are used, where the basis function in the periodic direction θ are Fourier modes and in the radial direction Bessel functions.
For a radially symmetric density $\rho(r)$ the continuous m^{th} order Hankel transform is defined

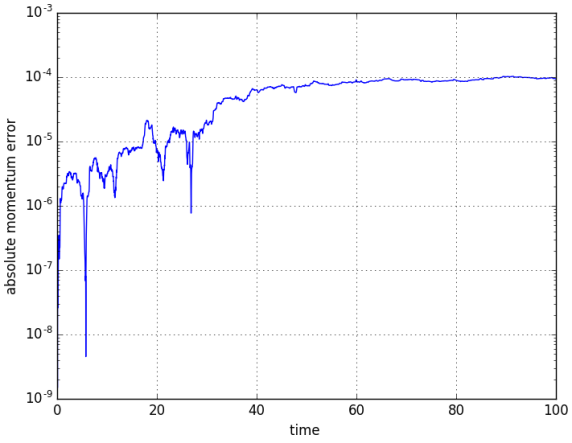


(a) electrostatic energy

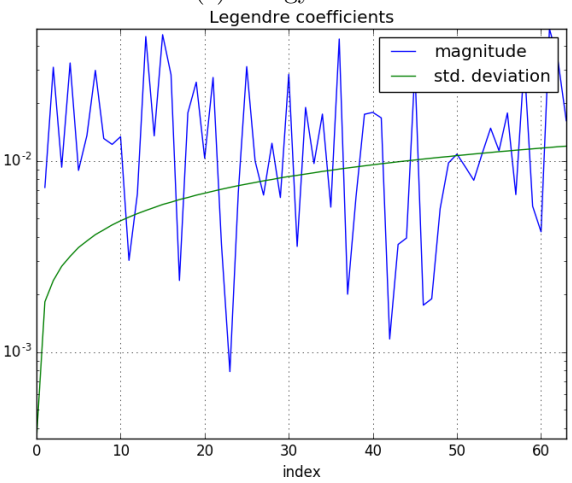
(b) spectral coefficients



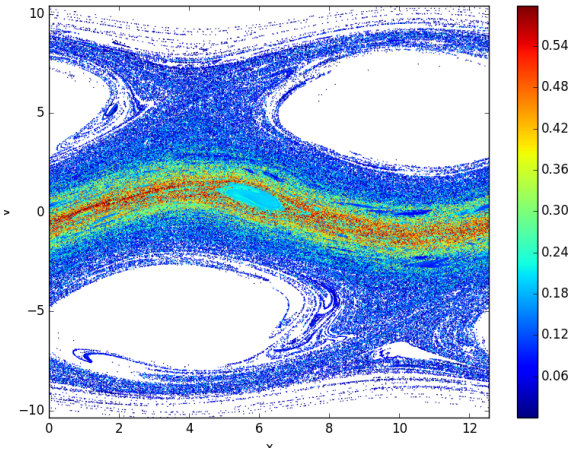
(c) energy error



(d) momentum error

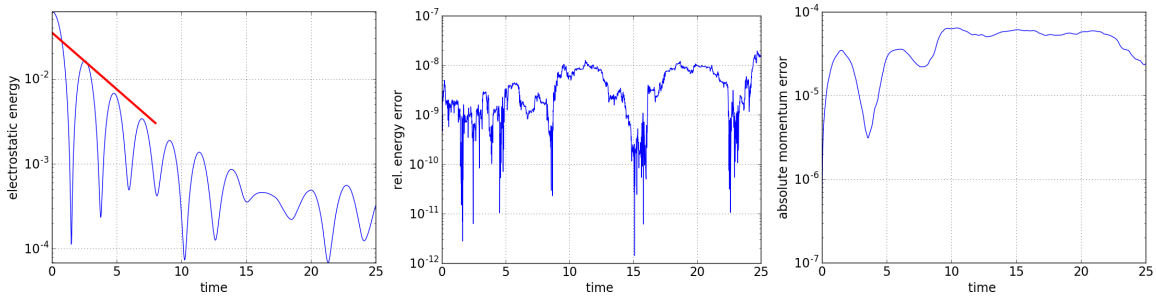


(e) Legendre coefficient noise at $t = 100$



(f) particle density at $t = 100$

Figure 3.9.: Nonlinear Landau damping with Legendre polynomials and periodic boundary via the Chebyshev representation provided by *Approxfun.jl*. ($N_p = 10^6$, $\Delta t = 0.01$, $N_x = 64$, $k = 0.5$)



(a) electrostatic energy (b) energy error (c) absolute momentum error

Figure 3.10.: Weak Landau damping with Legendre polynomials, periodic boundary conditions for the particles and homogeneous Dirichlet boundary conditions for the electrostatic potential Φ. By using the initial condition $f(x,v,0) = \mathrm{e}^{-\frac{v^2}{2}} \frac{1}{\sqrt{2\pi}} (1 + 0.05 \sin(kx))$ the excited mode is clamped to the homogeneous Dirichlet boundary conditions, such that the results from the periodic linear analysis can be used for code validation (a). In this way complicated boundary conditions for the particles are circumvented and energy conservation can be observed (b), but the momentum conservation is lost (c). ($N_p = 10^6,\ \Delta t = 0.01,\ N_x = 20,\ k = 0.5$)

as

$$\tilde{\rho}(k_r) = \int_0^\infty \rho(r) J_m(k_r r) \mathrm{d}r, k_r \in \mathbb{R}. \tag{3.76}$$

If we Fourier transform the usual way in θ, the function $\rho(r,\theta)$ can be decomposed into a series of Fourier modes and Bessel functions. The mth order Bessel function J_l is coupled to the mth Fourier mode in θ by $m = k_\theta \in \mathbb{Z}$.

$$f(r,\theta) = \sum_l \sum_m \tilde{\rho}(k_r, k_\theta) \mathrm{e}^{\mathrm{i}2\pi m\theta} J_m(r) \tag{3.77}$$

We recall some additional definitions and properties of the Bessel function of first kind, see also [177].

$$J_{-m}(r) := (-1)^m J_m(r) \tag{3.78}$$

The derivative of a Bessel function can again be expressed by Bessel functions of different order.

$$\partial_r J_m(r) = \begin{cases} -J_1(r) & \text{if } m = 0 \\ \frac{1}{2}\left[J_{m-1}(r) - J_m(r)\right] & \text{else} \end{cases} \tag{3.79}$$

Dirichlet Boundary condition

We want to solve the Poisson equation with Dirichlet boundary conditions. Here $a_{m,l}$ denotes the l^{th} zero of the m^{th} order Bessel function of first kind, see eqn. (3.80).

$$J_m(a_{m,l}) = 0, \quad l \in \mathbb{N},\ m = 0, 1, \ldots \tag{3.80}$$

We normalize by the radius $r_{\max}$ in order to have orthogonal basis functions.

$$J_{m,l}(r) := J_m\left(r \frac{a_{m,l}}{r_{\max}}\right) \tag{3.81}$$

$$\partial_r J_{m,l}(r) = \partial_r J_m\left(r \frac{a_{m,l}}{r_{\max}}\right) \frac{a_{m,l}}{r_{\max}} = \frac{a_{m,l}}{r_{\max}} \frac{1}{2} \left[J_{m-1}\left(r \frac{a_{m,l}}{r_{\max}}\right) - J_{m+1}\left(r \frac{a_{m,l}}{r_{\max}}\right)\right] \tag{3.82}$$

For the normalization we define for every pair of Bessel function and Fourier mode the normalizing constant

$$\lambda_{m,l}^2 := \int_0^{r_{\max}} J_{m,l}(r)^2 r \, \mathrm{d}r = \frac{r_{\max}^2}{2}[J_m'(a_{m,l})]^2 = \frac{r_{\max}^2}{2}[J_{m+1}(a_{m,l})]^2. \tag{3.83}$$

The Fourier–Bessel coefficients for the density ρ are obtained by

$$\tilde{\rho}_{m,l} := \frac{1}{2\pi}\int_0^{2\pi} \frac{1}{\lambda_{m,l}^2}\int_0^{r_{\max}} \mathrm{e}^{-\mathrm{i}m\theta} J_m\left(r\ \frac{a_{m,l}}{r_{\max}}\right)\rho(r,\theta) r \, \mathrm{d}r \, \mathrm{d}\theta, \tag{3.84}$$

yielding the expansion

$$\rho(r,\theta) = \sum_{m,l} \tilde{\rho}_{m,l}\ \mathrm{e}^{\mathrm{i}m\theta} J_m\left(r\ \frac{a_{m,l}}{r_{\max}}\right). \tag{3.85}$$

The Fourier-Bessel coefficients for the Poisson equation in polar coordinates

$$\Delta\Phi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\Phi}{\partial\theta}\right) + \frac{1}{r^2}\frac{\partial^2\Phi}{\partial\theta^2} = \rho, \tag{3.86}$$

are obtained as

$$\tilde{\Phi}_{m,l} = \left(\frac{r_{\max}}{a_{m,l}}\right)^2 \tilde{\rho}_{m,l} \tag{3.87}$$

yielding the electric potential

$$\Phi(r,\theta) = \sum_{m,l} \tilde{\Phi}_{m,l}\ \mathrm{e}^{\mathrm{i}m\theta} J_m\left(r\ \frac{a_{m,l}}{r_{\max}}\right). \tag{3.88}$$

In Cartesian coordinates the gradients are given as

$$\partial_x\Phi = \cos(\theta)\partial_r\Phi - \sin(\theta)\partial_\theta\Phi,\ \ \partial_y\Phi = \sin(\theta)\partial_r\Phi + \cos(\theta)\partial_\theta\Phi. \tag{3.89}$$

This suffices to implement a Vlasov–Poisson solver using Lagrangian particles, where the particles can either live in the logical coordinates (r,θ) or in Cartesian (x,y). One downside is the costly numerical evaluation of Bessel functions; in MATLAB it is around ten times slower than the complex exponential. This is the analog to PIF — the local costs increase but the field solve remains a scalar multiplication and is therefore, highly scalable.

But we can learn something different from this orthogonal series expansion. In most particle simulations there is some form of Fourier filtering applied in order to reduce the integrated variance of the field. In periodic directions Fourier modes are taken, yet for a finite element Fourier discretization of the polar plane it is unclear what to filter best, since one would like to filter something physically reasonable. Here the truncation of the Fourier-Bessel is a possible answer.

The Bessel functions are not the only orthogonal functions on the polar plane. For example, Zernike polynomials [178] constructed by Gram Schmidt orthogonalization of the monomial basis $[1, r, r^2, r^3, \dots]$ with respect to the scalar product $\langle f, g\rangle = \int_{r_{\min}}^{r_{\max}} f(r)g(r)r \, \mathrm{d}r$ are one choice. By construction, they form an orthogonal basis on the polar plane with Jacobian $J(r,\theta) = r$. They suffer from oscillatory behavior for higher degree limiting their application in numerics [179]. But it is even possible to generalize the Zernike polynomials on elliptical surfaces, see [180], such that they are possible candidates for describing a toroidal magnetic equilibrium.

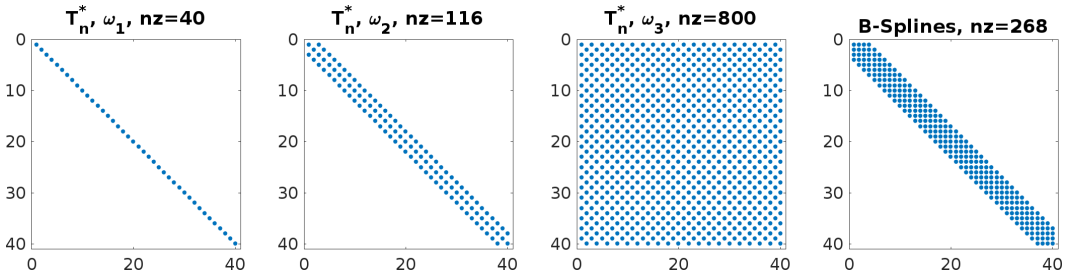


Figure 3.11.: Sparsity patterns of the mass matrix for 40 degrees for Chebyshev polynomials with varying weights and cubic B-splines.

3.3. Orthogonal series density estimation (OSDE)

Our favorite tool for obtaining density estimates is orthogonal series density estimation, because partial differential equations can be solved easily and noise is naturally filtered by truncation. For a stochastic overview and truncation rules we recommend [181]. We begin with an example for Legendre and Chebyshev methods, continue with the application onto control variates for nonlinear problems and conclude with multidimensional PIF for Vlasov–Poisson in order to demonstrate the drawbacks of increasing dimensionality.

3.3.1. Example

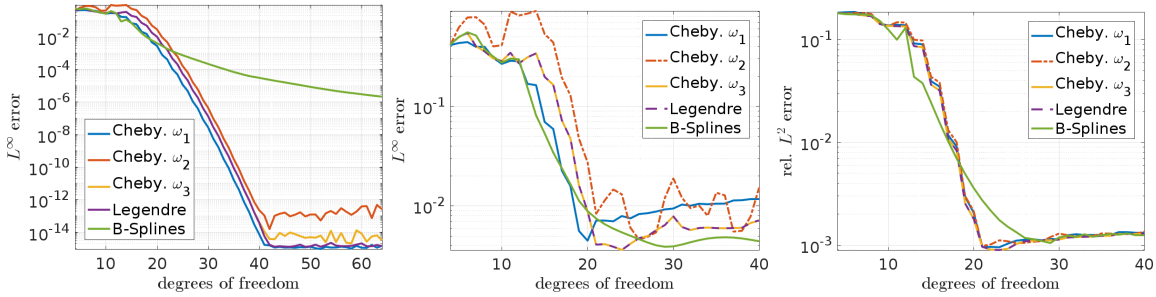
We demonstrate density estimation on the bounded domain $(0, 1)$ using orthogonal polynomials. For this, a scaled Bessel function $f(x) = \mathcal{J}_0(30 \cdot x)$ is reconstructed using the uniformly distributed random deviate $X \sim \mathcal{U}(0, 1)$ implying a constant sampling density $g(x) = 1$ and different orthogonal polynomials, see fig. 3.12. The standard PIC method is represented by Finite Elements based on cubic B-splines. For the shifted Chebychev polynomials $\{T_n^*\}$, the Galerkin scalar product is weighted by functions ω_i given in eqn. (3.90).

$$\omega_1(x) = \frac{1}{2\sqrt{x(1-x)}}, \quad \omega_2(x) = 2\sqrt{x(1-x)}, \quad \omega_3(x) = 1 \tag{3.90}$$

This heavily impacts the sparsity of the mass matrix and also the variance of the Galerkin right hand side, which is estimated from eqn. (3.91) by $N_p = 10^6$ samples.

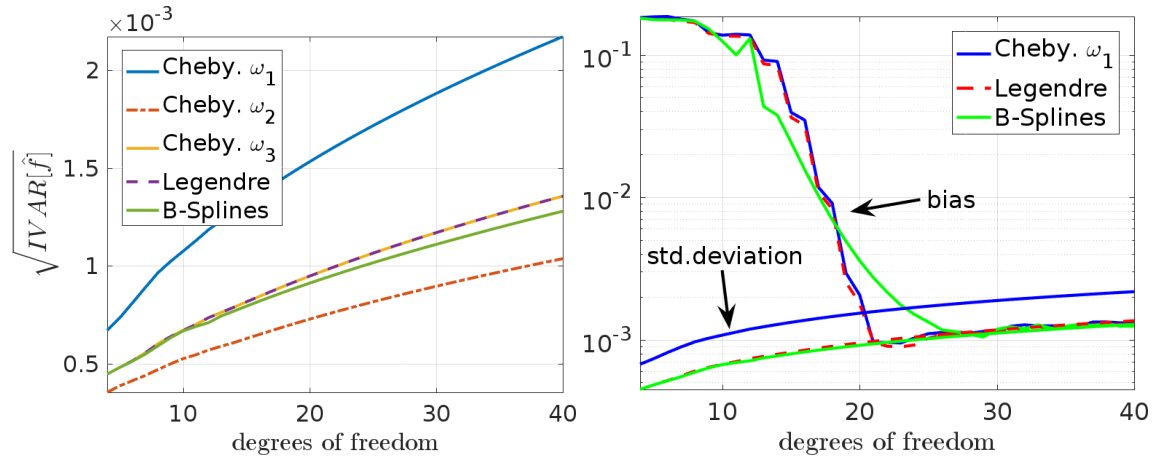
$$\int_0^1 f(x)\psi_j(x)\omega(x)\ \mathrm{d}x = \mathbb{E}[f(X)\psi_j(X)\omega(X)] \tag{3.91}$$

As we can see in fig. 3.11 the sparsity pattern for the Chebyshev polynomials depends on the weight ω beating the sparse-by-construction B-splines in efficiency for ω_1 and ω_2. The complete orthogonality is achieved by ω_1, which unfortunately leads to the worst integrated variance, see fig. 3.13a. This is due to the singularity in $\omega_1(\{0, 1\})$. Thus, for OSDE naturally orthogonal Legendre polynomials are better suited. Damping the boundary by $\omega_2\{0, 1\} = 0$ leads to the best integrated variance. When the right hand side is obtained by numerical quadrature, the B-splines yield a good initial precision but are ultimately outperformed by the spectral methods, see fig. 3.12a. For the Bessel function the spectral methods could be even better if we take into account the parity of the steps shown in fig. 3.12a, which was already pointed out by [54]. In the Monte Carlo approximation the errors can be seen in figs. 3.12b and 3.12c. Saturation is reached for more than 21 degrees of freedom. Here we have chosen the number of particles large enough such that the difference between the spectral and the B-spline approximation is already visible. The convergence rate shall not



(a) L^2 Galerkin projection (b) Monte Carlo $N_p = 10^6$ (c) Monte Carlo $N_p = 10^6$

Figure 3.12.: Orthogonal series density estimation of a scaled Bessel function $\mathcal{J}_0(30 \cdot x)$ represented by uniformly distributed MC samples. Ultimately the spectral methods, Chebyshev and Legendre, outperform the cubic B-splines in the standard L^2 Galerkin projection (a), which is almost irrelevant here given the high sample noise.



(a) integrated variance (b) bias and variance

Figure 3.13.: The integrated variances (a) increase with the degrees of freedom where choice of the additional weighting ω for Chebyshev polynomials has a visible impact. Similar to the L^2 orthogonal Fourier modes in PIF, the L^2 orthogonal Legendre polynomials exhibit a slightly higher variance than the B-splines. In this manufactured example the bias can be calculated, such that the variance bias trade-off problem is solved by the intersection of the two curves in b).

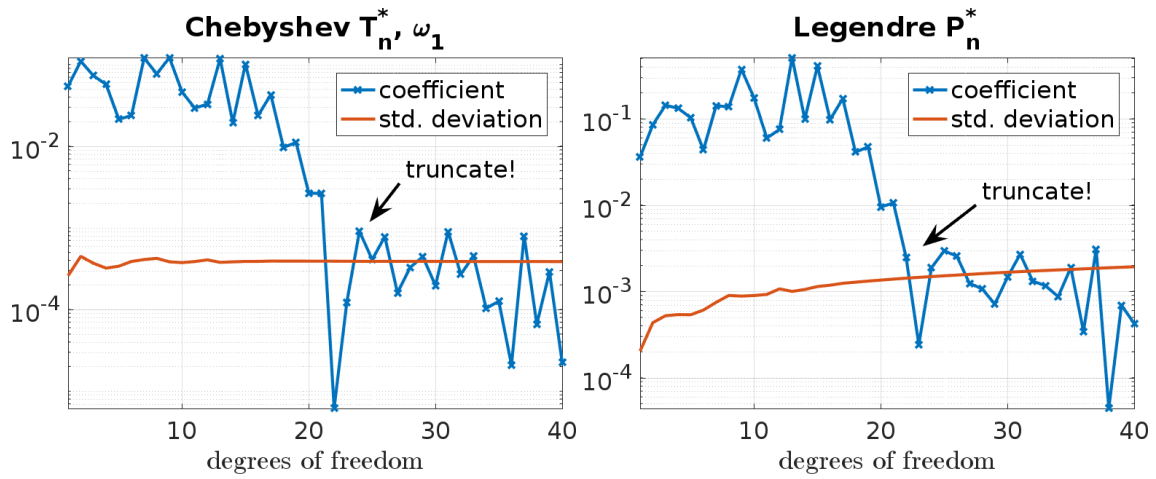
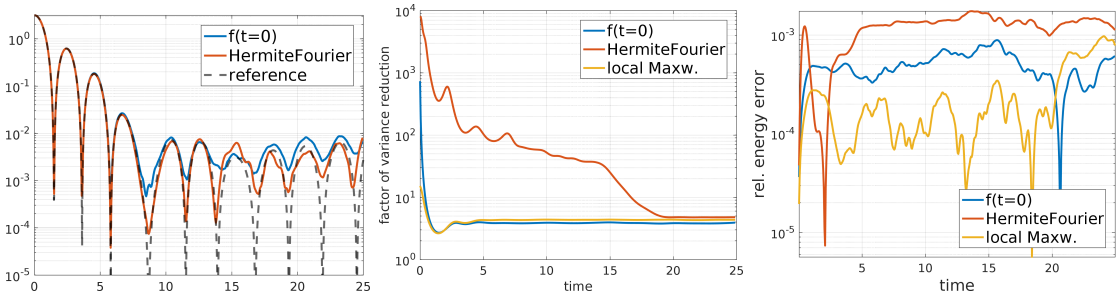


Figure 3.14.: Coefficients of the Chebyshev (left) and Legendre expansion (right) and their respective variance provide an a posteriori rule-of-thumb truncation criteria. Even without the variance the point of truncation can be guessed.

be the main objective here yet constructing a suitable filter reducing computational costs is what we are mainly interested in. Here f is known; therefore, the error can be calculated directly and it is obvious when convergence is reached and the series can be truncated. This yields a natural filter reducing the computational costs. The same can be accomplished with the finite elements and principal component analysis, except coefficients cannot be truncated a posteriori because any change in the grid size yields different basis functions. It is possible, but quite a hassle to implement, giving the orthogonal spectral methods a clear advantage. In fig. 3.13b the error and the respective integrated variance are plotted and it is obvious that the convergence is reached at their intersection. As always, having the bias and the variance available yields a good truncation criteria. However, absolute error estimation with finite elements requires complicated h or p refinement, which will also affect the variance. The coefficients of the spectral expansion decay *fast*, where the last coefficient is already a measure for the discretization error [54], here the bias. Thus when they stop decaying: truncate them! In the orthogonal cases the variance of the coefficients can be estimated very cheap and directly. This means having both variance and bias from a purely data driven estimation, the point of truncation in fig. 3.14 is obvious and compares very well to fig. 3.12c. There is much more theory on truncation rules available in [181] allowing for a self-tuning method.

3.3.2. Fourier–Hermite control variate

Hermite functions are, due to their Gaussian envelope, well suited to approximate a Maxwellian distribution. Therefore, many spectral solvers make successfully use of a Fourier–Hermite representation of the plasma distribution [3, 157, 158, 182]. If there exists a basis that can approximate the distribution f with few degrees of freedom it may be well suited as a control variate using the δf method. Due to the fine structure in nonlinear Landau damping it is very hard to find a good control variate, nevertheless low rank Fourier–Hermite OSDE of the distribution appears to work, see fig. 3.15. For nonlinear Landau damping the resolution in velocity space ensures longer effectiveness, such that the Ansatz with few spatial modes seems reasonable, see fig. 3.16. When the distribution enters a strongly nonlinear phase too many Fourier–Hermite modes are required, such that the control variate de-correlates. This



(a) electrostatic energy (b) variance reduction (c) energy error

Figure 3.15.: Fourier–Hermite series $(N_f \times N_v) = (3 \times 45)$ as a control variate for nonlinear Landau damping in comparison to the local Maxwellian and the initial condition. $(N_p = 10^4, N_f = 32,\ RQMC)$. Although the energy error (c) remains unchanged, the Hermite–Fourier control variates yields an electrostatic energy (a) that lies closer to the reference solution than a δf scheme using the initial condition as control variate. This can also be explained by the larger variance reduction (b) that unfortunately decays with increasing nonlinearity.

raises the question whether the δf scheme is worth the effort in the nonlinear phase, when a simple spectral solver using the same basis achieves the same result without noise.

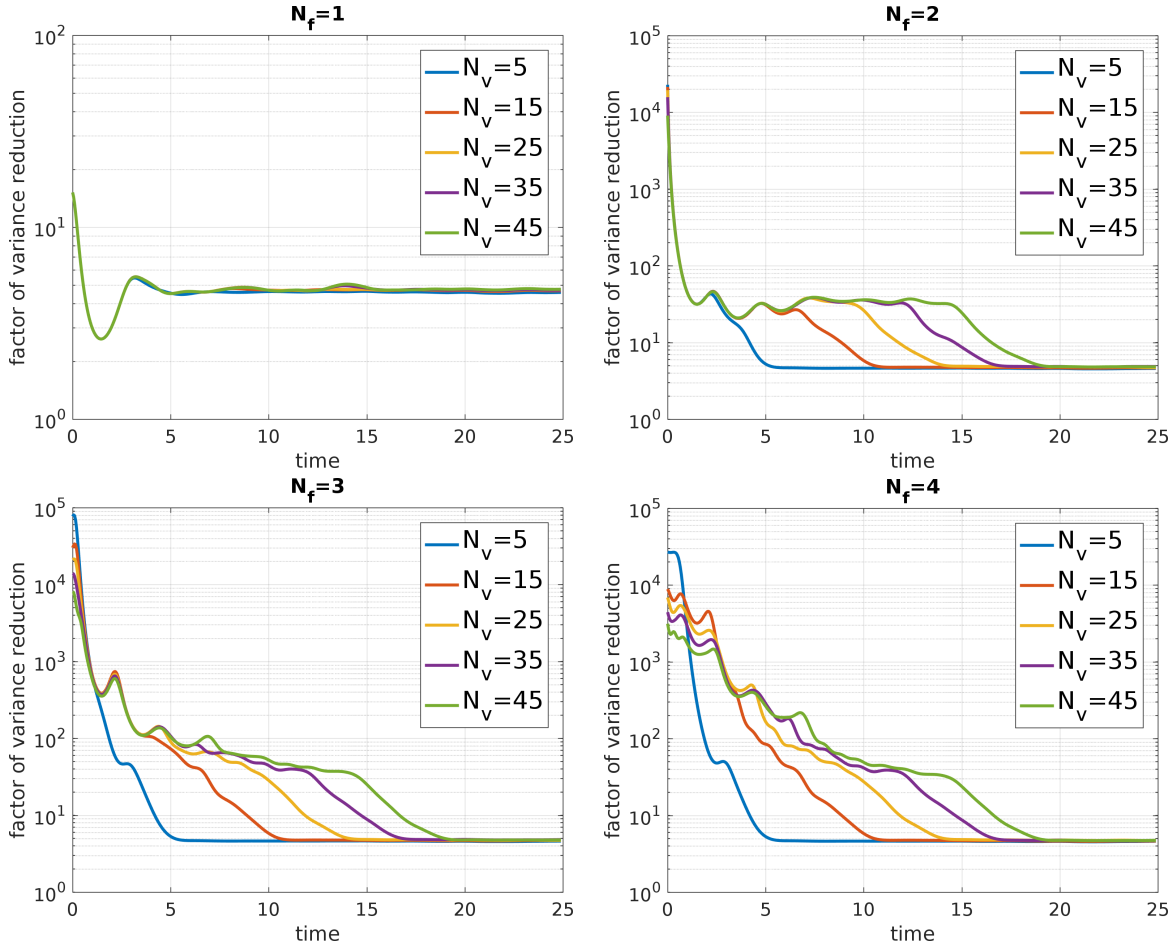


Figure 3.16.: For nonlinear Landau damping the Fourier–Hermite ($N_f \times N_v$) control variate starts gaining efficiency once more than one Fourier mode is present. The more resolution in velocity space, the longer the control variate is well correlated. Here $N_f = 1$ means, that only the zeroth mode is present such that the control variate is nothing more than an enhanced Maxwellian.

3.3.3. PIF for multidimensional Vlasov–Poisson

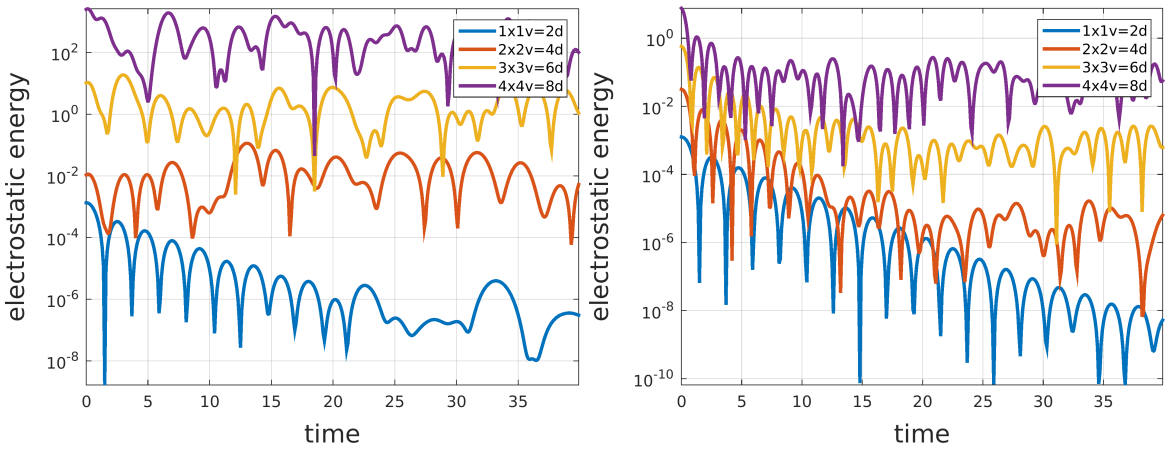
The straightforward implementation of PIF allows for solving Vlasov–Poisson in an arbitrary dimension $d \in \mathbb{N}$ with the same implementation. The spatial domain is set to be a d-dimensional periodic box $[0, L]^d$.

$$f(x, v, t = 0) := \left(1 + \epsilon \cos\left(\sum_{j=1}^{d} x_j \frac{2\pi}{L}\right)\right) \frac{1}{\left(\sqrt{2\pi}\right)^d} e^{-\frac{|v|^2}{2}} \tag{3.92}$$

The markers are sampled uniformly in x and v, with $v_{\min} = -10, v_{\max} = 10$ sampling using RQMC Sobol numbers.

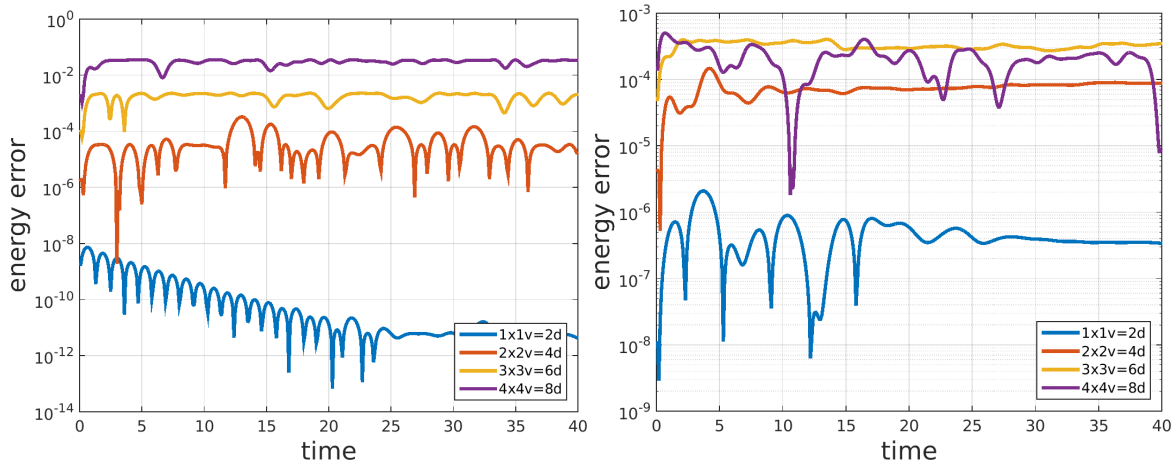
$$g(x, v, t = 0) := \frac{1}{L^d} \frac{1}{(v_{\max} - v_{\min})^d} \tag{3.93}$$

The correlation coefficient ρ is estimated for every Fourier mode respectively. The fourth order symplectic Runge Kutta scheme [32] suitable for Vlasov–Poisson is used for time integration. We begin with testing linear $\epsilon = 0.1$ and nonlinear $\epsilon = 0.5$ Landau damping in dimensions $d = 1, \ldots, 4$ and for $k = 0.5$, $N_p = 10^5, L = \frac{2\pi}{k}$ $\Delta t = 0.1$.
For linear Landau damping only the excited mode $m = 1$ is calculated. This allows for a fast field solve, which leads to a linear increase in the simulation time with respect to the dimension d. Fig. 3.18 shows that with full f energy conservation is obtained as one would expect also by a PIC method. Additionally, PIF delivers momentum conservation up to roundoff, see fig. 3.19. But since the spatial disturbance is too small, the simulation is governed by noise and not capable of finding the correct damping rate, see fig. 3.17. Although we are in the linear phase of a linear problem the problem becomes harder with higher dimension. Therefore, it is clear, that despite the Monte Carlo Ansatz we do not have convergence independent of the dimension and thus a heavy curse of dimensionality. With introduction of the control variate, we are able to alleviate the problem yet we lose the high precision in momentum conservation. But due to the absence of any self force, the error seems bounded.



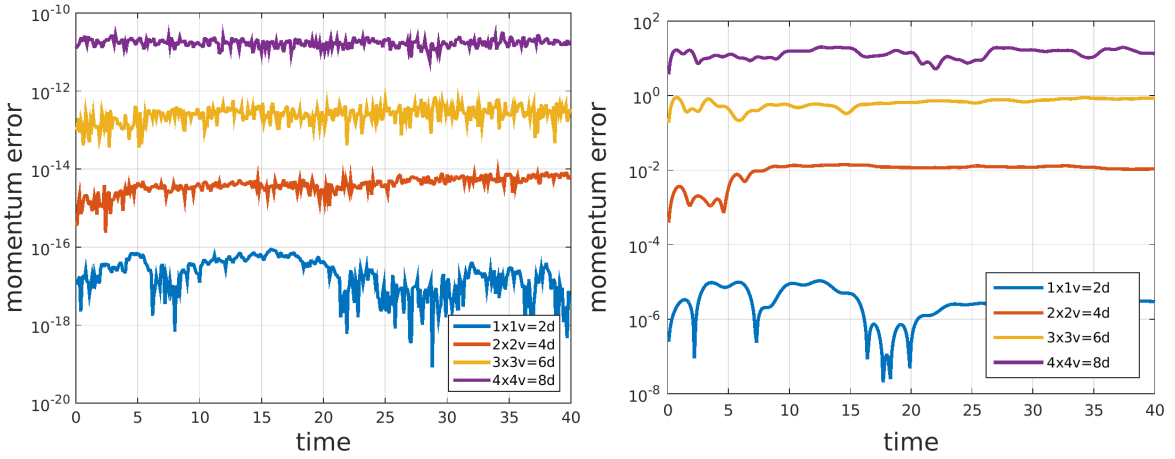
(a) full f (b) control variate δf

Figure 3.17.: Electrostatic energy for linear Landau damping with PIF.



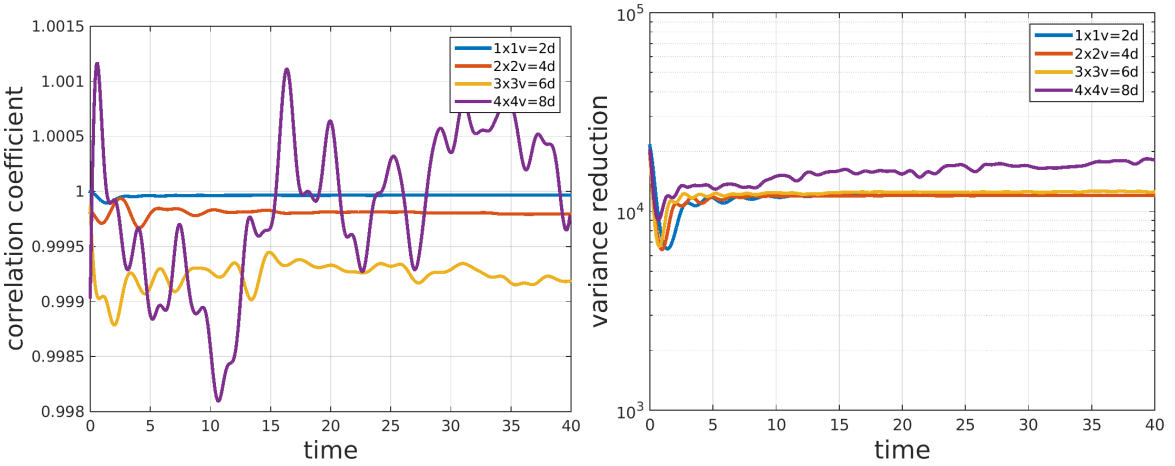
(a) full f (b) control variate δf

Figure 3.18.: Relative energy error for linear Landau damping with PIF



(a) full f (b) control variate δf

Figure 3.19.: Absolute momentum error for linear Landau damping with PIF



(a) estimated correlation coefficient $\hat{\alpha}$ (b) estimated variance reduction

Figure 3.20.: Control variate diagnostics for linear Landau damping

3.4. Electromagnetic Particle-in-Fourier

A Hamiltonian splitting for the Vlasov–Maxwell equations is introduced in [19]. Here the discretization with Lagrangian particles uses PIF instead of PIC. Contrary to PIC [19] the global basis functions simplify the discretization. The finite element exterior calculus is not needed, since de Rham complex is trivially formed by Fourier modes as their derivatives are obtained by a scalar multiplication. This leads completely analog to the electrostatic Vlasov–Poisson system to energy conservation with respect to the splitting error and momentum conservation to machine precision also for the electromagnetic case. In the following the particle discretization with PIF is discussed using the reduced 1d2v Vlasov–Maxwell model as an introductory example. A more detailed overview of the Vlasov–Maxwell system is given in appendix B.1.1 as well as the extension of PIF to six dimensions C.3.

3.4.1. Vlasov–Maxwell (1d2v)

We denote the spatial Fourier transform of the fields along with their back-transforms as

$$\begin{aligned}
\tilde{B}(k,t) :=& \frac{1}{L}\int_0^L B(x,t)\mathrm{e}^{-\mathrm{i}kx}\,\mathrm{d}x, \quad B(x,t) = \sum_k \tilde{B}(k,t)\mathrm{e}^{\mathrm{i}kx},\\
\tilde{E}_1(k,t) :=& \frac{1}{L}\int_0^L E_1(x,t)\mathrm{e}^{-\mathrm{i}kx}\,\mathrm{d}x, \quad E_1(x,t) = \sum_k \tilde{E}_1(k,t)\mathrm{e}^{\mathrm{i}kx},\\
\tilde{E}_2(k,t) :=& \frac{1}{L}\int_0^L E_2(x,t)\mathrm{e}^{-\mathrm{i}kx}\,\mathrm{d}x, \quad E_2(x,t) = \sum_k \tilde{E}_2(k,t)\mathrm{e}^{\mathrm{i}kx},
\end{aligned} \tag{3.94}$$

where the the discrete one dimensional wave vector k is given as

$$k = z\frac{2\pi}{L}, \quad z \in \mathbb{Z}. \tag{3.95}$$

We turn to the discretization of the Hamiltonian splitting of the Vlasov–Maxwell system, which is introduced in the appendix B in eqns.(B.47,B.48,B.50,B.49). Here each Hamiltonian is solved exactly in the time interval $(0,t)$, which corresponds to a time-step of length t.

- Kinetic energy ($d=1$), $\hat{\mathcal{H}}_{p_1} = \frac{1}{2}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{1,n}^2$
  Because the velocity V_1 is constant in eqn. (B.47) and therefore $\dot{x}_n(t) = v_{1,n}(0)$ we obtain $x_n(t) = x_n(0) + tv_{1,n}(0)$ and can integrate V_2 exactly:

$$\begin{aligned}
v_{2,n}(t) &= v_{2,n} - \int_0^t v_{1,n}(\tau)B(x_n(\tau),\tau)\mathrm{d}\tau\\
&= v_{2,n} - \int_0^t v_{1,n}(0)B(x_n(\tau),0)\mathrm{d}\tau\\
&= v_{2,n} - \sum_k \tilde{B}(k,0)v_{1,n}(0)\int_0^t \mathrm{e}^{\mathrm{i}kx_n(\tau)}\mathrm{d}\tau\\
&= v_{2,n} - \sum_k \tilde{B}(k,0)v_{1,n}(0)\int_0^t \mathrm{e}^{\mathrm{i}kx_n(0)+\tau v_{1,n}(0)}\mathrm{d}\tau\\
&= v_{2,n} - \sum_k \tilde{B}(k,0)v_{1,n}(0)\frac{1}{v_{1,n}(0)}\int_{x_n(0)}^{x_n(t)} \mathrm{e}^{\mathrm{i}ks}\mathrm{d}s\\
&= v_{2,n} - \sum_{k\neq 0} \tilde{B}(k,0)\frac{1}{\mathrm{i}k}\left[\mathrm{e}^{\mathrm{i}kx_n(t)} - \mathrm{e}^{\mathrm{i}kx_n(0)}\right] - t\tilde{B}(0,0)v_{1,n}(0).
\end{aligned} \tag{3.96}$$

The Fourier modes of $E_1(k,t)$ for $k \neq 0$ read then

$$\begin{aligned}
\tilde{E}_1(k,t) &= \tilde{E}_1(k,0) - \int_0^t \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n \dot{x}_n(\tau) \mathrm{e}^{-\mathrm{i}kx_n(\tau)} \mathrm{d}\tau \\
&= \tilde{E}_1(k,0) - \int_0^t \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{1,n}(0) \mathrm{e}^{-\mathrm{i}kx_n(\tau)} \mathrm{d}\tau \\
&= \tilde{E}_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{1,n}(0) \int_0^t \mathrm{e}^{-\mathrm{i}k(x_n(0)+tv_{1,n}(0))} \mathrm{d}\tau \\
&= \tilde{E}_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{1,n}(0) \frac{1}{v_{1,n}(0)} \int_{x_n(0)}^{x_n(t)} \mathrm{e}^{-\mathrm{i}ks} \mathrm{d}s \\
&= \tilde{E}_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n \frac{-1}{\mathrm{i}k} \left[\mathrm{e}^{-\mathrm{i}ks}\right]_{x_n(0)}^{x_n(t)} \\
&= \tilde{E}_1(k,0) + \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n \frac{1}{\mathrm{i}k} \left[\mathrm{e}^{-\mathrm{i}kx_n(t)} - \mathrm{e}^{-\mathrm{i}kx_n(0)}\right]
\end{aligned} \tag{3.97}$$

and for $k = 0$

$$\tilde{E}_1(0,t) = \tilde{E}_1(0,0) - \int_0^t \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n \dot{x}_{1,n}(0) \mathrm{d}\tau = \tilde{E}_1(0,0) - t \cdot \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{1,n}(0). \tag{3.98}$$

The entire discretization of $\mathcal{H}_{p_1}$ is then summarized in eqn. (3.99).

$$\begin{aligned}
x_n(t) &= x_n(0) + tv_{1,n}(0) \\
v_{2,n}(t) &= v_{2,n}(0) - \sum_k \tilde{B}(k,0) \frac{1}{\mathrm{i}k} \left[\mathrm{e}^{\mathrm{i}kx_n(t)} - \mathrm{e}^{\mathrm{i}kx_n(0)}\right] \\
\tilde{E}_1(k,t) &= \tilde{E}_1(k,0) + \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n \frac{1}{\mathrm{i}k} \left[\mathrm{e}^{-\mathrm{i}kx_n(t)} - \mathrm{e}^{-\mathrm{i}kx_n(0)}\right] \qquad \text{for } k \neq 0 \\
\tilde{E}_1(0,t) &= \tilde{E}_1(0,0) - t \cdot \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{1,n}(0)
\end{aligned} \tag{3.99}$$

- Kinetic energy ($d = 2$), $\hat{\mathcal{H}}_{p_2} = \frac{1}{2}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{2,n}^2$
  For this reduced model the system (B.48) is linear, such that the discretization is obtained straightforward in eqn. (3.100).

$$\begin{aligned}
v_{1,n}(t) &= v_{1,n}(0) + tv_{2,n}(0) \sum_k \tilde{B}(k,0) \mathrm{e}^{\mathrm{i}kx_n(0)} \\
E_2(k,t) &= E_2(k,0) - t\frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} w_n v_{2,n} \mathrm{e}^{-\mathrm{i}kx_n(0)} \qquad \text{for all } k
\end{aligned} \tag{3.100}$$

- Electric energy $\hat{\mathcal{H}}_E = \frac{1}{2}\sum_k L\left(|\tilde{E}_1(k,t)|^2 + |\tilde{E}_2(k,t)|^2\right)$
  Independent of the dimensionality the system (B.49) is always linear such that the exact

discrete solution reads

$$\begin{aligned} v_1(t) &= v_1(0) + t\sum_k \tilde{E}_1(k,t)\mathrm{e}^{\mathrm{i}kx}, \\ v_2(t) &= v_2(0) + t\sum_k \tilde{E}_2(k,t)\mathrm{e}^{\mathrm{i}kx}, \\ \tilde{B}(k,t) &= \tilde{B}(k,0) - t(\mathrm{i}k)\tilde{E}_2(k,0). \end{aligned} \tag{3.101}$$

- Magnetic energy $\hat{\mathcal{H}}_B = \frac{1}{2}\sum_k L|\tilde{B}_1(k,t)|^2$
  After Fourier transformation the solution to eqn. (B.50) is given in eqn. (3.102).

$$\tilde{E}_2(k,t) = \tilde{E}_2(k,0) - t(\mathrm{i}k)\tilde{B}(k,0) \tag{3.102}$$

Discretization of $\mathcal{H}_p$

In case we do not split $\mathcal{H}_p$ we face the system

$$\begin{aligned} \partial_t f + v_1\partial_x f - v_1 B(x,t)\partial_{v_2} f + v_2 B(x,t)\partial_{v_1} f &= 0 \\ \partial_t B(x,t) &= 0 \\ \partial_t E_1(x,t) &= -\int\int v_1 f(x,v_1,v_2,t)\mathrm{d}v_1\mathrm{d}v_2 \\ \partial_t E_2(x,t) &= -\int v_2 f(x,v,t)\mathrm{d}v, \end{aligned} \tag{3.103}$$

leading to the characteristics

$$\begin{aligned} \dot{x}_n(t) &= v_{1,n}(t) \\ \dot{v}_{1,n}(t) &= v_{2,n}q_n B(x_n(t),0) \\ \partial_t v_{2,n}(t) &= -v_{1,n}q_n B(x_n(t),0) \\ \partial_t E_1(k,t) &= -\frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \underbrace{v_{1,n}(\tau)}_{=\dot{x}_{1,n}(\tau)} \mathrm{e}^{-\mathrm{i}k(x_n(\tau))}\mathrm{d}\tau \\ \partial_t E_2(k,t) &= -\frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \underbrace{v_{2,n}(\tau)}_{=\dot{x}_{2,n}(\tau)} \mathrm{e}^{-\mathrm{i}k(x_n(\tau))}\mathrm{d}\tau. \end{aligned} \tag{3.104}$$

We can simplify the field integrals by substituting $s = x_n(\tau)$, and therefore $v_{1,n}\tau\mathrm{d}\tau = \mathrm{d}s$.

$$\begin{aligned} E_1(k,t) &= E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \dot{x}_{1,n}(\tau)\mathrm{e}^{-\mathrm{i}k(x_n(\tau))}\mathrm{d}\tau \\ &= E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_{x_n(0)}^{x(t)} \mathrm{e}^{-\mathrm{i}ks}\mathrm{d}s \\ &= E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \frac{1}{-ik}\left[\mathrm{e}^{-\mathrm{i}kx_n(t)} - \mathrm{e}^{-\mathrm{i}kx_n(0)}\right] \end{aligned} \tag{3.105}$$

For E_2 it is not possible to obtain an analytical expression, thus one can approximate this integral with the midpoint rule or other integrators.

Implicit midpoint for $\mathcal{H}_p$

As already noted in [19], in general the Hamiltonian $\mathcal{H}_p$ cannot be integrated exactly, which is the reason why it is again split into multiple components. But $\mathcal{H}_p$ contains the gyromotion such that it is unnatural to split it into separated steps for every spatial direction. We would favor the use of an exponential integrator [183, 184] in oder to take much larger time steps by integrating the gyromotion exactly. Such exponential time differencing schemes have already been successfully applied to the Vlasov–Poisson system in order to resolve the gyromotion [185] or fast oscillations in the electric field [186]. In our Hamiltonian framework possible candidates are symmetric implicit schemes [187], but we do not understand yet how to apply them. Therefore, we start with the design of a simple implicit integrator for $\mathcal{H}_p$ such that we learn the necessary steps on the way. We search for two trajectories $x(\tau)$ and $v(\tau)$ such that they consistently approximate the following system of ODEs,

$$\dot{x}(\tau) = v(\tau), \quad \dot{v}(\tau) = v(\tau) \times B(x(\tau), 0), \quad \tau \in [0, t]. \tag{3.106}$$

But $\dot{v}(\tau)$ and thus also $v(\tau)$ depends on $x(\tau)$ such that the coefficients to the Legendre series have to be chosen consistently to eqn. (3.106). Here the orthogonal Legendre polynomials are used in order to obtain spectral convergence when approximating the true trajectory of the particles. Using the roots of the Legendre polynomials and the corresponding weights of Gauss–Legendre quadrature, the Legendre series for $\dot{v}(\tau)$ can be expressed as a sum of Lagrange polynomials. In this context such a representation is favored because it is much more straightforward to approximate a trajectory via the values at some nodes. This node driven view-point corresponding to collocation yields the name Legendre–Gauss collocation methods [188]. Once both trajectories $x(\tau)$ and $v(\tau)$ are approximated consistently for $\tau \in [0, t]$ the Ampère increment, here for PIF,

$$E(k, t) = E(k, 0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \dot{x}(\tau) \mathrm{e}^{-\mathrm{i}k(x_n(\tau))} \mathrm{d}\tau \tag{3.107}$$

can be calculated exactly since all included functions and their polynomial degree is known. Whether Lagrange polynomials at nodes or coefficients to some series, the mechanism stays the same:

1. Chose an Ansatz for $v(\tau)$ and $x(\tau)$,
2. Interpolate from eqn. (3.106), determining the free parameters of the Ansatz for $v(\tau)$,
3. Determine the integral in eqn. (3.107) to machine precision.

The ultimate goal is to do this for an Ansatz that approximates the gyromotion very well, while retaining the spectral convergence. We continue with the simplest example by using the first Legendre polynomial $P_0(x) = 1$, which yields the implicit midpoint rule.

$$\begin{aligned}
\dot{v}(\tau) &\approx \frac{v(t) - v(0)}{t} \quad \left[= \sum_{n=0}^{0} a_n P_n(\tau) = a_0 \cdot 1\right], \quad \tau \in [0, t] \\
v(\tau) &= v(0) + (v(t) - v(0))\,\frac{\tau}{t} \\
v(t) &= v(0) + \int_0^t \dot{v}(\tau)\mathrm{d}\tau \\
&\approx v(0) + t\left[\dot{v}\left(\frac{t}{2}\right)\right] = v(0) + t\left[v\left(\frac{t}{2}\right) \times B\left(x\left(\frac{t}{2}\right), 0\right)\right] \\
&= v(0) + t\left(\frac{v(0) + v(t)}{2}\right) \times B\left(x\left(\frac{t}{2}\right), 0\right)
\end{aligned} \tag{3.108}$$

The same approximations are made for $x(t)$.

$$\begin{aligned}
\dot{x}(\tau) &\approx \frac{x(t)-x(0)}{t} \quad \left[= \sum_{n=0}^{0} b_n P_n(\tau) = b_0 \cdot 1 \right], \quad \tau \in [0,t] \\
x(\tau) &= x(0) + (x(t)-x(0))\,\frac{\tau}{t} \\
x(t) &= x(0) + \int_0^t \dot{x}(\tau)\mathrm{d}\tau \\
&\approx x(0) + t\left[\dot{x}\left(\frac{t}{2}\right)\right] = x(0) + tv\left(\frac{t}{2}\right) \approx x(0) + t\frac{v(0)+v(t)}{2}
\end{aligned} \tag{3.109}$$

Inserting the approximation yields the trajectory $x(\tau)$ with

$$\begin{aligned}
x(\tau) &= x(0) + \tau\frac{v(0)+v(t)}{2} \quad \forall \tau \in (0,t), \\
x\left(\frac{t}{2}\right) &= x(0) + \frac{t}{2}\frac{v(0)+v(t)}{2}, \\
x(t) &= x(0) + t\frac{v(0)+v(t)}{2}.
\end{aligned} \tag{3.110}$$

Therefore, $x\left(\frac{t}{2}\right)$ can be used in eqn. (3.108). An implicit system of equations for $v(t)$ and $x(t)$ is then obtained.

$$\begin{aligned}
v(t) &= v(0) + t\left(\frac{v(0)+v(t)}{2}\right) \times B\left(x(0) + t\frac{v(0)+v(t)}{4}\ ,\ 0\right) \\
x(t) &= x(0) + t\frac{v(0)+v(t)}{2}
\end{aligned} \tag{3.111}$$

In case of the reduced 1d2v Vlasov–Maxwell this reads

$$\begin{aligned}
v_1(t) &= v_1(0) + t\left(\frac{v_2(0)+v_2(t)}{2}\right) \cdot B\left(x(0) + t\frac{v(0)+v(t)}{4}\ ,\ 0\right), \\
v_2(t) &= v_2(0) - t\left(\frac{v_1(0)+v_1(t)}{2}\right) \cdot B\left(x(0) + t\frac{v(0)+v(t)}{4}\ ,\ 0\right), \\
x(t) &= x(0) + t\frac{v_1(0)+v_1(t)}{2}.
\end{aligned} \tag{3.112}$$

The implicit system can be solved by Picard iterations as fixed point $F(v(t)) + v(t) = v(t)$, where F is given with $v(0) = (v_0^1, v_0^2, v_0^3)^t$ as

$$\begin{aligned}
\tilde{x}(v) &:= x^0 + t\frac{v^0+v}{4} \\
F(v) &= v^0 + \frac{t}{2}\frac{q}{m}\left(v^0+v\right) \times B(\tilde{x}(v),0) - v \\
&= \begin{pmatrix} v_1^0 \\ v_2^0 \\ v_3^0 \end{pmatrix} + \frac{t}{2}\frac{q}{m}\begin{pmatrix} (v_2^0+v_2)B_3(\tilde{x}(v),0) - (v_3^0+v_3)B_2(\tilde{x}(v),0) \\ (v_3^0+v_3)B_1(\tilde{x}(v),0) - (v_1^0+v_1)B_3(\tilde{x}(v),0) \\ (v_1^0+v_1)B_2(\tilde{x}(v),0) - (v_2^0+v_2)B_1(\tilde{x}(v),0) \end{pmatrix} - \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}
\end{aligned} \tag{3.113}$$

The cross product is denoted using a skew symmetric matrix $v \times B = \lfloor v \rfloor_\times \cdot B = -B \times v = -\lfloor B \rfloor_\times \cdot v$. Our peculiar definition of F is more useful when we apply the Newton method

$v^{k+1} := v^k - \mathrm{D}F(v^k)^{-1}F(v^k)$, where the Jacobi matrix $\mathrm{D}F$ reads

$$
\begin{aligned}
\mathrm{D}\tilde{x}(v) &= \frac{t}{4}\mathrm{id}_{3\times 3} \\
\mathrm{D}F(v) &= \frac{t}{2}\frac{q}{m}\left[-\lfloor B(\tilde{x}(v),0)\rfloor_\times + \lfloor v^0+v\rfloor_\times \cdot \mathrm{D}B(\tilde{x}(v),0)\cdot \mathrm{D}\tilde{x}(v)\right] - \mathrm{id}_{3\times 3} \\
&= \frac{t}{2}\frac{q}{m}\left[-\lfloor B(\tilde{x}(v),0)\rfloor_\times + \frac{t}{4}\lfloor v^0+v\rfloor_\times \cdot \mathrm{D}B(\tilde{x}(v),0)\right] - \mathrm{id}_{3\times 3} \\
&= -\begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix} + \frac{t}{2}\frac{q}{m}\Bigg[\begin{pmatrix} 0 & B_3(\tilde{x}(v),0) & -B_2(\tilde{x}(v),0) \\ -B_3(\tilde{x}(v),0) & 0 & B_1(\tilde{x}(v),0) \\ B_2(\tilde{x}(v),0) & -B_1(\tilde{x}(v),0) & 0 \end{pmatrix} \\
&\quad + \frac{t}{4}\begin{pmatrix} 0 & -(v_3^0+v_3) & (v_2^0+v_2) \\ (v_3^0+v_3) & 0 & -(v_1^0+v_1) \\ -(v_2^0+v_2) & (v_1^0+v_1) & 0 \end{pmatrix} \\
&\quad \cdot \begin{pmatrix} \partial_{x_1}B_1(\tilde{x}(v),0) & \partial_{x_2}B_1(\tilde{x}(v),0) & \partial_{x_3}B_1(\tilde{x}(v),0) \\ \partial_{x_1}B_2(\tilde{x}(v),0) & \partial_{x_2}B_2(\tilde{x}(v),0) & \partial_{x_3}B_2(\tilde{x}(v),0) \\ \partial_{x_1}B_3(\tilde{x}(v),0) & \partial_{x_2}B_3(\tilde{x}(v),0) & \partial_{x_3}B_3(\tilde{x}(v),0) \end{pmatrix}\Bigg].
\end{aligned}
\tag{3.114}
$$

The approximated trajectories $x(\tau)$ and $v(\tau)$ are then used in the Ampère equation for the increment of the electric fields. The integral in the first dimension is merely a line integral such that it is straightforward to evaluate if the anti-derivative is at hand, but this is actually a special case. Yet with the implicit midpoint discretization a consistency problem arises since $v_{1,n}(\tau) \neq \dot{x}_{1,n}(\tau)$, and we sometimes see the Ampère equation in the following incorrect form

$$
E_1(k,t) = \left[E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \underbrace{v_{1,n}(\tau)}_{wrong} \mathrm{e}^{-\mathrm{i}k(x_n(\tau))}\mathrm{d}\tau\right]. \tag{3.115}
$$

In the correct form, stemming from discretization on the level of the Lagrangian for $\mathcal{H}_p$, $v_{1,n}(\tau)$ is substituted by $\dot{x}_{1,n}(\tau)$.

$$
\begin{aligned}
E_1(k,t) &= E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \dot{x}_{1,n}(\tau)\mathrm{e}^{-\mathrm{i}k(x_n(\tau))}\mathrm{d}\tau \\
&= E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \frac{v_{1,n}(0)+v_{1,n}(t)}{2}\mathrm{e}^{-\mathrm{i}k\left(x(0)+\tau\frac{v_{1,n}(0)+v_{1,n}(t)}{2}\right)}\mathrm{d}\tau \\
&= E_1(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \frac{1}{-ik}\left[\mathrm{e}^{-\mathrm{i}kx_n(t)} - \mathrm{e}^{-\mathrm{i}kx_n(0)}\right]
\end{aligned}
\tag{3.116}
$$

For a consistent discretization with respect to the six dimensional model we approximate $\dot{x}_{2,n}(\tau)$ as $\dot{x}_{2,n}(\tau) = \frac{v_{2,n}(0)+v_{2,n}(t)}{2}$ and use the reduction on 1d2v afterwards. Thus, the

second component presents us a more general situation where we face the following integral

$$\begin{aligned}
E_2(k,t) &= E_2(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \dot{x}_{2,n}(\tau)\mathrm{e}^{-\mathrm{i}k(x_n(\tau))}\mathrm{d}\tau \\
&= E_2(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \frac{v_{2,n}(0)+v_{2,n}(t)}{2}\mathrm{e}^{-\mathrm{i}k\left(x(0)+\tau\frac{v_{1,n}(0)+v_{1,n}(t)}{2}\right)}\mathrm{d}\tau \\
&= E_2(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \frac{v_{2,n}(0)+v_{2,n}(t)}{v_{1,n}(0)+v_{1,n}(t)}\int_0^t \frac{v_{1,n}(0)+v_{1,n}(t)}{2}\mathrm{e}^{-\mathrm{i}k\left(x(0)+\tau\frac{v_{1,n}(0)+v_{1,n}(t)}{2}\right)}\mathrm{d}\tau \\
&= E_2(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \frac{v_{2,n}(0)+v_{2,n}(t)}{v_{1,n}(0)+v_{1,n}(t)}\frac{1}{-ik}\left[\mathrm{e}^{-\mathrm{i}kx_n(t)} - \mathrm{e}^{-\mathrm{i}kx_n(0)}\right],
\end{aligned} \tag{3.117}$$

which is not a standard line integral that can be simplified by substitution. In three dimensions and for $k\cdot(v_{j,n}(0)+v_{j,n}(t)) \neq 0$, the increment is then given as

$$\begin{aligned}
E_j(k,t) &= E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \dot{x}_{j,n}(\tau)\mathrm{e}^{-\mathrm{i}k\cdot x_n(\tau)}\mathrm{d}\tau \\
&= E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \frac{v_{j,n}(0)+v_{j,n}(t)}{2}\mathrm{e}^{-\mathrm{i}k\cdot\left(x_(0)+\tau\frac{v_n(0)+v_n(t)}{2}\right)}\mathrm{d}\tau \\
&= E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \mathrm{e}^{-\mathrm{i}k\cdot x_n(0)}\int_0^t \frac{v_{j,n}(0)+v_{j,n}(t)}{2}\mathrm{e}^{-\mathrm{i}\frac{\tau}{2}k\cdot(v_n(0)+v_n(t))}\mathrm{d}\tau \\
&= E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \mathrm{e}^{-\mathrm{i}k\cdot x_n(0)}\frac{v_{j,n}(0)+v_{j,n}(t)}{-\mathrm{i}k\cdot(v_n(0)+v_n(t))}\left[\mathrm{e}^{-\mathrm{i}\frac{\tau}{2}k\cdot(v_n(0)+v_n(t))}\right]_{\tau=0}^{\tau=t} \\
&= E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \frac{v_{j,n}(0)+v_{j,n}(t)}{-\mathrm{i}k\cdot(v_n(0)+v_n(t))}\left[\mathrm{e}^{-\mathrm{i}k\cdot x_n(t)} - \mathrm{e}^{-\mathrm{i}k\cdot x_n(0)}\right].
\end{aligned} \tag{3.118}$$

For the implicit midpoint method this is still the case yet for higher order methods the trajectory $x(\tau)$ becomes a polynomial of higher degree such that analytic integration for PIF relies on the expensive complex error function erf. In general Gauss–Legendre or Gauss-Lobatto quadrature can be used. The latter one uses the endpoints $\tau \in 0, t$ and may, therefore, be more practicable, see [177][p.888]. For global orthogonal polynomials - spectral methods - the number of quadrature nodes increases directly with the polynomial degree of T_n or P_n but the number of nodes required for exact integration is known a priori. The cubic B-splines used in [19] require just very few quadrature nodes for exact integration, but since they are discontinues the trajectory $x(\tau)$ has to be integrated piecewise on each cell. Thus, we conclude that more complicated integrators are much easier to implement with global Fourier or Chebyshev methods.

3.4.2. Multispecies Vlasov–Maxwell (1d2v)

We extend the reduced three dimensional model to a simulation containing two species, where each species is simulated with the same number of markers. The general initial conditions

for electrons and the ions are given in eqn. (3.119).

$$\begin{aligned} f_e(x, v_1, v_2, t=0) &= \frac{1+\epsilon_e \cos(kx)}{2\pi\sigma_1\sigma_2^2} \mathrm{e}^{-\frac{v_1^2}{2\sigma_1^2}} \left(\delta \mathrm{e}^{-\frac{(v_2-v_{0,1})^2}{2\sigma_2^2}} + (1-\delta)\mathrm{e}^{-\frac{(v_2-v_{0,2})^2}{2\sigma_2^2}} \right) \\ f_i(x, v_1, v_2, t=0) &= \frac{1+\epsilon_i \cos(kx)}{2\pi\sigma_i^2} \mathrm{e}^{-\frac{v_1^2+v_2^2}{2\sigma_i^2}} \\ B_3(x, t=0) &= \beta_r \cos(kx) + \beta_i \sin(kx) \\ E_2(x, t=0) &= \alpha_r \cos(kx) + \alpha_i \sin(kx) \\ \partial_x E_1(x, t=0) &= \sum_s \frac{q_s}{e} \int_{\mathbb{R}^d} f_s(x, v_1, v_2, t) \, \mathrm{d}v \end{aligned} \tag{3.119}$$

For a simplified model we set $q_e = -1$, $q_i = 1$ and $\left(\frac{c}{v_{th,e}}\right)^2 = 1$. The terms $v_{th,e} = 1$, $T_e = 1$ and $m_e = 1$ can be set but will always cancel out since everything is relative to the electrons. The ions are usually colder and heavier then the electrons, therefore σ_i is determined by the mass and temperature ratio

$$\sigma_i = \sqrt{\frac{T_i}{T_e}\frac{m_e}{m_i}}, \quad L = \frac{k}{2\pi}. \tag{3.120}$$

In the following we investigate energy and momentum conservation and the variances of the electric field E_1. The number of Fourier modes is set to $N_f = 3$. We use the symmetric second order Strang splitting described in [19] with the additional symmetric composition resulting in a fourth order Strang splitting.
At the end of the simulation we check the conservation of the Poisson structure, which is always conserved up to a roundoff error. Additionally the integrated sample variance of the resulting electric field is estimated separately for the electron and ion contribution. The same analysis is done on the Ampère equation. Since the contribution of the current density of each species to the electric field via the Ampère equation is integrated over the time of one time-step, the same integration has to be applied for the integrated variance in eqn. (3.121).

$$\mathrm{IVAR}\left[\int_t^{t+\Delta t} \partial_t \hat{E}_1(x, \tau) \, \mathrm{d}\tau\right] = \sum_s \mathrm{IVAR}\left[\int_t^{t+\Delta t} j_{1,s}(x, \tau) \, \mathrm{d}\tau\right] \tag{3.121}$$

We consider four test-cases with parameters given in fig. 3.21. The results are shown in table 3.1 and fig. 3.22. In all cases energy is well conserved, but contrary to the 1d1v electrostatic solver the momentum error is not at roundoff. The ion-acoustic wave as the true multi-scale test-case can already be observed in the beginning of the Weibel instability on the electric field E_1. We use 20 times less particles than [19], but due to the restriction onto few Fourier modes the noise level is moderate, although it can clearly be seen in the ion-acoustic wave. The new insight here, is that for the Poisson equation the integrated variance is independent of the species. This makes sense, since apart from the oscillations the spatial density is uniformly in both cases. Yet for the Ampère equation there is a much larger difference, stemming from the scale difference in the thermal velocity. The standard deviation is proportional to $\sqrt{N_p}$, which means, e.g. in the Weibel test-case, that the amount of ions can be decreased by a factor of 2 in oder to have the same noise level.
Here no control variate is used and, of course, if the ions exhibit no great perturbation a simple Gaussian based control variate then allows for less ion markers. But with two species only a factor up to two can be gained hence we do not proceed further in this direction.

default	$\epsilon_e, \epsilon_i, \alpha_r, \alpha_i, \beta_r, \beta_i, v_{0,1}, v_{0,2}, \delta, B_0 = 0,\ c = 1$ $m_e = 1,\ T_e = 1,\ m_i = 1038, T_i = 0.1T_e, \sigma_1, \sigma_2 = 1$ $N_p = 10^5, N_f = 3, \Delta t = 0.05$
Landau	$\epsilon_e = 0.5,\ k = 0.5,$
Weibel	$\beta_r = -10^{-3},\ k = 1.25,\ \sigma_1 = \frac{0.02}{\sqrt{2}},\ \sigma_2 = \sqrt{12}\sigma_1,$
Weibel streaming	$\sigma_1 = \sigma_2 = \frac{0.1}{\sqrt{2}},\ k = 0.2,\ \beta_i = 10^{-3},\ v_{0,1} = 0.5,\ v_{0,2} = -0.1,\ \delta = \frac{1}{6}$
Ion acoustic	$k = 0.6283185,\ L = 10,\ m_i = 200,\ T_i = 10^{-4}T_e, \epsilon_i = 0.1, \alpha_i = \frac{0.2}{k},\ N_f = 1$
Light wave	$k = 0.4,\ c = 10,\ \beta_r = 0.001,\ \Delta t = 0.001$

Figure 3.21.: Parameters for multi-species Vlasov–Maxwell(1d2v) test-cases

		Landau	Weibel	Weibel streaming	Ion acoustic	Light
Ampère	ion	0.0035	2.5e-05	9.1e-04	0.0022	0.0034
	electron	2.7e-05	1.1e-05	6.6e-05	6.1e-06	3.3e-05
Poisson	ion	0.0078	0.003	0.018	0.005	0.0079
	electron	0.0074	0.003	0.018	0.005	0.0079
Poisson eqn. error		1.8e-14	1.5e-17	3.0e-15	7.8e-16	6.4e-18

Table 3.1.: Standard deviations from the integrated sample variance of the electric field for Poisson, respectively the increment for Ampère at the last time step. Additionally the error for the discrete conservation of the Poisson equation is given.

3.4.3. Semi-implicit Vlasov–Maxwell (1d2v)

Instead of splitting the Hamiltonian $\mathcal{H}_p$ into the components $\mathcal{H}_{p_1}$ and $\mathcal{H}_{p_2}$ we use the the implicit midpoint discretization of $\mathcal{H}_{p_1}$ from eqn. (3.112). The overall Hamiltonian splitting into $\mathcal{H}_p$, $\mathcal{H}_E$ and $\mathcal{H}_B$ is kept, hence the scheme is semi-implicit. We use the initial conditions (3.119) and the parameters given in fig. 3.21 from the multi-species example, except that the ions are set as a constant background precisely as in [19]. The most challenging test-case, the Weibel streaming instability, is presented in fig. 3.23. Although the time step is with $\Delta t = 0.5$ quite large there is no visible difference between the two schemes. A closer look reveals a slightly smaller energy error for the implicit method in fig. 3.24. The implicit equations are solved using Picard (fixed point) iterations or a Newton method, where we found the latter to be much more efficient in the nonlinear phase, see fig. 3.25. With direct integration of the Ampère equations using primitives, the Newton method takes with 234.3s roughly double as long as the splitting 114.5s. The Picard iterations are with 424.6s too slow. This might also depend on the initial guess which is an explicit Euler step. Instead of using the primitive function the Ampere equation ((3.122) can also be integrated using a quadrature rule with weights and knots (ω_m, τ_m) according to

$$\begin{aligned} E_j(k,t) &= E_j(k,0) - q\frac{1}{L}\int_0^t \int_0^L \int_{\mathbb{R}^2} v_j f_p(x,v,t) \mathrm{e}^{-\mathrm{i}k\cdot x}\ \mathrm{d}v\ \mathrm{d}x\ \mathrm{d}\tau \\ &= E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \int_0^t \frac{v_{j,n}(0) + v_{j,n}(t)}{2} \mathrm{e}^{-\mathrm{i}k\cdot\left(x_n(0) + \tau\frac{v_n(0)+v_n(t)}{2}\right)} \mathrm{d}\tau \\ &\approx E_j(k,0) - \frac{1}{L}\frac{1}{N_p}\sum_{n=1}^{N_p} q_n w_n \sum_m \omega_m \frac{v_{j,n}(0) + v_{j,n}(t)}{2} \mathrm{e}^{-\mathrm{i}k\cdot\left(x_j(0) + \tau_m\frac{v_n(0)+v_n(t)}{2}\right)}. \end{aligned} \tag{3.122}$$

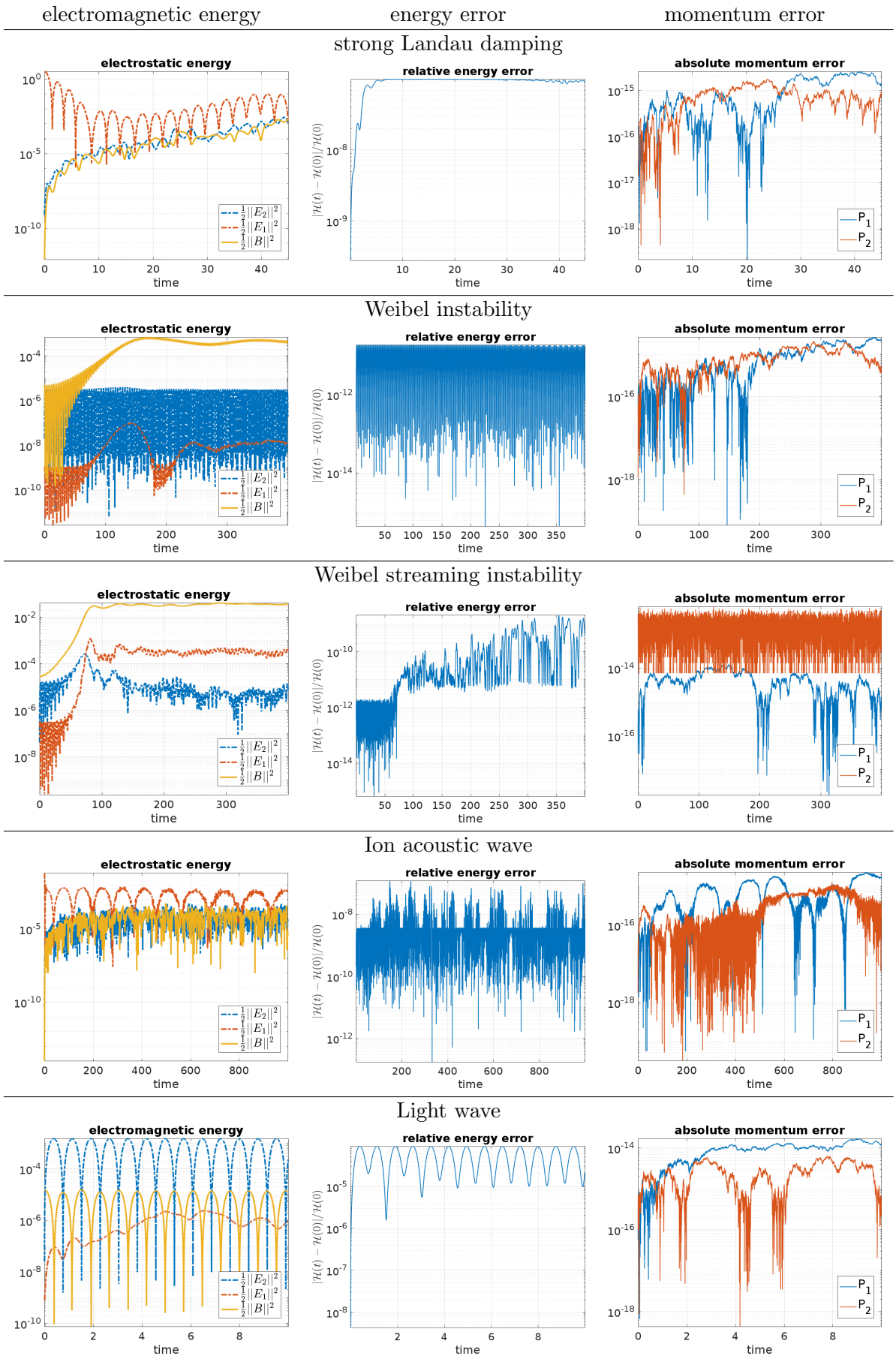
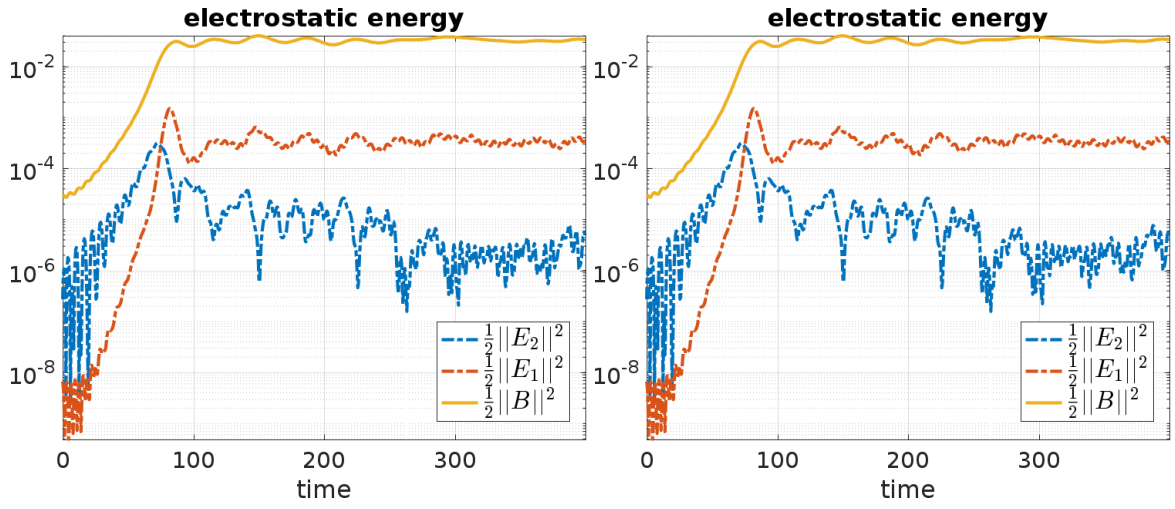


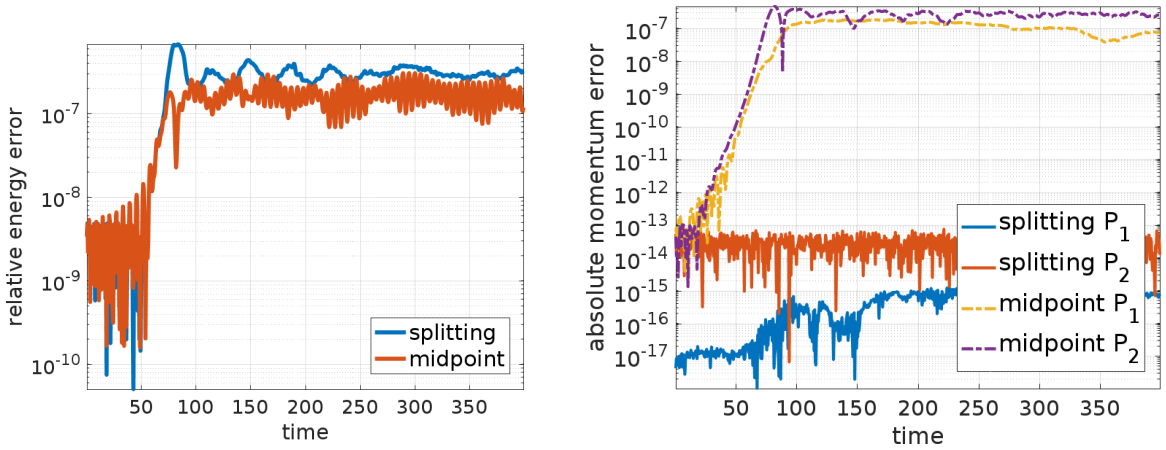
Figure 3.22.: Electrostatic and magnetic energy, relative energy error and the momentum error in the two velocity components for the four test-cases of the Vlasov–Maxwell 1d2v PIF.



(a) splitting $\mathcal{H}_{p,1}$ and $\mathcal{H}_{p,2}$ (b) implicit midpoint for $\mathcal{H}_p$

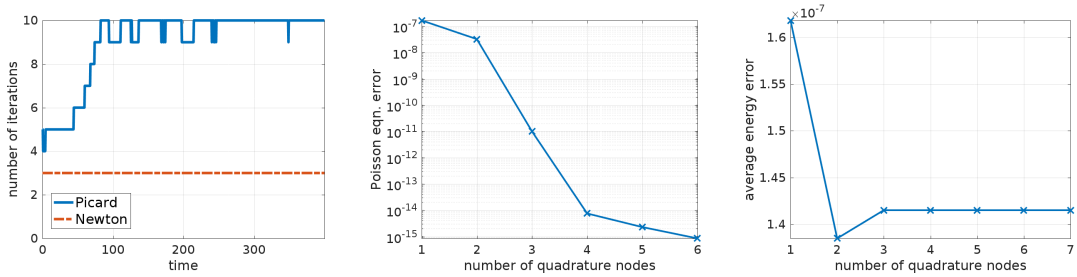
Figure 3.23.: Electrostatic and magnetic field energies under the Weibel streaming instability with $\Delta t = 0.5$ for the standard splitting (a) and the implicit midpoint method (b).

At this point a discussion on the exact conservation of Gauss' law is needed, since it is unclear to which precision it is actually required. The simulation is anyhow subjected to roundoff and the numerical stability does not depend on the level of precision, opposed to the long term stability. For polynomial basis functions it is a priori clear which quadrature rule is needed for exact integration of Ampere's equation where on the other hand for the exponential function this question is a bit more involved. Given the large Monte Carlo error, precision can be sacrificed while increasing a small bias, which does for example not appear in the energy error, see fig. 3.25. Nevertheless for any quadrature rule, the achieved precision depends on the density f and hence a certain precision cannot be guaranteed a priori. Here this is quite attractive, because the solution converges already for few quadrature nodes. Nevertheless the achieved precision depends on the scenario (the density f) and is not a priori known, such there is no universal guarantee for long term stability. On the other hand introducing some adaptivity solves this potential problem easily. If we are already comfortable with sacrificing precision, we realize that the additional integration over time yields a four dimensional integral(1d2v1t), see eqn. (3.122). Instead of applying only a three dimensional Monte Carlo estimator and a separate quadrature for time we can raise the dimensionality for the Monte Carlo integration to four by drawing a random time $\tau \in (0, t)$. Depending on the number of particles this will not result in sufficient precision on Gauss' law such that a randomized quadrature rule in time has to be used. This idea corresponds to the methods applied for the gyroaverage operator and the linearized Vlasov–Poisson system. The randomized quadrature rules are explained in detail in section A.1. Here the randomized quadrature rules win in the large particle limit, which is not always reached, see fig. 3.26.
In future research one can expand the Monte Carlo integral into the time domain because there is no curse of dimensionality such that one can *sample* the gyromotion especially when using exponential time differencing schemes.



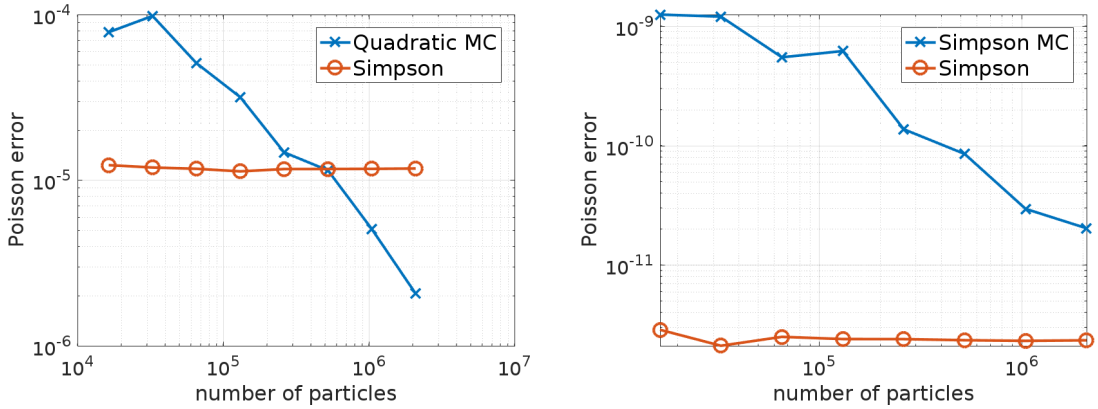
(a) energy error (b) absolute momentum error

Figure 3.24.: Comparison of the energy and momentum error between the explicit splitting (a) and the implicit midpoint method (b).



(a) iterations in the implicit solver (b) Conservation of Gauss' law under quadrature (c) Average energy error under quadrature

Figure 3.25.: The Newton method converges much faster than the Picard iterations and is also more efficient since for PIF the derivative $\partial_x B(x,t)$ is obtained by only one complex multiplication. For exact integration of the Ampère equation Gauss–Legendre quadrature with a varying number of quadrature nodes is used, which yields a fast decreasing error on the Poisson equation over the entire simulation time $t_{tmax} = 400$. Here it is not necessary to conserve the Poisson equation up to machine precision, since the energy error barely changes for fewer quadrature nodes. (Weibel streaming instability).



(a) nonlinear Landau damping (b) Weibel streaming instability

Figure 3.26.: Simpson's Rule and the Monte Carlo estimator with a quadratic control variate use both three quadrature nodes yielding the same costs. Although the Simpson rule has higher accuracy it can be outperformed by the unbiased Monte Carlo estimator in the many particle limit. Here this works for nonlinear Landau damping (a) but seem to require much more particles for the Weibel streaming instability (b).

3.5. Mixing PIF and PIC

When discretizing arbitrary domains with PIC any boundary condition can be incorporated, but it is also possible to use PIF in periodic directions. In the following we introduce a mixture between PIC and PIF, where for typical geometries of fusion devices PIF is used in the toroidal and poloidal direction and B-spline finite elements for the radial coordinate.

3.5.1. General coordinate elliptic Fourier-FEM solver

The most general field equation, for a field Φ and charge density ρ, is a general elliptic equation

$$-\operatorname{div}(A\nabla\Phi) + b\cdot\nabla\Phi + c\Phi = \rho, \tag{3.123}$$

where A denotes a 3×3 tensor, b a vector field and c a scalar field. The weak form of eqn. (3.123) is given as

$$\langle(\nabla\Phi)^\dagger\cdot A\cdot\nabla\varphi\rangle + \langle b\cdot\nabla\Phi,\varphi\rangle + \langle c\Phi,\varphi\rangle = \langle\rho,\varphi\rangle. \tag{3.124}$$

In the following the Galerkin method using splines and Fourier modes is used to discretize eqn. (3.124). The Poisson equation is obtained with A as the identity, $b=0$ and $c=0$ as a special case. We are interested in the geometry of fusion devices, which are mostly described by a global mapping in coordinates $\xi=(r,\theta,\varphi)$, where r is bounded and θ and φ are periodic. Transforming eqn. (3.124) in the previously introduced notation yields

$$\begin{aligned}\int_{\tilde\Omega}\tilde\nabla\tilde\Phi(\xi)^\dagger\cdot J_T^{-1}(\xi)\tilde A(\xi)J_T^{-\dagger}(\xi)\cdot\tilde\nabla\tilde\varphi(\xi)\,\det(J_T(\xi))\,\mathrm{d}\xi\\ +\int_{\tilde\Omega}\tilde b(\xi)^\dagger\cdot J_T^{-1}(\xi)J_T^{-\dagger}\cdot\tilde\nabla\tilde\Phi(\xi)\tilde\varphi(\xi)\,\det(J_T(\xi))\,\mathrm{d}\xi\\ +\int_{\tilde\Omega}\tilde c(\xi)\tilde\Phi(\xi)\tilde\varphi(\xi)\,\det(J_T(\xi))\,\mathrm{d}\xi=\int_{\tilde\Omega}\tilde\rho(\xi)\tilde\varphi(\xi)\,\det(J_T(\xi))\,\mathrm{d}\xi.\end{aligned} \tag{3.125}$$

We already suppose that the boundary conditions are incorporated in the choice of basis functions. Fourier modes are the obvious choice in the poloidal φ and toroidal θ direction. For the one dimensional VP-PIC code periodic B-splines of degree d on a uniform grid were used, but for the radial direction r non-periodic basis functions are needed. In this case we define

- ψ_r^l as the l-th B-spline on a mesh over $[r_{\min}, r_{\max}]$ with N_r cells.
- $\psi_\theta^m(\theta) := e^{-i\theta m}$ as the basis of the m^{th} Fourier mode over $[0, 2\pi]$
- $\psi_\varphi^n(\varphi) := e^{-i\varphi n}$ as the basis of the n^{th} Fourier mode over $[0, 2\pi]$.

The basis functions $\psi_{l,m,n}$ are then obtained by the tensor product

$$\psi_{l,m,n} = \psi_r^l \cdot \psi_\theta^m \cdot \psi_\varphi^n \qquad 1 \le l \le N_r,\ 1 \le m \le N_\theta,\ 1 \le n \le N_\varphi. \tag{3.126}$$

In most cases a B-spline library will take a set of knots and the appropriate boundary conditions are obtained by multiple knots. The basis functions in radial direction $\psi_r^l(r),\ l = 1, \dots, N_r$ are given as B-splines on

$$\underbrace{r_{\min} \dots r_{\min}}_{\text{spline degree}} r_1\ r_2\ r_3\ \dots\ r_{\max} \tag{3.127}$$

with $r_k \in [r_{\min}, r_{\max}]$. The construction of the B-splines is found in Deboor's book [49][pp. 87-90]. Not duplicating $r_{\max}$ imposes natural Dirichlet boundary conditions at $r_{\max}$ and homogeneous Neumann boundary conditions at $r_{\min}$. This choice is, of course, not fixed and also mixed boundary conditions for different Fourier modes are possible e.g. to resolve the singularity in a polar mesh.
Inserting the basis functions into the weak form (3.125) yields the mass matrix $\mathcal{M}$ incorporating the entire general elliptic equation.

$$\begin{aligned}
\mathcal{M}_{(l_1,m_1,n_1),(l_2,m_2,n_2)} = \\
&\int_0^{2\pi}\int_0^{2\pi}\int_{r_{\min}}^{r_{\max}} \begin{pmatrix} \psi_r^{l_1}(r) \\ \mathrm{i}m_1 \mathrm{e}^{\mathrm{i}m_1\theta} \\ \mathrm{i}n_1 \mathrm{e}^{\mathrm{i}n_1\varphi} \end{pmatrix}^t \left[J_T^{-1}\tilde{A}J_T^{-\dagger}\right](r,\theta,\varphi) \begin{pmatrix} \psi_r^{l_2}(r) \\ -\mathrm{i}m_2 \mathrm{e}^{-\mathrm{i}m_2\theta} \\ -\mathrm{i}n_2 \mathrm{e}^{-\mathrm{i}n_2\varphi} \end{pmatrix} \\
&\quad + \left[bJ_T^{-1}J_T^{-\dagger}\right](r,\theta,\varphi) \begin{pmatrix} \psi_r^{l_1}(r) \\ \mathrm{e}^{\mathrm{i}m_1\theta} \\ \mathrm{e}^{\mathrm{i}n_1\varphi} \end{pmatrix} \psi_r^{l_2}(r)\mathrm{e}^{-\mathrm{i}(m_2\theta+n_2\varphi)} \\
&\quad + c(r,\theta,\varphi)\psi_r^{l_1}(r)\psi_r^{l_2}(r)\mathrm{e}^{\mathrm{i}[(m_1-m_2)\theta+(n_1-n_2)\varphi]}\ \det(J_T(r,\theta,\varphi))\ \mathrm{d}r\mathrm{d}\theta\mathrm{d}\varphi
\end{aligned} \tag{3.128}$$

For most curvilinear coordinates, $\mathcal{M}$ will be a dense matrix with respect to the Fourier modes. This is the disadvantage of spectral Galerkin methods [54], but there are remedies available using preconditioning based on finite differences [189]. Since the number of Fourier modes in the toy models presented in this work remains small, we do not have to deal with large dense and ill-conditioned matrices.
Nevertheless, at this point we have to discuss the coupling of Fourier modes in different geometries. For this we study the spline-spectral Galerkin mass matrix arising from Poisson equation in different domains. Depending on the coordinate transformation, the tensor $J_T^{-1}J_T^{-t}$ along with $\det(J_T(r,\theta,\varphi)$ will destroy the orthogonality of the Fourier modes yielding full matrices. Additionally, Fourier filtering is not natural anymore in the sense that the Fourier modes do not correspond to the exact eigenfunctions of the Laplace operator for

arbitrary geometries. In polar coordinates the Fourier modes in θ still decouple yielding the sparse matrix

$$\begin{aligned}\mathcal{K}_{(l_1,m_1),(l_2,m_2)} &:= \int_0^{2\pi}\int_0^{r_{\max}} \nabla_{(r,\varphi)}\psi_{l_1,m_1} J_T^{-1} J_T^{-t} \left(\nabla_{(r,\varphi)}\psi_{l_1,m_1}\right)^\dagger \; r \; \mathrm{d}r\mathrm{d}\theta \\ &= \int_0^{2\pi}\int_0^{r_{\max}} \begin{pmatrix} (\partial_r\psi_r^{l_1})\psi_\theta^{m_1} \\ \psi_r^{l_1}(\partial_\theta\psi_\theta^{m_1}) \end{pmatrix}^t \begin{pmatrix} r & 0 \\ 0 & \frac{1}{r} \end{pmatrix} \begin{pmatrix} (\partial_r\psi_r^{l_2})\psi_\theta^{m_2 *} \\ \psi_r^{l_2}(\partial_\theta\psi_\theta^{m_2})^* \end{pmatrix} \mathrm{d}r\mathrm{d}\theta \\ = \int_0^{2\pi}\int_0^{r_{\max}} & r(\partial_r\psi_r^{l_1})(\partial_r\psi_r^{l_2})\psi_\theta^{m_1}\psi_\theta^{m_2} + \frac{1}{r}\psi_r^{l_1}\psi_r^{l_2}(-\mathrm{i}m_1)(-\mathrm{i}m_2)^*\psi_\theta^{m_1}\psi_\theta^{m_2 *}\mathrm{d}r\mathrm{d}\theta \\ &= 2\pi\;\delta_{m_1,m_2}\left(\underbrace{\int_{\Omega_r} r(\partial_r\psi_r^{l_1})(\partial_r\psi_r^{l_2})dr}_{:=s_1} + \underbrace{m_1 m_2\int_{\Omega_r}\frac{1}{r}\psi_r^{l_1}\psi_r^{l_2}dr}_{:=s_2}\right). \end{aligned} \tag{3.129}$$

But for the common pseudo-toroidal coordinates the stiffness matrix has a more involved structure

$$\begin{aligned}\mathcal{K}_{(l_1,m_1,n_1),(l_2,m_2,n_2)} = \int_0^{2\pi}\psi_\varphi^{n_1}\left(\psi_\varphi^{n_2}\right)^*\mathrm{d}\varphi \cdot \int_0^{2\pi}\int_0^{r_{\max}}\psi_\theta^{m_1}\left(\psi_\theta^{m_2}\right)^* \cdot \\ \left[\partial_r\psi_r^{l_1}\partial_r\psi_r^{l_2}(\cos(\theta)r^2 + R_0 r) + \frac{n_1 n_2 r^2 + m_1 m_2\left(R_0 + r\cos\theta\right)^2}{\left(R_0 + r\cos\theta\right)r}\psi_r^{l_1}\psi_r^{l_2}\right]\mathrm{d}r\mathrm{d}\theta. \end{aligned} \tag{3.130}$$

Since toroidal coordinates may often be hard-coded for simpler research codes, we have a closer look and split eqn. (3.130) in three separate parts (3.131), (3.132) and (3.133).

$$\begin{aligned} s_1 = \int_0^{2\pi}\int_0^{r_{\max}}\psi_\theta^{m_1}\left(\psi_\theta^{m_2}\right)^*\partial_r\psi_r^{l_1}\partial_r\psi_r^{l_2}\left(\cos(\theta)r^2 + R_0 r\right)\mathrm{d}r\mathrm{d}\theta = \\ \int_0^{r_{\max}}\int_0^{2\pi}\mathrm{e}^{-\mathrm{i}(m_1-m_2)\theta}\partial_r\psi_r^{l_1}\partial_r\psi_r^{l_2}\left(\cos(\theta)r^2 + R_0 r\right)\mathrm{d}r\mathrm{d}\theta = \\ \begin{cases} 2\pi R_0\int_0^{r_{\max}}\partial_r\psi_r^{l_1}\partial_r\psi_r^{l_2} r\;\mathrm{d}r & \text{for } m_1 = m_2 \\ \pi\int_0^{r_{\max}}\partial_r\psi_r^{l_1}\partial_r\psi_r^{l_2} r^2\;\mathrm{d}r & \text{for } |m_1 - m_2| = 1 \\ 0 & \text{else} \end{cases} \end{aligned} \tag{3.131}$$

$$\begin{aligned} s_2 = (n_1 n_2)\int_0^{r_{\max}}\int_0^{2\pi}\mathrm{e}^{-\mathrm{i}(m_1-m_2)\theta}\frac{r}{R_0 + r\cos(\theta)}\psi_r^{l_1}\psi_r^{l_2}\mathrm{d}r\mathrm{d}\theta \\ = (n_1 n_2)\int_0^{r_{\max}}\int_0^{2\pi}\frac{\mathrm{e}^{-\mathrm{i}(m_1-m_2)\theta}}{\frac{R_0}{r} + \cos(\theta)}\psi_r^{l_1}\psi_r^{l_2}\mathrm{d}r\mathrm{d}\theta \end{aligned} \tag{3.132}$$

$$\begin{aligned} s_3 = (m_1 m_2)\int_0^{r_{\max}}\int_0^{2\pi}\mathrm{e}^{-\mathrm{i}(m_1-m_2)\theta}\left(\frac{R_0}{r} + \cos(\theta)\right)\psi_r^{l_1}\psi_r^{l_2}\mathrm{d}r\mathrm{d}\theta = \\ (m_1 m_2)\begin{cases} 2\pi R_0\int_{\Omega_r}\psi_r^{l_1}\psi_r^{l_2}\frac{1}{r}\;\mathrm{d}r & \text{for } m_1 = m_2 \\ \pi\int_{\Omega_r}\psi_r^{l_1}\psi_r^{l_2}\;\mathrm{d}r & \text{for } |m_1 - m_2| = 1 \\ 0 & \text{else} \end{cases} \end{aligned} \tag{3.133}$$

Collecting terms yields

$$\mathcal{K}_{(l_1,m_1,n_1),(l_2,m_2,n_2)} = 2\pi\delta_{n_1,n_2}\left(s_1 + s_2(n_1,n_2) + s_3(m_1,m_2)\right), \tag{3.134}$$

where the toroidal mode dependence can be explicitly extracted by defining

$$n_1 n_2 \tilde{s}_2 = s_2(n_1, n_2), \qquad m_1 m_2 \tilde{s}_3 = s_2(m_1, m_2), \tag{3.135}$$

which finally results in

$$\mathcal{K}_{(l_1,m_1,n_1),(l_2,m_2,n_2)} := 2\pi\delta_{n_1,n_2}\Big[s_1(l_1,l_2,m_1,m_2) + n_1^2\tilde{s}_2(l_1,l_2,m_1,m_2) + m_1 m_2 \tilde{s}_3(l_1,l_2,m_1,m_2)\Big]. \tag{3.136}$$

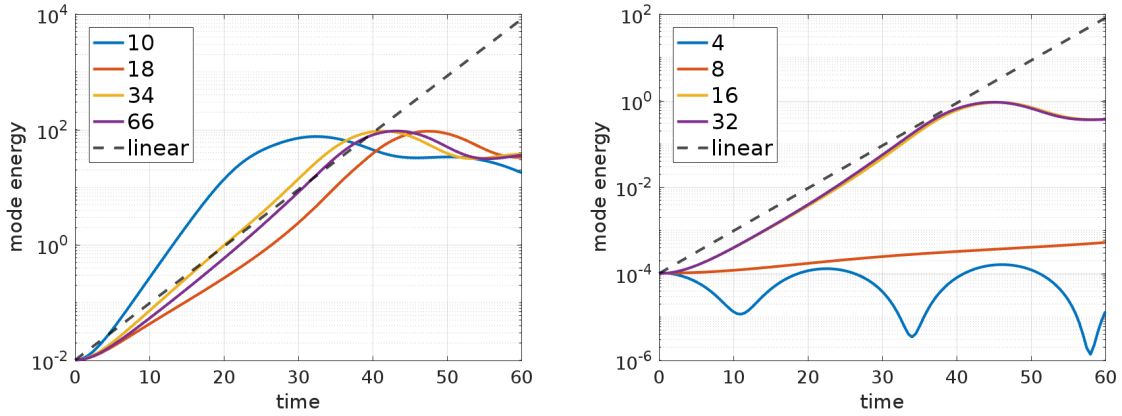
Although the B-splines provide sparsity in the radial direction, $\mathcal{K}$ in eqn. (3.136) exhibits dense blocks for the poloidal Fourier modes, such that a solver for sparse matrices with many entries is needed. The source of the problem is the $J_T^{-1}J_T^{-\dagger}$ tensor, which also appears in the transformed Maxwell's equation. This implies that, although the global Poisson solve can be circumvented in Vlasov–Maxwell, the Fourier modes will still couple in the standard geometry.
If the coordinate transformation is given in a Fourier-spline basis the corresponding matrices can be assembled algebraically. Nevertheless, the most straightforward approach is to use numerical quadrature, where in radial direction Gauss–Legendre points have to be used in each cell. Note that the fast Fourier transform should be used in order to speed up the initialization.

3.5.2. Diocotron instability with B-splines and Bessel functions

We previously already encountered the guiding center model in polar geometry (see also appendix B.1.3) and the Diocotron instability with the initial condition

$$r^- = 4,\ r^+ = 5, r_{\max} = 10, \epsilon = 10^{-2}, \gamma = -1, N_p = 10^5, \Delta t = 0.1$$
$$\rho(t=0,r,\theta) = \begin{cases} 1 + \epsilon\cos(l\theta) & \text{for } r^- \le r \le r^+ \\ 0 & \text{else.} \end{cases} \tag{3.137}$$

In this special geometry the eigenmodes of the Laplace operator are known as Fourier-Bessel functions. Thus, two different field solvers are considered. The obvious choice is particle in Fourier in θ-direction. For the radial direction finite elements based on cubic B-splines or Bessel functions are used. Here the problem size is so small that in both cases the field solver is neglectable, although it is trivial for the Bessel functions. In the current MATLAB implementation the Bessel functions itself are about 50-times more expensive than the Fourier modes prohibiting larger runs. A particular downside of global spectral methods becomes clear from fig. 3.27. The sharp annulus has to be resolved correctly and, therefore, a certain radial resolution is needed, such that the Bessel functions (fig. 3.27b) have no advantage over the cubic B-splines (fig. 3.27a). Nevertheless, the Fourier approximation in θ seems to be very effective for this problem, since the linear phase is correctly obtained, see fig. 3.28



(a) Fourier-FEM (B-splines) (b) Fourier-Hankel (Bessel)

Figure 3.27.: Growth rates for different number of degrees of freedom for the Diocotron instability compared to linear theory ($l = 3$). Because of the sharp annulus in the initial condition, a certain resolution is required for the correct linear phase, such that radial filtering with Bessel functions (b) poses no advantage over the unfiltered Fourier-FEM discretization (a).

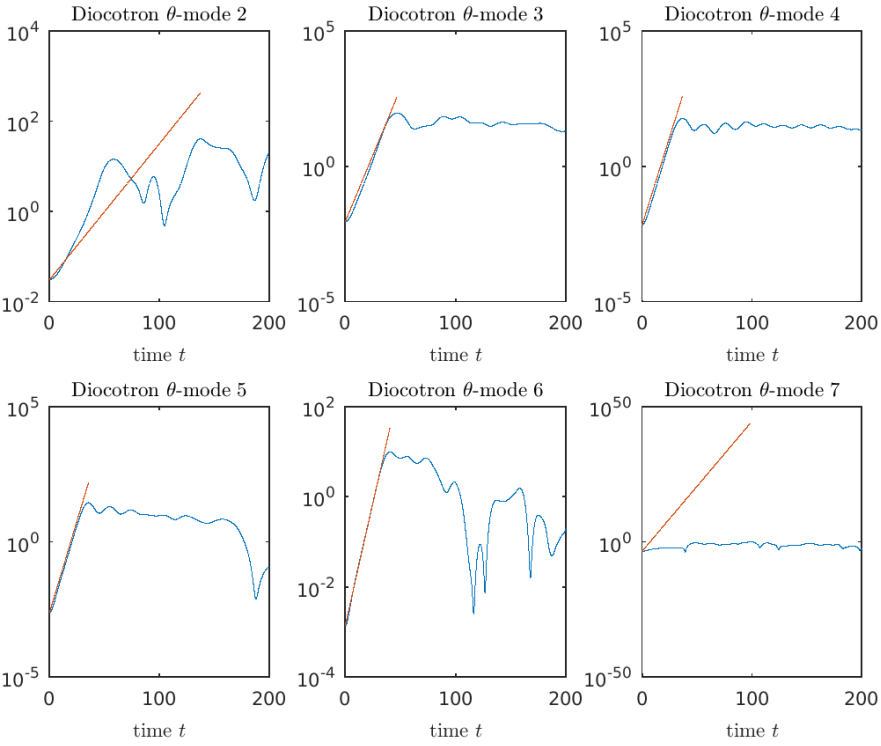


Figure 3.28.: Growth rates for the Diocotron instability for $N_p = 10^5$ particles. The highest modes are difficult to obtain, because of the particle noise.

3.5.3. Drift kinetic ion temperature gradient instability

The ion temperature gradient instability is a popular instability that emerges quite natural. An ion temperature gradient provokes an instability and ultimately leads to turbulence that slowly eats up this gradient. Physicists are interested in the ITG instability because it may help predicting parts of the turbulent transport inside fusion devices, which is important because you would like to keep the gradient and with it your confinement. For us it shall be just another accumulation of parameters and an initial condition to use in the reduced gyrokinetic model. The purpose of these tests is to demonstrate that PIF allows us to carry out single mode simulations in any curvilinear coordinate system on a desktop computer.
Here the spatial coordinates $r \in [r_{\min}, r_{\max}],\ \theta \in [0, 2\pi],\ \varphi \in [0, L_\varphi]$ are used in the plain box, also called slab. In the slab we denote radius, poloidal angle and toroidal angle. The same goes for cylinder and variants of the torus including a helical device with $s = 5$. Although we could easily consider inhomogeneous magnetic fields, we are only interested in the flexibility of the coordinate transformation and we try to keep this as comprehensive as possible. Technically a curvature in the magnetic field can be incorporated in the coordinate transformation to be field aligned. The only thing that one might want to change is the magnitude of B varying over the radius r. We consider an ITG test case [190], where we excite an eigenvalue of the linearized version of eqn. (B.61). This test-case simulates only one species, the ions $s = i$ such that we denote $f_s = f$. Nevertheless, there are radial profiles given for the ion temperature $T_i(r)$ the electron temperature $T_e(r)$ and the number density of all species $n_0(r)$ required. These profiles centered at r_p have always the same shape $\mathcal{P}(r)$ depending on different parameters. The constants $C_\mathcal{P}$ is only defined in order to normalize the density n_0 to one.

$$\begin{aligned}
\mathcal{P} &\in \{T_i, T_e, n_0\} \\
\mathcal{P}(r) &= C_\mathcal{P} \exp\left(-\kappa_\mathcal{P} \delta r_\mathcal{P} \tanh\left(\frac{r - r_p}{\delta r_\mathcal{P}}\right)\right) \\
C_{T_i} &= C_{T_e} = 1 \\
C_{n_0} &= \frac{r_{\max} - r_{\min}}{\int_{r_{\min}}^{r_{\max}} \exp\left(-\kappa_\mathcal{P} \delta r_\mathcal{P} \tanh\left(\frac{r - r_p}{\delta r_\mathcal{P}}\right)\right)} \\
r_p &:= \frac{r_{\max} - r_{\min}}{2}
\end{aligned} \tag{3.138}$$

These profiles are then used to define an equilibrium for the initial condition, that we will also used as control variate.

$$f_{eq}(r, v) = \frac{n_0(r)}{\sqrt{2\pi T_i(r)}} \exp\left(-\frac{v^2}{2T_i(r)}\right) \tag{3.139}$$

The initial condition itself is then a perturbation of the equilibrium by a Gaussian over the radial direction. The radial profile of the resulting mode does not correspond to a Gaussian and, therefore, the initial state is just an approximation to the true eigenvalue.

$$f(r, \theta, \varphi, v, t = 0) = f_{eq}(r, v) \left[1 + \epsilon \cdot \exp\left(-\frac{-(r - \frac{r_{\max} - r_{\min}}{2})^2}{\delta r}\right) cos\left(\frac{2\pi n}{L}\varphi + m\theta\right)\right] \tag{3.140}$$

$$\begin{aligned}
r_{\min} = 0.1,\ r_{\max} = 14.5,\ L_\theta = 2\pi,\ L_\varphi = 15606.759067, \\
\kappa_{n_0} = 0.055,\ \kappa_{T_i} = \kappa_{T_e} = 0.27586, \\
\delta r = 8,\ \delta r_{T_i} = \delta r_{T_i} = 1.45,\ \delta r_{n_0} = 2\delta r_{T_i}, \\
n = 1,\ m = 5
\end{aligned} \tag{3.141}$$

For the ITG test case we have to draw a velocity distribution with radial dependent temperature

$$g(r,\theta) = \frac{r}{\left(r_{\max}^2 - r_{\min}^2\right)\pi} 1_{r_{\min}\leq r\leq r_{\max}}. \tag{3.142}$$

With the substitution $\tilde{v} := \frac{v}{\sqrt{(T_i(r))}}$ we can substitute the integral

$$\iint \frac{1}{\sqrt{2\pi T_i(r)}} \mathrm{e}^{-\frac{v^2}{2T_i(r)}}\ r\ drdv = \iint \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{\tilde{v}^2}{2}}\ r\ dr d\tilde{v} \tag{3.143}$$

and therefore draw $\tilde{v}_k \sim \mathcal{N}(0,1)$ and set $v_k = \sqrt{T_i(r_k)}\tilde{v}_k$. This is independent of the sampling in r.

Quasi-neutrality test-case

The general elliptic solver is tested by use of a manufactured solution to the quasi-neutrality equation in cylindrical coordinates (3.144). By performing such tests, bugs and bottlenecks can be found in the implementation such that

$$\begin{aligned} -\left[n_0(r)\partial_{rr}\Phi(r,\theta,\varphi) + \left(\frac{n_0(r)}{r} + \partial_r n_0(r)\right)\partial_r\Phi(r,\theta,\varphi)\ +\ \frac{n_0(r)}{r^2}\partial_{\theta\theta}\Phi(r,\theta,\varphi)\right] \\ + \frac{n_0(r)}{T_e(r)}\left(\Phi(r,\theta,\varphi) - \bar{\Phi}(r,\theta)\right) = \rho(r,\theta,\varphi) \\ \bar{\Phi}(r,\theta) = \frac{1}{L_\varphi}\int_0^{L_\varphi} \Phi(r,\theta,\varphi)\mathrm{d}\varphi. \end{aligned} \tag{3.144}$$

Inserting the potential

$$\Phi(r,\theta,\varphi) = \sin\left(2\cdot 2\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right)\left[\cos(\varphi+2\theta) + \sin\left(2\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right)\cos(2\theta)\right] \tag{3.145}$$

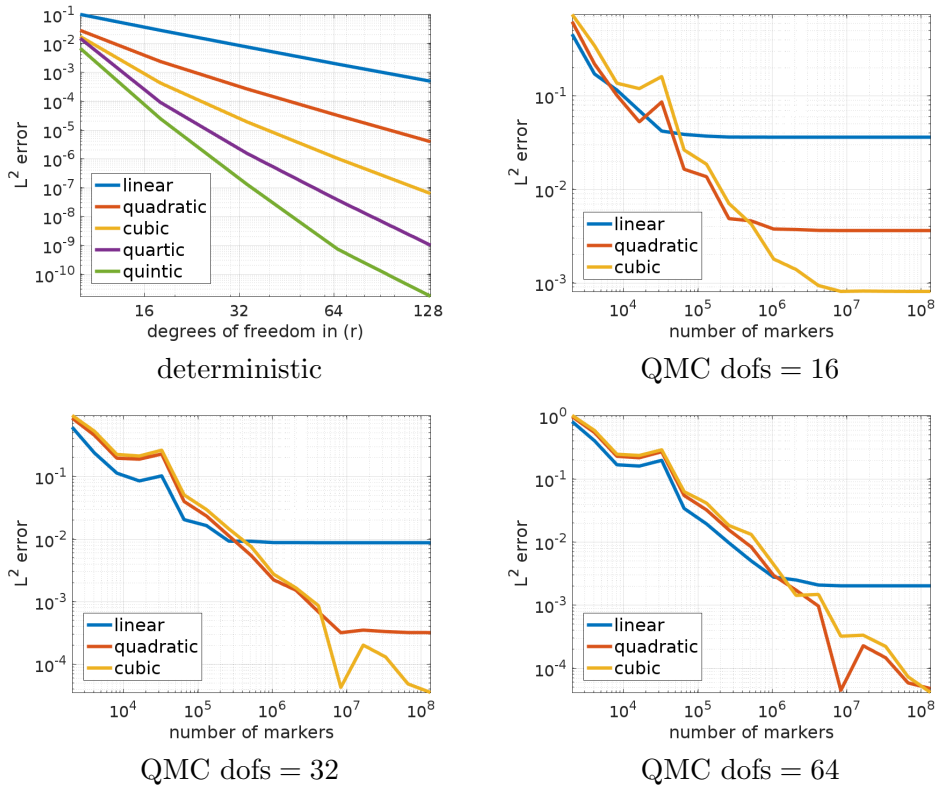
into eqn. (3.144) yields

$$\begin{aligned} \rho(r,\theta,\varphi) = \frac{n_0(r)}{r^2}\left[4\cos(\varphi+2\theta)\sin\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right) + 4\cos(2\theta)\sin\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right)\right] \\ + \frac{4\pi^2 n_0(r)}{(r_{\max}-r_{\min})^2}\left[4\cos(\varphi+2\theta)\sin\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right) + \cos(2\theta)\sin\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right)\right] \\ + \frac{n_0(r)}{L_\varphi T_e(r)}\left\{\sin\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right)\left[\sin(2\theta) - \sin(L_\varphi+2\theta) + L_\varphi\cos(\varphi+2\theta)\right]\right\} \\ -\frac{2\pi(rn_0'(r)+n_0(r))}{r*(r_{\max}-r_{\min})}\left[2\cos(\varphi+2\theta)\cos\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right) + \cos(2\theta)\cos\left(4\pi\frac{r-r_{\min}}{r_{\max}-r_{\min}}\right)\right]. \end{aligned} \tag{3.146}$$

This tests the interplay between coordinate transformation and matrix assembly. Additionally the particle mesh coupling is tested by sampling particles uniformly in the cylinder according to

$$\begin{aligned} u\sim\mathcal{U}(0,1),\ r = \sqrt{u(r_{\max}^2 - r_{\min}^2) + r_{\min}^2},\ \theta\sim\mathcal{U}(0,2\pi), \\ \varphi\sim\mathcal{U}(0,L_\varphi),\ w = 2\pi L_\varphi\frac{r_{\max}^2 - r_{\min}^2}{2}. \end{aligned} \tag{3.147}$$

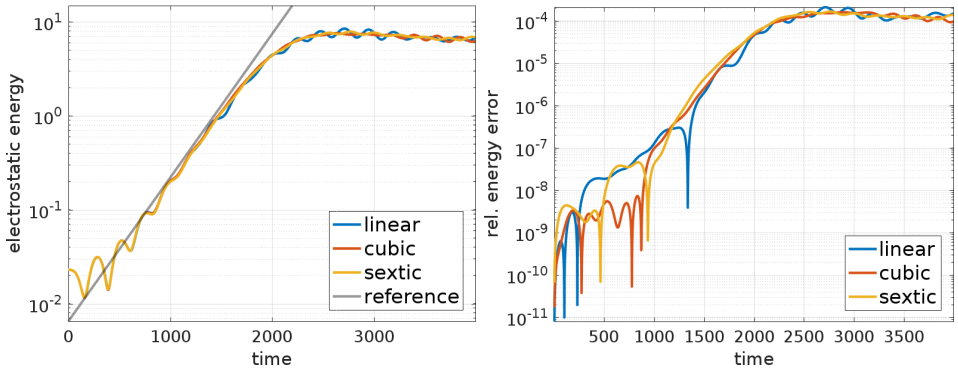
Using the L^2 projection of ρ and three Fourier modes in each dimension, the convergence for different B-spline degrees is checked. A correct implementation yields an increasing order



deterministic | QMC dofs = 16

QMC dofs = 32 | QMC dofs = 64

Figure 3.29.: Testing a general elliptic solver for the quasi-neutrality equation in cylindrical coordinates by means of a manufactured solution. L^2 error on the electric potential for a given ρ with $r_{\min} = 0, r_{\max} = 5$ and $L_\varphi = 2\pi$. For large particle numbers the variance drops below the discretization error for the low oder splines such that the bias appears.



(a) electrostatic energy (b) energy error

Figure 3.30.: Single mode drift kinetic ITG instability with varying spline degree and $N_r = 16$ and $N_p = 4 \cdot 10^5$. The electrostatic energy (a) follows the reference in the linear phase. It exhibits less oscillations for higher order splines, yet the energy error (b) does not change drastically.

of convergence with increasing spline degree. Additionally the variance and bias can be observed in the particle mesh coupling, especially the small increase in variance for radial cells, see fig. 3.29. From fig. 3.29 it also becomes clear that even for such a simple example an enormous number of markers is needed in order to decrease the variance to the level of the bias. Further tests can be done for other geometries, where differential operators for various geometries can be found in [191].

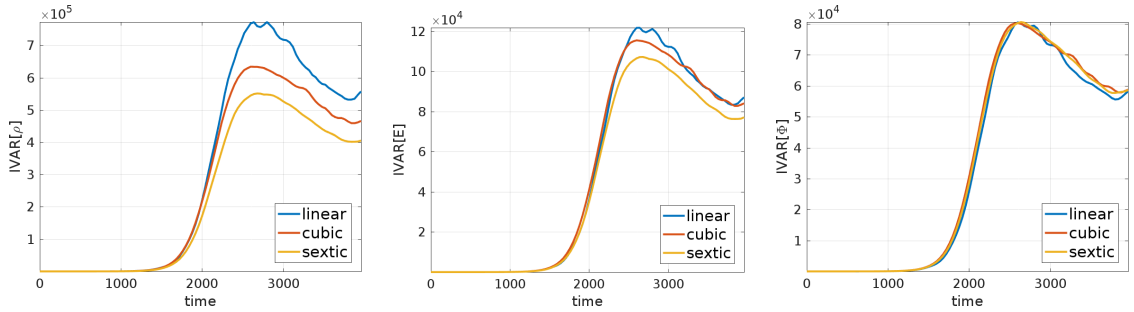
Spline degree

In the linear phase the noise obscures small perturbations, such that the δf method is highly effective. It is quite common that smooth particle shapes are seen as a form of noise reduction method, but they rather reduce the degrees of freedom. Nevertheless, increasing the spline degree increases the spectral fidelity such that we are interested in the effects for high order splines. The ITG test-case requires some basic resolution, thus, it was not possible to use less than $N_r = 16$ degrees of freedom (not number of cells) in radial direction and obtain reasonable growth rates, even for high order splines. The energy error, a quantity merely depending on the splitting error, does not change but the electrostatic exhibits less oscillations in the nonlinear phase, see fig. 3.30. This is explained by the slight variance reduction of the high order B-splines, see fig. 3.31. Although this cannot be generalized, it is obviously worthwhile investigating the variances with respect to the order and not only the resolution.

Adapted polar mesh

In curved geometries the measure of particle per cell becomes non-intuitive since the cells are deformed, while the uniform background density does not depend on the geometry as the Jacobian is always included. More concretely in polar geometry with Jacobian r the uniform sampling density $g(r) = r$ is used yielding the sampling in eqn. (3.147). It represents a constant Vlasov density very well. To allow for different types of grids in radial direction, a parameter $\alpha_r > 0$ is introduced. The knot sequence is then defined as

$$r_k := r_{\max} \left(\frac{k}{N} \right)^{\alpha_r}, \qquad k = 1, \dots, N_r. \tag{3.148}$$



IVAR $[\rho]$ IVAR $[E]$ IVAR $[\Phi]$

Figure 3.31.: Integrated variance of the charge density ρ, electric field E and potential Φ. Increasing the B-spline degree from linear to sextic yields a slight variance reduction by smoothing away the smalls scales. Since the Laplace operator damps the small scales anyhow there is no effect on the potential Φ.

The cell number for a position r is then given as

$$\text{cell}(r) = \left\lfloor \left(\frac{r}{r_{\max}} \right)^{1/\alpha_r} N \right\rfloor . \tag{3.149}$$

In polar coordinates the area of the $k-th$ cell is

$$\begin{aligned} A_k = \left(r_k^2 - r_{k-1}^2\right)\pi &= \left[\left(r_{\max} \left(\frac{k}{N} \right)^{\alpha_r} \right)^2 - \left(r_{\max} \left(\frac{k-1}{N} \right)^{\alpha_r} \right)^2 \right] \pi \\ &= \pi \frac{r_{\max}^2}{N^{2\alpha_r}} \left(k^{2\alpha_r} - (k-1)^{2\alpha_r} \right) . \end{aligned} \tag{3.150}$$

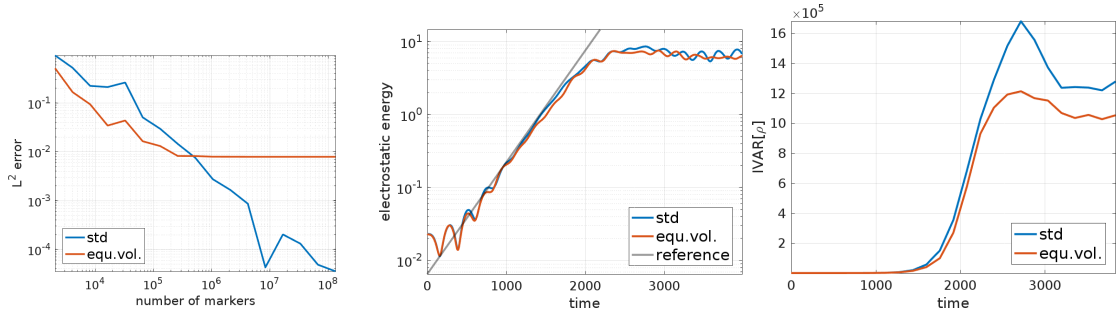
For $\alpha_r = 1$ equidistant spacing is obtained whereas $\alpha_r = \frac{1}{2}$ results in a partition of equal areas of the unit circle.

$$\alpha_r = \frac{1}{2} \Rightarrow A_k = \pi \frac{R_{\max}^2}{N^{2\alpha_r}} = const. \tag{3.151}$$

The canonical choice is to take an equidistant sequence, that means $r_k - r_{k-1} = const.$, since this yields the smallest discretization error (except for exotic cases involving the singularity). But in the particle-mesh coupling, the variance is also important such that such common knowledge might not apply. Indeed for the quasi-neutral test-case a reduction of error can be achieved by changing the grid to $\alpha_r = 1/2$, see fig. 3.32a. This also translates to the ITG test-case, see figs. 3.32b and 3.32c. It has to be mentioned that by the worse discretization error the eigenvalues of the field equations are also approximated not as well. This entirely depends on the problem and has unpredictable consequences for nonlinear simulations.

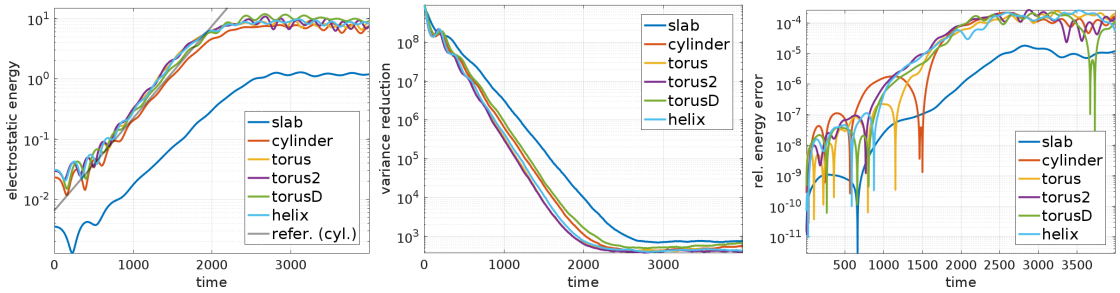
Multiple geometries

We run the same test-case with exactly the same parameters, with the only difference that the coordinate transformation is changed resulting in a different curved domain. Note that for an accurate physical representation, a different initial conditioned tailored to the domain has to be chosen. Mostly these equilibria are obtained by solving the high collisional limit to the original Vlasov equation, which essentially leads to MHD equilibria. These are then again parameterized yielding test-cases. Such extensions can be found for the torus as the DIII-D test-case in [192] and the D-shaped cylinder, see [193]. The results for quartic splines with $N_p = 10^4,\ N_r = 16,\ \Delta t = 5$ can be seen in figs. 3.33, 3.34 and 3.35.



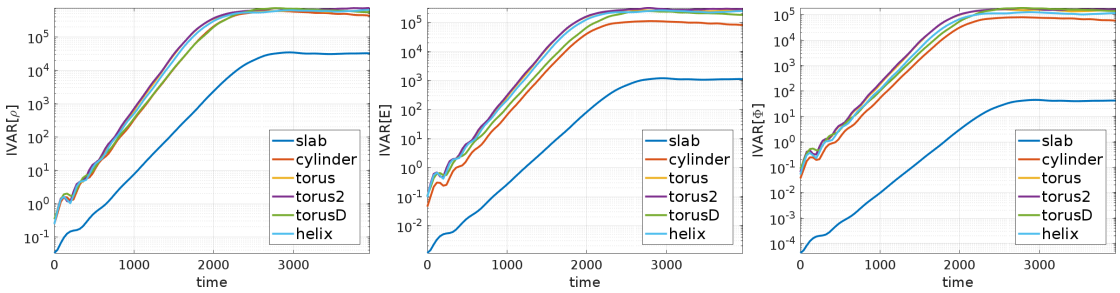
(a) quasi-neutrality test-case (cubic) (b) energy error (c) integrated variance of ρ

Figure 3.32.: In polar geometry the number of particles per cell can be kept constant by changing the radial grid spacing to an equi-volume map. This reduces the variance in the standard quasi-neutral test-case (a), but increases the bias compared to the standard equidistant grid (cubic, $N_r = 32$). Thus, for noise dominated simulations it is feasible to adapt the mesh and eventually increase the spline degree in order to compensate for the extreme increase in bias. For the single mode drift kinetic ITG instability (b) with varying spline degree a small variance reduction is observed (c).



(a) electrostatic energy (b) variance reduction (c) relative energy error

Figure 3.33.: Single mode ITG simulation for different domains. For the cylindrical model the linear phase matches the analytical prediction (a) and also does not change much for toroidal geometries due to the large aspect ratio. The control variate is very effective during the linear phase (b). Compared to nonlinear Landau damping the factor variance reduction > 100 is much better, such that it is still worth using the control variate in this nonlinear phase. The energy error increases with the complexity of the geometry (c).



(a) IVAR $[\hat{\rho}]$ (b) IVAR $\left[\hat{E}\right]$ (c) IVAR $\left[\hat{\Phi}\right]$

Figure 3.34.: Integrated variances of charge density ρ, electric field E and potential Φ over the linear phase of an ITG instability for different geometries.

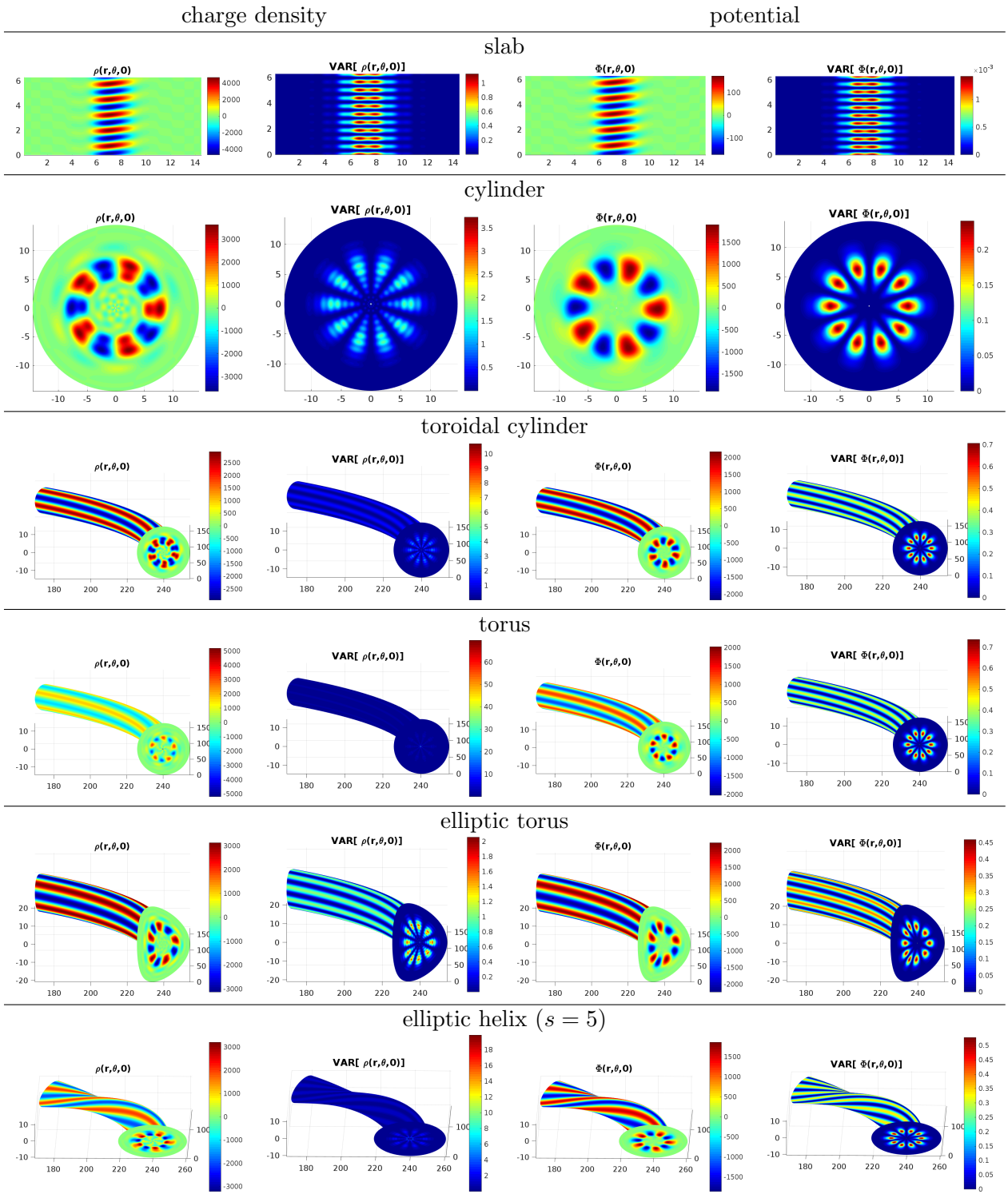


Figure 3.35.: Charge density and potential for a single mode drift kinetic ITG in the nonlinear phase at the end of the simulation. The potential Φ is by help of the Laplacian much smoother than the charge density ρ. Also the extent of the variance in the radial direction gets limited by this damping.

3.6. Implementation and benchmarks of Particle-In-Fourier

When the Particle-In-Fourier discretization of a system is already given, the implementation is mostly straightforward. For spectral methods in general a factor two can be gained easily by using the complex symmetry of the Fourier modes, see appendix C.2. Yet there are other formulas that result in a performance gain which can be easily applied a-posteriori.

3.6.1. Numerical evaluation of the Fourier modes

The orthogonal basis functions sin and cos have many advantages, but even on modern computer hardware they are expensive to evaluate. Since the Particle-In-Fourier heavily relies on the massive evaluation of trigonometric functions, we present different options. Although everything started with the CORDIC (coordinate rotation digital computer) algorithm, where Volder [194] gives a nice explanation, these algorithms are designed for limited hardware, like computers from the 50s or micro-controllers nowadays. Today we rely on software implementations provided by different libraries or direct hardware implementation on modern architectures such as Intel Skylake. In the beginning we seek for double precision accuracy of the method, although in almost every case the particle noise is more dominant. By trading off accuracy the conserved quantities of the geometric integration are affected at first. The solution quality does then depend on the particle noise. Since this can be done at compile time by enabling fast math option *(-ffast-math)*, it is not of our concern.
The first step in accelerating a function evaluation is by defining a lookup table up to a certain precision, or use some form of polynomial interpolation. Evaluating polynomials is thanks to Horners algorithm computationally accurate and cheap. Another option often used is the approximation by a Taylor series. By range reduction we restrict ourselves to fast evaluation of $\sin(x)$ for $x \in [0, \frac{\pi}{2}]$ by polynomials of order n. The corresponding Taylor series is obtained by truncating the expansion to a n^{th} order polynomial.

$$f(x) := \sin(x) \approx \sum_{m=0}^{\infty} (-1)^m \frac{x^{2m+1}}{(2m+1)!} \tag{3.152}$$

An enhanced lookup table uses cubic spline interpolation on $n+1$ equidistant points in $[0, \frac{\pi}{2}]$. On the same equidistant points standard Lagrange interpolation yields an unique polynomial of degree $n+1$. But the best way is to use Chebyshev polynomials, which we map from $[-1, 1]$ to $[0, \frac{\pi}{2}]$. We recall some useful equations for Chebyshev interpolation from [195]. The roots of the $(n+1)^{\text{th}}$ degree Chebyshev polynomial of first kind T_{n+1} read

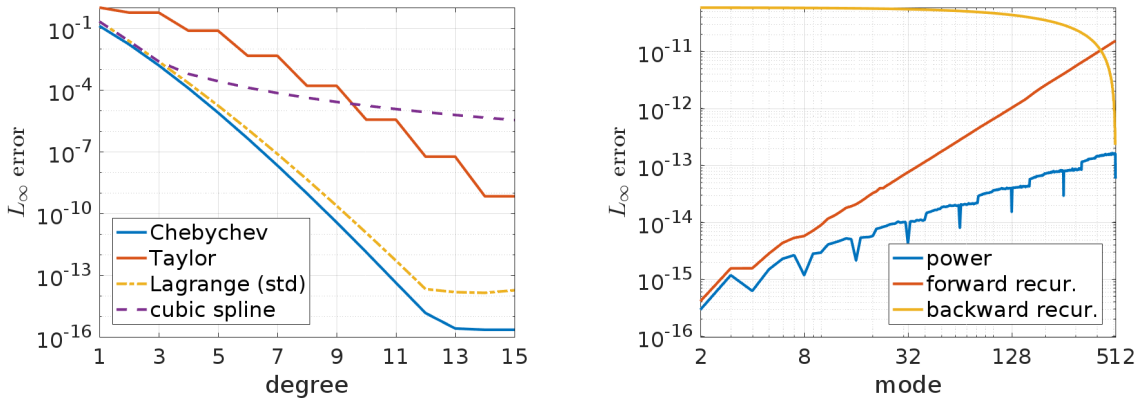
$$T_{n+1}(x_j) = 0, \quad x_j := \cos\left(\frac{j - \frac{1}{2}}{n+1}\pi\right), \quad \forall j = 1, \dots, n+1. \tag{3.153}$$

Chebyshev interpolation is done by mapping these roots to $\tilde{x}_j = \frac{x_j+1}{2}\frac{\pi}{2}, \quad \forall\, j = 1, \dots, n+1$ and interpolating with an n^{th} order polynomial through the points $(\tilde{x}_j, x_j)$. This polynomial can also be obtained by a Chebyshev sum

$$f(x) \approx \sum_{i=0}^{n} c_i T_i(x), \tag{3.154}$$

with coefficients

$$c_i = \frac{2}{n+1} \sum_{k=1}^{n+1} f(x_j) T_i(x_j), \quad i = 0, \dots, n. \tag{3.155}$$



(a) Polynomial approximation of sin in $[0, \frac{\pi}{2}]$ (b) Multiple Fourier modes by recursion

Figure 3.36.: Since polynomials is very fast expensive trigonometric functions such as sin can be approximated by a polynomial expansion up to a certain degree (a). For cubic splines the degree denotes the degrees of freedom for an underlying grid. For a 13^{th} degree Chebyshev series, the sin is approximated to machine precision, demonstrating that spectral expansion outperforms naive interpolation. Based on the first Fourier mode higher modes can be obtained by forward or backward application of the Chebyshev identity (3.157) or a plain exponential power by successive multiplication according to eqn. (3.156), which affects the precision (b).

	GPU		**CPU**	
	time [s]	error	time [s]	error
native	1.08e-09	6.1e-05	2.74e-09	6.82e-09
standard	1.11e-09	3.33e-16	3.45e-09	3.33e-16

Table 3.2.: The native implementation of *sin* in OpenCL is slightly faster yet inaccurate. Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz with Intel(R) HD Graphics, Intel OpenCL

Here fig. 3.36a identifies the Chebyshev polynomials clearly as the most efficient method tightly followed by the standard Lagrange interpolation. The cubic splines are only useful up to three degrees of freedom, so either we use a lookup table to the desired precision, or directly take the global polynomials. But for PIF we need to evaluate the exponential function of a purely complex argument. In many cases a function *sincos* is available providing sin and cos simultaneously.

$$e^{\mathrm{i}kx} = \cos(kx) + \mathrm{i}\sin(x) = \left(e^{\mathrm{i}x}\right)^k = (\cos(kx) + \mathrm{i}\sin(x))^k \tag{3.156}$$

Many modes have to be evaluated, thus, the costly trigonometric functions calls can be reduced to a single evaluation followed by many complex self multiplications. This is without doubt very cheap, but leads to numerical roundoff error as can be seen in fig. 3.36b. For compensation of numerical roundoff, see [196].
Another disadvantage is that this cannot be vectorized, hence in a highly vectorized environment like OpenCL it can be faster to directly call *sincos* for every mode. Although it is, of course, possible to vectorize the recurrence relation in eqn. (3.156) by hard-coded unrolling. In computer experiments an additional factor of two could be gained by unrolling at least 8 iterations for $N_f \geq 64$. On Intel Skylake vectorization is said to gain a factor of up to ten, which we were unable to achieve here by loop unrolling. Since such unrolling techniques restrict the codes flexibility they were not used for general tests. We recall the Chebyshev

identity, which is another formula using a two term recurrence for the sine and cosine

$$\begin{aligned} \cos(nx) &= 2\cos(x)\cos\left((n-1)x\right) - \cos\left((n-2)x\right), \\ \sin(nx) &= 2\cos(x)\sin\left((n-1)x\right) - \sin\left((n-2)x\right). \end{aligned} \tag{3.157}$$

It can be used forward starting from the zeroth mode, or backward. Yet fig. 3.36b shows that both methods suffer from heavy roundoff error. Sometimes hardware implementations are available, which can be easily accessed by OpenCL but lack the desired precision in our case, see table 3.2.

3.6.2. Micro-benchmark

Particle-In-Fourier is simple to implement, such that we can test different programming languages and hardware. For the standard Particle-In-Cell a vast variety of skeleton codes are provided by Decyk, see [197],[198], which are used as a basis for comparison. For the following tests we chose a standard Vlasov–Poisson simulation with the third order symplectic Runge Kutta as time integrator, which yields few lines of code for Landau damping. In contrast to PIC, where particles only contribute to their surrounding cells using cheap polynomials, in PIF every particle contributes to every Fourier mode via an expensive e−function. This makes PIF computationally heavier such that we expect it to have better performance in massively parallel environments like GPUs, where locally much more FLOP/s are possible. Dedicated GPU programming is cumbersome such that we search for a high level framework that allows for an easy implementation. MATLAB offers *gpuArrays* with Nvidia CUDA as back-end such that porting existing code to the GPU is trivial.
In the first example we use our most efficient MATLAB implementation of the particle mesh coupling using cubic B-splines finite elements for a Vlasov–Poisson PIC code. Then exactly the same problem is solved, once with PIC and PIF. Note that in PIC for one Fourier mode two cells are needed. Fig. 3.38 indicates that at least in MATLAB, PIF is more efficient on the GPU. Yet this takes place at such a high level of abstraction that it cannot be generalized. Therefore, we perform a micro-benchmark in order to test implementations of the same PIF algorithm in different languages. With a syntax similar to MATLAB yet performance of C we present implementations in the new language julia [199], in order to leave the old-age standard Fortran based high performance plasma-physics behind us. We, of course, include python with numpy [200]. OpenCL is a portable framework for high performance applications based on C, that allows us to use the same code on CPU and GPU. Via python the pyOpenCL package provides a simplified interface [201]. The advantage over CUDA is that we do not need expensive hardware but can use a Laptop with an Sky Lake Intel(R) Core(TM) i5-6300U CPU and the integrated Intel HD Graphics 520 GPU supporting double precision. If we seek a comparison between fortran, julia and python, we have to use the same compiler flags and of course the same compiler, and therefore, we choose gfortran with the highest optimization flag $-O3$. For the single core comparison we set the environment variable `OMP_NUM_THREADS=1`, in order to keep MATLAB and numpy from using several threads. A dramatic improvement is the architecture specific optimization by *-march=native*. For the Fortran example this is a trivial change, yet for julia the entire interpreter had to be rebuild. Since julia was installed from source, this was done by adding a file "Make.user" with the entry *MARCH=native*, `JULIA_CPU_TARGET=native` and then building as recommended. In the optimized variant of the algorithm the Fourier modes are calculated by successive multiplication according to eqn. (3.156). Yet this method cannot be vectorized without explicit unrolling such that it is not feasible with MATLAB and numpy. If not specified otherwise $N_p = 10^5$ particles, $N_f = 64$ Fourier modes and $N_t = 150$ time steps are used. The results for different languages can be

Interpreter	time [s]		
	standard		optimized
	single	multi-thread(4)	
gFortran (-O3, OMP)	403.37	134.01	7.67
julia (-O3, MPI)	361.73	-	8.90
python (numpy)	562.40	-	-
MATLAB (CPU)	237.43	122.15	-
pyOpenCL (CPU)	-	12.08	(8.91)
julia Yeppp!	209.35	83.03	-
pyOpenCL (GPU)	-	14.40	(10.81)
PIC (OMP)	1.06	0.29	-

Table 3.3.: Wall time for PIF with $N_p = 1e5$ particles, $N_f = 64$ Fourier modes and $N_t = 150$ third order time steps. For pyOpenCL the charge assignment could not be optimized. Decyks OpenMP one dimensional single precision sekeleton PIC code *fmpic1* is used as a reference. The number of cells is set to $N_f = 2 \cdot 64$ and $N_t = 3 \cdot 150$ in order to account for the difference in the leap frog scheme. The performance of PIC is insensitive to the number of cells N_f and the dominating costs are charge assignment such that the comparison is fair.

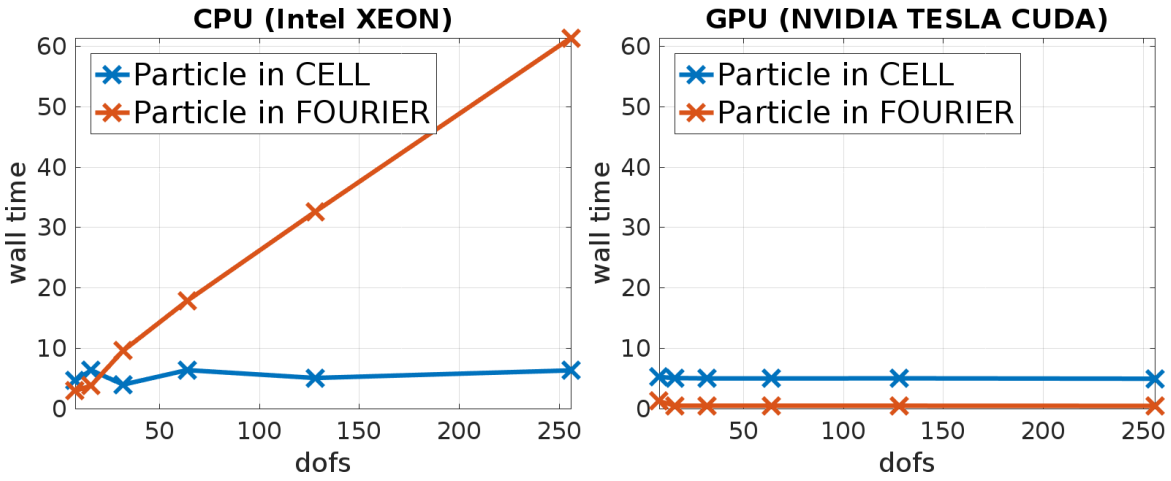
seen in table 3.3. MATLAB performs surprisingly well for the standard algorithm. For the standard PIF OpenCL outperforms the OpenMP Fortran by at least an order of magnitude, see also fig. 3.39a. Yet for the optimized variant the computational costs per particle are so low that a different kernel design is needed in order to implement the optimized charge projection. This is not done here and only the charge assignment is optimized. Figure. 3.39c shows that for GPU und CPU the optimized charge projection is much more efficient than the standard charge assignment causing an imbalance in costs, such that given the right design greater performance benefits are possible. Nevertheless for larger problem sizes the - not yet perfect - pyOpenCL still outperforms Fortran on the CPU, see fig. 3.39b. The newcomer julia is even slightly faster than the best Fortran implementation, but it uses MPI and not OpenMP such that we suspect the gain in performance to come from the overhead. Using the julia interface to the vectorized library *Yeppp!* [202] results in a speedup of two for the trigonometric functions but relies on vectorization. Scanning the number of Fourier modes reveals the overhead of *julia* and *python*, see fig. 3.39. Nevertheless *pyOpenCL* appears to get more work done in the same time, such that we have to ask ourselve the question how well the Fortran code is actually optimized. Under Linux the tool *perf* allows us to measure the percentage of cache misses and instructions per cycle, the results for PIF and Decyk's *mpic1* are summarized in fig. 3.37. It becomes clear that the hardware is not used to full extent in neither of the cases. Therefore, we recommend OpenCL kernels coupled to a high level language like python or julia since the development was fairly simple and there is much room for optimization in the kernels, as already discussed before.

```
gfortran -O3 -mtune=native -march=native -fopenmp \
         -ggdb -g -o pif_vp_OMP.fortran pif_vp_OMP.F90
export OMP_NUM_THREADS=4
perf stat -B -e cache-references,cache-misses,cycles,\
  instructions,branches,faults,migrations ./pif_vp_OMP.fortran
Performance counter stats for './pif_vp_OMP.fortran':

       378.289.955      cache-references
         6.217.327      cache-misses              #    1,644 % of all cache refs
    88.515.150.902      cycles
   109.372.024.305      instructions              #    1,24  insns per cycle
     8.118.313.560      branches
               709      faults
                 3      migrations

       7,673797924 seconds time elapsed
```

```
Performance counter stats for './fmpic1':

       213.051.927      cache-references
           400.951      cache-misses              #    0,188 % of all cache refs
    32.628.470.342      cycles
    26.403.571.347      instructions              #    0,81  insns per cycle
     2.291.879.005      branches
               528      faults
                 3      migrations

       2,841810770 seconds time elapsed
```

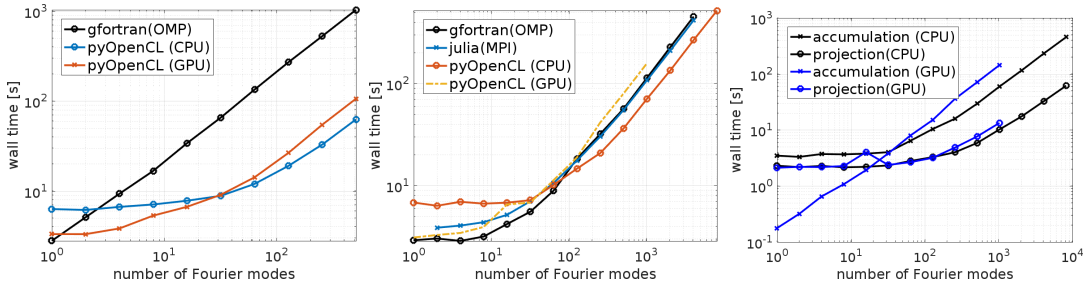
```
 Performance counter stats for 'python ./pif_vp_opencl.py':

       989.990.351      cache-references
       265.004.291      cache-misses              #   26,768 % of all cache refs
    70.462.157.878      cycles
    95.748.734.682      instructions              #    1,36  insns per cycle
     7.943.209.499      branches
           415.523      faults
                16      migrations

       8,910162779 seconds time elapsed
```

Figure 3.37.: Under Linux the tool *perf* provides a performance analysis in one simple step. Deeper analysis of the Fortran code by *perf annotate* revealed that the cache misses come from the trigonometric calls in *libm*. For the Intel Skylake architecture the theoretical maximum of instructions per cycle (IPC) is 16, such that there is still room for improvement by a factor ten. Despite the large amount of cache misses due to the flawed reduction kernel design the OpenCL code has definitely better trigonometric functions, since it does not use successive multiplication.



(a) CPU (b) GPU (with MATLAB's gpuArrays)

Figure 3.38.: Comparison of Particle-In-Cell and Particle-In-Fourier on CPU and GPU under MATLAB. On the GPU and a moderate problem size the computational costs are independent of the number of modes. The tests were performed on IPP's *draco* cluster with an Intel Xeon E5-2698 CPU and Nvidias PNY GF980GTX GPU.



(a) Standard PIF (b) Optimized PIF (c) PIF OpenCL kernels on CPU and GPU

Figure 3.39.: The computational costs of PIF increase linear with the number of Fourier modes, but the leading constant leaves room for improvements. The highly vectorized OpenCL beats fortran for the standard PIF (a), although exactly the same algorithm is used. When successive multiplication is applied (for pyOpenCL only in charge projection) OpenCL beats julia, which levels in with fortran (b). The optimized charge assignment using successive multiplications cannot be efficiently implemented for GPU using pyOpenCL templates (c). A possible solution are advanced reduce kernels specifically designed for many threads, but are more extensive to implement.

3.7. Eulerian versus Lagrangian in Fourier space

We have compared PIF and PIC, which are both Lagrangian methods subject to the same noise. Yet if there is no self-consistent field, there is additional noise on the particle dynamics. Eulerian methods converge faster in low dimensions, such that there is no noise but for advection, stabilization in the form of diffusion is required. Whenever a diffusion free transport is needed Lagrangian particles are mostly a better choice than the Eulerian ones. This is also one of the reasons why particle methods are so successful in e.g. high energy beam physics, where the particles run through many complicated magnetic fields. In the case of moderate degrees of freedom the error on the solution is not dominated by the order of convergence but the constant in front of it, which is for PIF the variance. If we can suppress the variance enough, then it should be possible to compete with an Eulerian solver for a moderate amount of particles.

3.7.1. Direct comparison

Since PIF is a spectral discretization, the Eulerian solver should also be spectral in spatial direction. The closest step is then to choose the standard pseudo-spectral solver. The spectral solver is equipped with a Fourier filter such that it approximates exactly the same system as the PIF. The time integrator is, for both solvers, the third order Runge Kutta [32] such that the discretization of the density f is the only difference. Note that this integrator is not symplectic but adjoint symplectic for the Eulerian discretization. The parameters for the Bump-on-tail instability and the grid of the Eulerian solver are given in eqns. (3.158).

$$\begin{aligned} \epsilon = 0.05,\ v_0 = 4.5,\ \sigma = 0.5,\ k = 0.3,\ m = 1,\ L = m \cdot \frac{2\pi}{k} \\ f(x,v,t=0) = \frac{1+\epsilon\cos(kx+\frac{\pi}{4})}{\sqrt{2\pi}}\left[(1-n_b)\mathrm{e}^{-\frac{v^2}{2}} + \frac{n_b}{\sigma}\mathrm{e}^{-\frac{(v-v_0)^2}{2\sigma^2}}\right] \\ g(x,v,t=0) = \frac{1}{\sqrt{2\pi}L}\left[(1-n_b)\mathrm{e}^{-\frac{v^2}{2}} + \frac{n_b}{\sigma}\mathrm{e}^{-\frac{(v-v_0)^2}{2\sigma^2}}\right] \\ v_{\max} = 9,\ v_{\min} = -7 \end{aligned} \tag{3.158}$$

Here, for the MATLAB implementation of both solvers the raw run time coincides up to 5%, when only $N_f = 1$ Fourier mode is used. Figure 3.40 shows that in that, case the PIF can absolutely compete with the pseudo-spectral solver despite the low dimension. This is only possible because Sobol's Quasi-Monte-Carlo numbers are used, which have a theoretical convergence order of $\mathcal{O}(\frac{1}{N})$ and are definitely not the first choice for two dimensional integration. We should also mention that the vortices in phase space emerging from the bump-on-tail instability, also called BGK modes, form actually a nonlinear stable state [203]. This means, that they are less susceptible to the particle noise. Thus small amplitude Landau damping is unfair for PIF and the BGK modes are eventually unfair for the Eulerian solver. Yet we have to admit that this test-case is specifically designed such that the noisy particles perform well. Since $m = 1$ the longest wavelength is also the most unstable one. By increasing $m = 3$ the third Fourier mode is excited, and if we do not filter the first two modes the PIF has no chance against the pseudo-spectral solver, see fig. 3.41. This behavior is also observed in a more comprehensive study comparing PIC with a Hermite-Fourier spectral solver for a turbulence test-case [75], where the PIC is too noisy in the small wavelengths. A study comparing a six dimensional Vlasov–Poisson PIF against a Semi-Lagrangian solver also draws the conclusion that PIF can be efficient for very few modes, tailored to the scenario [204]. Another

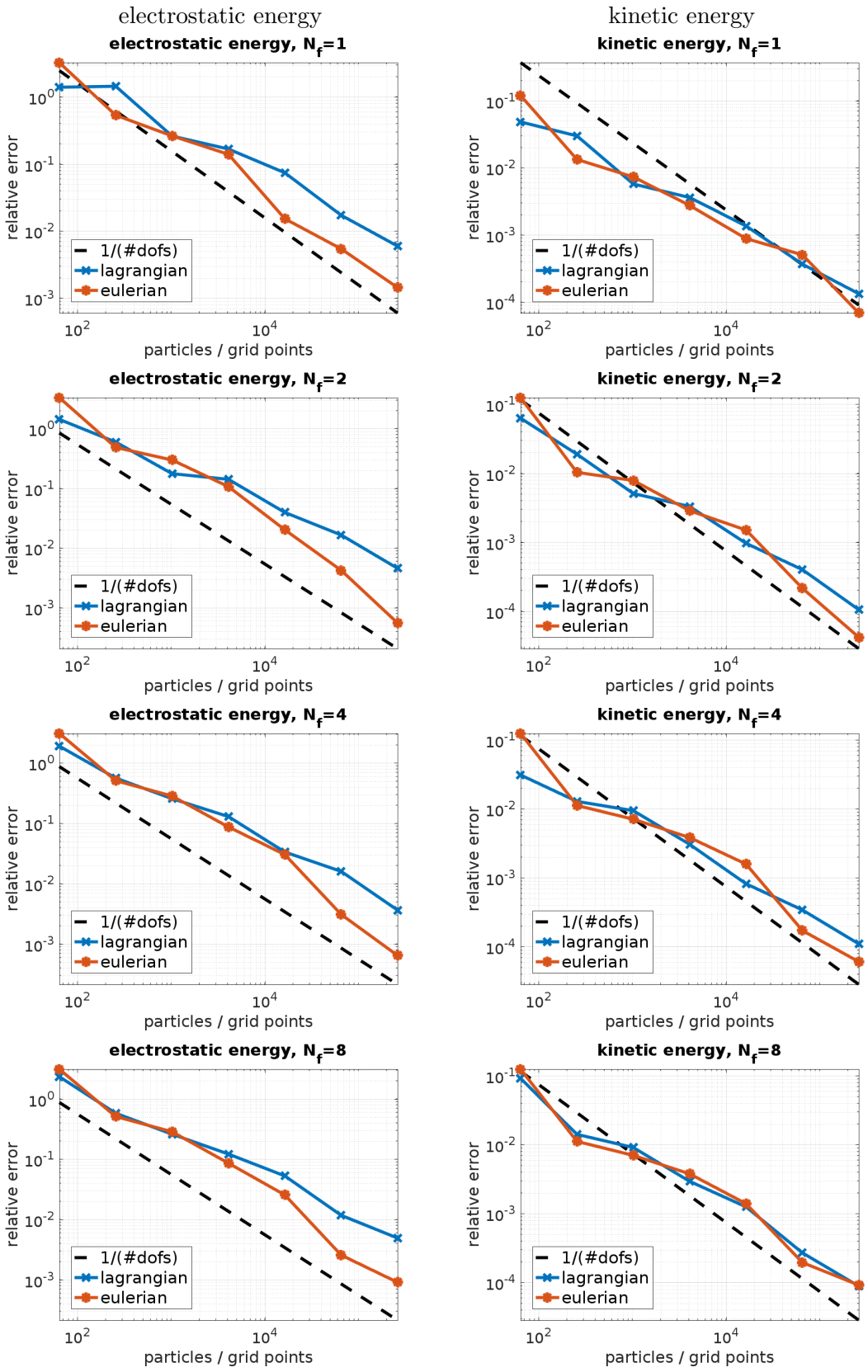
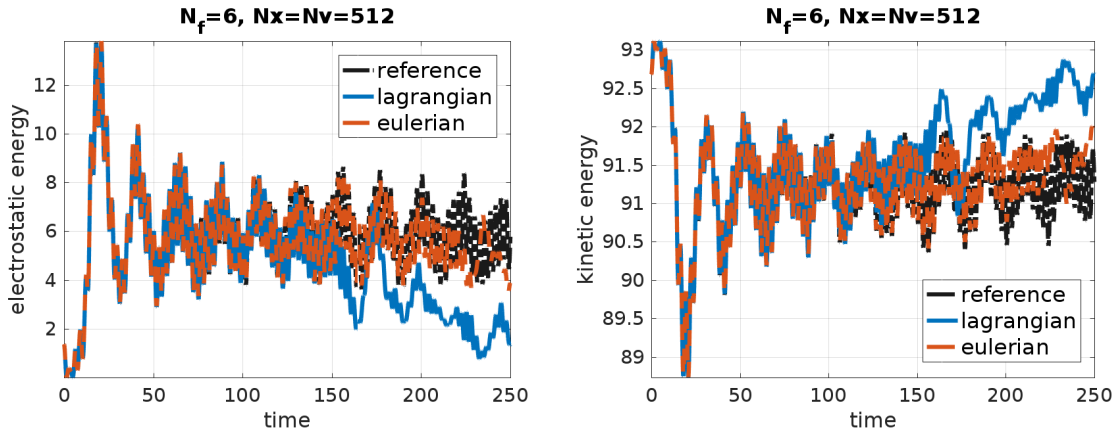


Figure 3.40.: Comparison between a pseudo-spectral Eulerian and the Lagrangian PIF Vlasov–Poisson solver for a Bump-on-tail instability for different number of Fourier modes N_f. For PIF degrees of freedom are particles N_p respectively total number of grid points $N_x \cdot N_v$ for the spectral solver. For better comparison the errors are obtained from the electrostatic energy and the kinetic energy.



(a) electrostatic energy

(b) kinetic energy

Figure 3.41.: For smaller wavelengths $m = 3$ the pseudo-spectral solver is clearly more efficient than PIF, since PIF is unable to capture the nonlinear plasma oscillations for long time. $N_p = 512^2$, $N_x = N_v = 512$.

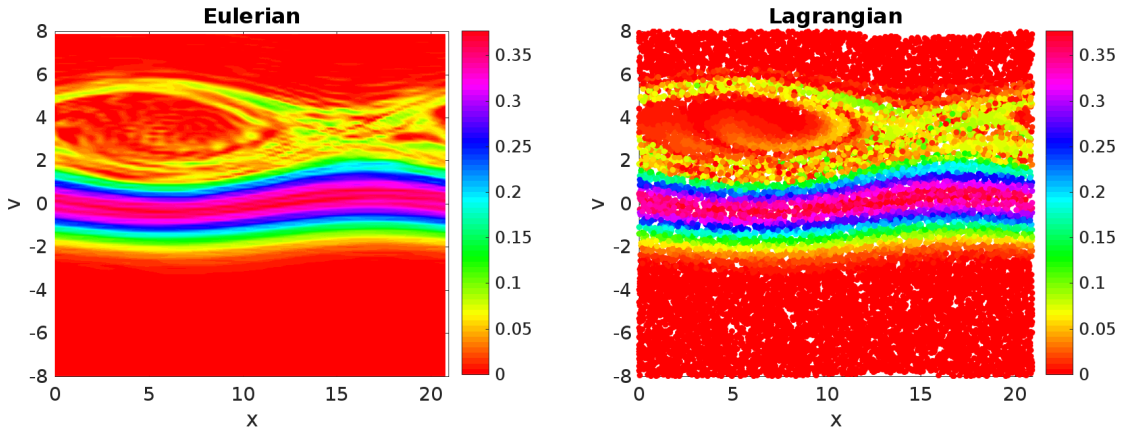
extensive PIC versus spectral comparison can be found in [158, 157], which absolutely favors the spectral solver for already a moderate error.

3.7.2. Variance reduction

Can a possibly coarse result of the Eulerian solver be used to reduce the variance of the Lagrangian method? Since the phase space density is directly given in Fourier modes, those can be easily applied as a control variate $h(x, v) = f_{Eulerian}(x, v)$ for PIF. For this the particles are sampled uniformly in the same domain as the spectral solver, $g(x, v) = [L(v_{\max} - v_{\min}]^{-1}$. Uniform sampling enhances the correlation thus the variance reduction by the initial condition as control variate is also given. The test moment is set to $\int_0^L x(x - \frac{L}{2})(x - L)f(x, v)\ \mathrm{d}x\mathrm{d}v$. The variance reduction by the spectral solution levels in at a factor of 100, which is five times larger then the reduction by the initial condition at 21, see fig. 3.43c. The results of both method coincide, as can bee seen in fig. 3.43a and fig. 3.43b. When comparing the phase space at the end of the simulation, the particles draw a sharper image, see fig. 3.42b, than the spectral representation which suffers from aliasing, see fig. 3.42a.

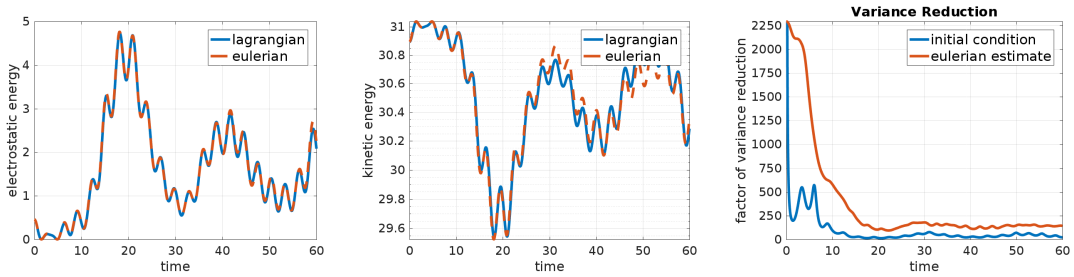
Long term high k modes

It is known that particle methods have problems resolving the high k. Nevertheless, an advantage of Monte Carlo integration over the grid based quadrature is that the error of estimating a higher mode does not increase with the mode number, but only depends on the modes amplitude. By considering only a single Fourier mode at a large k no extra amplitudes are present such that the coefficient of variance depends only on the amplitude of this mode. With larger k the Eulerian solver requires more resolution and grid-points but not the PIF solver. By construction of such an example we are able to outperform the Eulerian solver with our Lagrangian particles, see fig. 3.44.



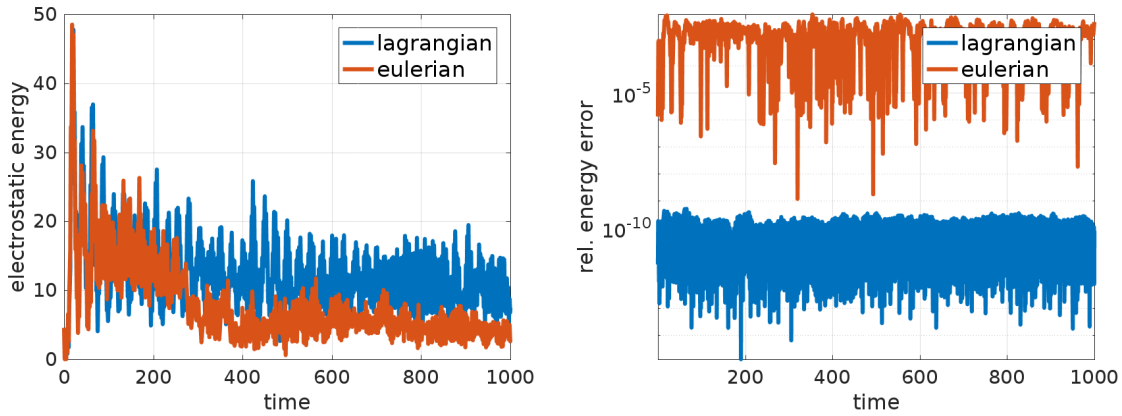
(a) Eulerian (pseudo-spectral) (b) Lagrangian (PIF)

Figure 3.42.: Phase space at $t = 60$ in the bump-on-tail instability. PIF (b) uses the pseudo-spectral solver (a) as control variate. $N_f = 1,\ N_p = 128^2,\ N_x = N_v = 128,\ \Delta t = 0.05$



(a) Electrostatic energy in the hybrid simulation. (b) Kinetic energy in the hybrid simulation. (c) Variance reduction for initial condition and Eulerian solver as control variate.

Figure 3.43.: Energies (a,b) and variance reduction (c) in a VP Eulerian-Lagrangian hybrid simulation.



(a) electrostatic energy (b) energy error

Figure 3.44.: Filtering exactly one small wavelength $m = 10, N_f = 1$ by a Fourier filter in the Eulerian and Lagrangian simulation PIF exhibits a greater stability than the pseudo spectral solver for long times ($N_p = 128^2$, $N_x = N_v = 128$).

Chapter 4.

Pseudo spectral discretizations

Complementary to the Lagrangian PIF, the next closest relative in the Eulerian family of discretization are pseudo-spectral solvers. Of course they suffer from the curse of dimensionality but not on the computational level here, since the FFTW library is well optimized, see fig. 4.1. Constant coefficient advection in a periodic domain can be solved exactly in Fourier space. In all cases treated here there is a Hamiltonian splitting available yielding constant advection possible. Fourier for the Vlasov equation spectral solvers that employ also a Fourier transform in velocity space date back to [20, 205]. Such Fourier-Fourier solver were further developed for higher dimensions [206, 207] and also extended to the Vlasov–Maxwell equation [207, 113],[22]. For Vlasov–Poisson it has been shown that Fourier filtering can be used to suppress the recurrence phenomenon [156] or filter filamentations [155]. For Vlasov–Poisson the Hamiltonian splitting has also been known [21], but for Maxwell none of these splitting methods are of geometric origin.
It should be mentioned that for the velocity space discretization also Chebyshev and Hermite polynomials have been used [20, 75]. A low degree Hermite polynomials provide an elegant way to approximate a fluid model on the numerical level.
A priori structure should be conserved for long terms and e.g. energy conservation is just a consequence but not the goal itself. Fourier spectral methods do not conserve positivity of the distribution function. In this context we neglect the question on positivity conserving schemes although for other forms of discretizations there have been improvements in that direction [208, 4, 209].
After the setting in Fourier space is discussed by means of our favorite Vlasov–Poisson example a new Fourier spectral Vlasov–Maxwell solver is presented based on a Hamiltonian splitting.

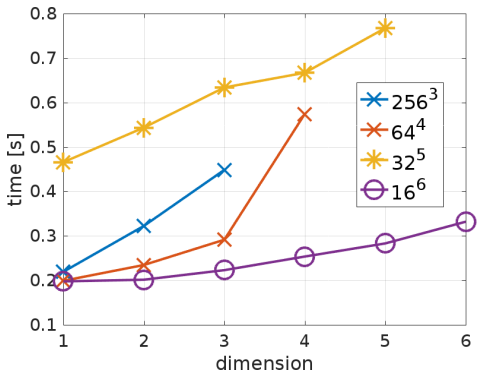


Figure 4.1.: Fourier transforming a multidimensional array along one particular dimension yields a strided access pattern resulting in a slowdown. Timings are shown for forth- and back-transform in MATLAB (using FFTW) on a laptop. Although a slowdown is visible it is not prohibitive for high dimensional spectral methods.

4.1. Vlasov–Poisson–Fokker–Planck (1d1v)

We consider the one dimensional Vlasov–Fokker–Planck equation (4.1).

$$\partial_t f(x,v,t) + v\partial_x f(x,v,t) + \frac{q}{m}\left(E(x,t) + E_{ext}(x,t)\right)\partial_v f(x,v,t) = \theta\frac{\partial}{\partial v}\left((v-\mu(x))f(x,v,t)\right) - \underbrace{\frac{\sigma(x)^2}{2}}_{=D(x)}\frac{\partial^2 f(x,v,t)}{\partial v^2} \tag{4.1}$$

Here we Fourier transform in velocity and spatial space where $\hat{f}$ denotes a transformation. For notational simplicity the transformed dimension is indicated by k_x or k_v in the argument. The spatial, velocity and fully Fourier transformed densities are defined as

$$\hat{f}(k_x,v,t) = \frac{1}{L}\int_0^L f(x,v,t)\mathrm{e}^{-\mathrm{i}xk_x}\,\mathrm{d}x, \tag{4.2}$$

$$\hat{f}(x,k_v,t) = \frac{1}{v_{\max}-v_{\min}}\int_{v_{\min}}^{v_{\max}} f(x,v,t)\mathrm{e}^{-\mathrm{i}(v-v_{\min})k_v}\,\mathrm{d}v, \tag{4.3}$$

$$\hat{f}(k_x,k_v,t) = \frac{1}{L}\frac{1}{v_{\max}-v_{\min}}\int_{v_{\min}}^{v_{\max}}\int_0^L f(x,v,t)\mathrm{e}^{-\mathrm{i}(xk_x+(v-v_{\min})k_v)}\,\mathrm{d}x\mathrm{d}v, \tag{4.4}$$

where the wave vectors are $k_x = n\frac{2\pi}{L}$ and $k_v = \frac{2\pi}{v_{\max}-v_{\min}}$ for $n \in \mathrm{Z}$. Note that one can easily by a Fourier forth and back-transform switch between those three representations on a discrete level. We split the integration in three parts in $\tau[0,t]$, where the Vlasov steps can be integrated exactly in Fourier space.

1. Advection in x

$$\partial_t f(x,v,t) + v\partial_x f(x,v,t) = 0 \tag{4.5}$$

2. Advection in v and Poisson solve

$$\partial_t f(x,v,t) + \frac{q}{m}\left(E(x,0) + E_{ext}(x,0)\right)\partial_v f(x,v,t) = 0 \tag{4.6}$$

   Here we solve the Poisson equation with constant background (for $q=-1$), but other fields are also possible.

$$\partial_x E(x,t) = 1 + q\int f(x,v,t)\,\mathrm{d}v \tag{4.7}$$

3. Fokker–Planck Collisions

$$\underbrace{\theta\frac{\partial}{\partial v}\left((v-\mu(x))f(x,v,t)\right)}_{\text{drift}} - \underbrace{D(x)\frac{\partial^2 f(x,v,t)}{\partial v^2}}_{\text{diffusion}} \tag{4.8}$$

For the splitting we consider the time $[0,t]$ to be one time step.

1. Advection in x in spatially transformed space

$$\partial_t \hat{f}(k_x,v,t) = -v\mathrm{i}k_x\hat{f}(k_x,v,t). \tag{4.9}$$

   The constant coefficient advection is integrated exactly over this splitting step.

$$\begin{aligned}\hat{f}(k_x,v,t) &= \hat{f}(k_x,v,0) + \int_0^t(-v\mathrm{i}k_x)\hat{f}(k_x,v,\tau)\mathrm{d}\tau \\ &= \hat{f}(k_x,v,0)\mathrm{e}^{-v\mathrm{i}k_x t}\end{aligned} \tag{4.10}$$

2. Advection in v in velocity transformed space

$$\begin{aligned}\partial_t \hat{f}(x,k_v,t) &= -\frac{q}{m}\left(E(x,0)+E_{ext}(x,0)\right)\mathrm{i}k_v\hat{f}(k,k_v,t)\\ \hat{f}(x,k_v,t) &= \hat{f}(x,k_v,0)\mathrm{e}^{-\frac{q}{m}(E(x,0)+E_{ext}(x,0))\mathrm{i}k_v t}.\end{aligned} \tag{4.11}$$

Note that in this step the advection in v cancels out under the velocity integral.

$$\int_{\mathbb{R}} f(x,v,t)\mathrm{d}v = \int_{\mathbb{R}} f(x,v+t\frac{q}{m}\left[E(x,0)+E_{ext}(x,0)\right],0)\mathrm{d}v = \int_{\mathbb{R}} f(x,v,0)\mathrm{d}v \tag{4.12}$$

Therefore, the electric field can be obtained in the spatially transformed space before or at the end of the split step.

$$\hat{E}(k_x,0) = q\frac{1}{\mathrm{i}k_x}\int \hat{f}(k_x,v,0)\ \mathrm{d}v,\ \text{ for } k_x \neq 0 \tag{4.13}$$

3. The drift term in the Fourier transformed Fokker–Planck collision operator (4.14) poses a problem since it contains a derivative in Fourier space $\partial_{k_v}\hat{f}(x,k_v,t)$.

$$\begin{aligned}\partial_t\hat{f}(x,k_v,t) &= \theta\left[(1-\mu(x)\mathrm{i}k_v)\hat{f}(x,k_v,t)+\mathrm{i}\partial_{k_v}\left(\mathrm{i}k_v\hat{f}(x,k_v,t)\right)\right]+D(x)k_v^2\hat{f}(x,k_v,t)\\ &= \theta\left[(1-\mu(x)\mathrm{i}k_v)\hat{f}(x,k_v,t)-\hat{f}(x,k_v,t)-k_v\partial_{k_v}\hat{f}(x,k_v,t)\right]+D(x)k_v^2\hat{f}(x,k_v,t)\\ &= -\theta k_v\left[\mu(x)\mathrm{i}+\partial_{k_v}\hat{f}(x,k_v,t)\right]+D(x)k_v^2\hat{f}(x,k_v,t)\end{aligned} \tag{4.14}$$

The remaining terms form an ODE, which is nothing new. Let $\mathcal{F}_v$ denote the Fourier transform v, $\mathcal{F}_v^{-1}$ the corresponding back-transform and $v\cdot$ the multiplication with v. The derivative in Fourier space can be expressed by back-transforming according to eqn. (4.15).

$$\mathrm{i}\partial_{k_v}\hat{f}(x,k_v,t) = \mathcal{F}_v\underbrace{[v\cdot]\,\mathcal{F}_v^{-1}\hat{f}(x,k_v,t)}_{=vf(x,v,t)} \tag{4.15}$$

This introduces aliasing on the discrete level such that exact integration is only guaranteed to a certain precision that depends on the amount of Fourier filtering. Recall that Fourier transform is only a linear operation and on the coefficient level $v\cdot$ corresponds to multiplication with a diagonal matrix containing the corresponding velocity grid points. This means that a linear operator $L = \mathcal{F}_v\,[v\cdot]\,\mathcal{F}_v^{-1}$ can be defined, which has the following property

$$\mathrm{e}^{tL} = \mathrm{e}^{t\mathcal{F}_v[v\cdot]\mathcal{F}_v^{-1}} = \mathrm{e}^{\mathcal{F}_v t[v\cdot]\mathcal{F}_v^{-1}} = \mathcal{F}_v\mathrm{e}^{t[v\cdot]}\mathcal{F}_v^{-1}. \tag{4.16}$$

For further investigation we refer to [210], where a pseudo spectral based Fokker–Planck solver with an exponential time differentiating scheme is discussed and also [211].

The Lie steps can be composed by symmetric composition, see [21]. The symplectic Runge Kutta scheme from Forest and Ruth [32] also works as it is just shifted by a half step and, therefore, adjoint symplectic for the Eulerian discretization.

4.2. Vlasov–Maxwell (1d2v)

The Hamiltonian splitting was already discussed extensively for Lagrangian particles, nevertheless it is also possible to derive the same method for a spectral discretization. For a different, but incorrect [212], splitting this has already been done in [22]. Here we use the

correct Hamiltonian splitting from [213]. Let $f(x, v_1, v_2, t)$ denote the plasma density and $\hat{f}$ the Fourier transform. Since there are six different combinations of transforms $\hat{f}$ denotes a transformation, where the transformed dimension is indicated as before by k_x, k_{v_1} or k_{v_2} in the argument. That means $\hat{f}(k_x, v_1, k_{v_2})$ denotes the Fourier transform of f in x and v_2. We begin by treating the Hamiltonian splitting for time integration from 0 to t.

- Kinetic energy ($d = 1$), $\mathcal{H}_{p_1} = \frac{1}{2}\iiint v_1^2 f(x, v, t)\,\mathrm{d}x\mathrm{d}v_1\mathrm{d}v_2$

$$\begin{aligned} \partial_t f(x, v_1, v_2, t) + v_1 \partial_x f(x, v_1, v_2, t) - \frac{q}{m} B_3(x, t) v_1 \partial_{v_2} f(x, v_1, v_2, t) = 0 \\ \partial_t B_3(x, t) = 0 \\ \partial_t E_1(x, t) = -q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_1 f(x, v_1, v_2, t) \mathrm{d}v_1 \mathrm{d}v_2 \end{aligned} \tag{4.17}$$

  The first problem, but luckily the only problem we will encounter, is the Fourier transform for the Vlasov density, since Fourier transforming in x and v_1 simultaneously results in terms containing convolutions:

$$\partial_t \hat{f}(k_x, v_1, k_{v_2}, t) + v_1 \mathrm{i} k_x \hat{f}(k_x, v_1, k_{v_2}, t) - \frac{q}{m} \hat{B}_3(k_x, t) *_{k_x} v_1 \mathrm{i} k_{v_2} \hat{f}(k_x, v_1, k_{v_2}, t) = 0. \tag{4.18}$$

  This can be avoided by considering only the Fourier transform in v_2 such that (4.17) can be solved exactly by

$$\begin{aligned} & \partial_t \hat{f}(x, v_1, k_{v_2}, t) + v_1 \partial_x \hat{f}(x, v_1, k_{v_2}, t) - \frac{q}{m} B_3(x, 0) v_1 \mathrm{i} k_{v_2} \hat{f}(x, v_1, k_{v_2}, t) = 0 \\ \Leftrightarrow\ & \partial_t \hat{f}(x, v_1, k_{v_2}, t) = -\left[v_1 \partial_x \ - \frac{q}{m} B_3(x, 0) v_1 \mathrm{i} k_{v_2} \right] \hat{f}(x, v_1, k_{v_2}, t) \\ \Rightarrow\ & \hat{f}(x, v_1, k_{v_2}, t) = \exp\left\{ -t \underbrace{v_1 \left[\partial_x \ - \frac{q}{m} B_3(x, 0) \mathrm{i} k_{v_2} \right]}_{=\mathcal{L}} \right\} \hat{f}(x, v_1, k_{v_2}, 0) \end{aligned} \tag{4.19}$$

  Here the exponential contains still the derivative ∂_x which can be — and this is a critical point here — exactly obtained at the grid points $x_1, \ldots x_{N_x}$ for the spectral discretization by Fourier forth and back-transform. For this recall that the discrete Fourier transform can be denoted in a matrix[1] $\mathcal{F}_x \in \mathbb{R}^{N_x \times N_x}$ and $\mathcal{F}_x^{-1}$. Hence the matrix $L \in \mathbb{R}^{N_x \times N_x}$ representing the discrete but exact counterpart of $\mathcal{L}$ reads

$$L = \underbrace{v_1 \mathcal{F}_x^{-1} \mathrm{diag}\left(\mathrm{i}k_1, \ldots, \mathrm{i}k_{N_x}\right) \mathcal{F}_x}_{=L_A} \underbrace{- v_1 \frac{q}{m} \mathrm{i} k_{v_2} \mathrm{diag}\left(B(x_1, 0), \ldots, B(x_{N_x}, 0)\right)}_{L_L}. \tag{4.20}$$

  By calculating the matrix exponential $\exp(-tL)$ the systems of ODE arising from evaluating eqn. (4.19) at every spatial grid point can be solved exactly for each v_1 and k_{v_2}. Now it is obviously highly questionable to replace a fast Fourier transform with a matrix, and although there are matrix free variants of the standard algorithms available [214] we follow a much simpler approach. Note that $\exp(tL_A)$ and $\exp(tL_L)$ are as (transformed) diagonal matrices trivial to calculate respectively to apply onto a vector $\left(\hat{f}(x_1, v_1, k_{v_2}), \ldots, \hat{f}(x_{N_x}, v_1, k_{v_2})\right)$ but unfortunately L_L and L_A do not commute. In such a situation Moler [215] suggests to use the Trotter product formula

$$\mathrm{e}^{-\frac{t}{m} L} \mathrm{e}^{-\frac{t}{m}(L_A + L_L)} = \lim_{m \to \infty} \left(\mathrm{e}^{-\frac{t}{m} L_A} \mathrm{e}^{-\frac{t}{m} L_L} \right)^m. \tag{4.21}$$

[1] Instead of assembling the matrix by hand, one can just Fourier transform an identity matrix of the appropriate size. In this way one always obtains the correct normalization, e.g. in MATLAB $fft(eye(N_x), [], 1)$ and $ifft(eye(N_x), [], 1)$.

Essentially this means, we should split $\hat{\mathcal{H}}_{p_1}$ into two parts which can be solved exactly in Fourier space and then sub-step these parts to the desired accuracy. Splitting eqn. (4.17) in the Vlasov—Ampère $\mathcal{H}_{p_{1,A}}$ part and the remaining terms of the Lorentz force $\mathcal{H}_{p_{1,L}}$ yields

$$\begin{aligned}
\mathcal{H}_{p_{1,A}} &\begin{cases} \partial_t f(x,v_1,v_2,t) + v_1 \partial_x f(x,v_1,v_2,t) = 0, \\ \partial_t E_1(x,t) = -q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_1 f(x,v_1,v_2,t) \mathrm{d}v_1 \mathrm{d}v_2, \end{cases} \\
\mathcal{H}_{p_{1,L}} &\begin{cases} \partial_t f(x,v_1,v_2,t) - \frac{q}{m} B_3(x,t) v_1 \partial_{v_2} f(x,v_1,v_2,t) = 0, \\ \partial_t B_3(x,t) = 0. \end{cases}
\end{aligned} \tag{4.22}$$

The advection in $\mathcal{H}_{p_{1,A}}$ can be again directly solved by a Fourier transform in x,

$$\hat{f}(k_x,v_1,v_2,\tau) = \hat{f}(k_x,v_1,v_2,0) \mathrm{e}^{-v_1 \mathrm{i} k_x \tau} \text{ for } \tau \in [0,t]. \tag{4.23}$$

The electric field is, identical as in Vlasov–Ampère, obtained by inserting the time evolution (4.23) yielding:

$$\begin{aligned}
\hat{E}(k_x,t) &= \hat{E}(k_x,0) - q \int_0^t \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_1 \hat{f}(k_x,v_1,v_2,\tau) \mathrm{d}\tau \mathrm{d}v_1 \mathrm{d}v_2 \\
&= \hat{E}(k_x,0) - q \int_0^t \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_1 \hat{f}(k_x,v_1,v_2,0) \mathrm{e}^{-v_1 \mathrm{i} k_x \tau} \mathrm{d}\tau \mathrm{d}v_1 \mathrm{d}v_2 \\
&= \hat{E}(k_x,0) - q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_1 \hat{f}(k_x,v_1,v_2,0) \int_0^t \mathrm{e}^{-v_1 \mathrm{i} k_x \tau} \mathrm{d}\tau \mathrm{d}v_1 \mathrm{d}v_2 \\
&= \hat{E}(k_x,0) - q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_1 \hat{f}(k_x,v_1,v_2,0) \frac{1}{-v_1 \mathrm{i} k_x} \left[\mathrm{e}^{-v_1 \mathrm{i} k_x \tau} \right]_0^t \mathrm{d}v_1 \mathrm{d}v_2 \\
&= \hat{E}(k_x,0) + q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} \hat{f}(k_x,v_1,v_2,0) \frac{1}{\mathrm{i} k_x} \left[\mathrm{e}^{-v_1 \mathrm{i} k_x t} - 1 \right] \mathrm{d}v_1 \mathrm{d}v_2.
\end{aligned} \tag{4.24}$$

The second part $\mathcal{H}_{p_{1,L}}$ reduces to a constant coefficient advection in v_2 and is solved directly by

$$\hat{\mathcal{H}}_{p_{1,L}} \Big\{ \ \hat{f}(x,k_{v_2},t) = \mathrm{e}^{\frac{q}{m} B_3(x,t) v_1 \ \mathrm{i} k_{v_2} t} \hat{f}(x,k_{v_2},0). \tag{4.25}$$

Note that the split step $\mathcal{H}_{p_{1,A}}$, given in eqns. (4.23) and (4.24) can also be performed in v_2 transformed space, thus, both eqn.(4.26) and eqn. (4.27) can be used.

$$\hat{\mathcal{H}}_{p_{1,A}} \begin{cases} \hat{f}(k_x,v_1,v_2,t) & = \hat{f}(k_x,v_1,v_2,0) \mathrm{e}^{-v_1 \mathrm{i} k_x t} \\ \hat{E}(k_x,t) & = \hat{E}(k_x,0) + q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} \hat{f}(k_x,v_1,v_2,0) \frac{1}{\mathrm{i} k_x} \left[\mathrm{e}^{-v_1 \mathrm{i} k_x t} - 1 \right] \mathrm{d}v_1 \mathrm{d}v_2. \end{cases} \tag{4.26}$$

$$\hat{\mathcal{H}}_{p_{1,A}} \begin{cases} \hat{f}(k_x,v_1,k_{v_2},t) = \hat{f}(k_x,v_1,k_{v_2},0) \mathrm{e}^{-v_1 \mathrm{i} k_x t} \\ \hat{E}(k_x,t) = \hat{E}(k_x,0) \\ \qquad\qquad + q \int_{v_1^{\min}}^{v_1^{\max}} \hat{f}(k_x,v_1,k_{v_2}=0,0) \frac{1}{\mathrm{i} k_x} \left[\mathrm{e}^{-v_1 \mathrm{i} k_x t} - 1 \right] \mathrm{d}v_1 \left(v_2^{\max} - v_2^{\min} \right) \end{cases} \tag{4.27}$$

In order to obtain a symmetric splitting of $\mathcal{H}_{p_1}$ the following two second order options are available by Strang splitting, where φ denotes the corresponding flux:

$$\begin{aligned}
\varphi_{p_1}(\Delta t) &= \varphi_{p_{1,A}}\left(\frac{\Delta t}{2}\right) \circ \varphi_{p_{1,L}}(\Delta t) \circ \varphi_{p_{1,A}}\left(\frac{\Delta t}{2}\right) \\
\varphi_{p_1}(\Delta t) &= \varphi_{p_{1,L}}\left(\frac{\Delta t}{2}\right) \circ \varphi_{p_{1,A}}(\Delta t) \circ \varphi_{p_{1,L}}\left(\frac{\Delta t}{2}\right)
\end{aligned} \tag{4.28}$$

With and without sub-stepping of this sub-splitting there was no visible difference (relative error at $\sim 10^{-6}$ to the fields obtained with the exact full matrix exponential for our test-cases, although there is a difference to the exact integration, see fig. 4.3. For the sake of efficiency we used only the single split step in the presented simulations. The reason for this could be that the advection in eqn. (4.25) takes only place in the v_2-component such that it would not affect the integration of the Ampère eqn. (4.24) in $\mathcal{H}_{p_{1,A}}$ where the velocity v_2 is integrated out. This means that the resulting field E is exactly the same as in the original $\mathcal{H}_{p_1}$ and Gauss' law is conserved.

- Kinetic energy ($d = 2$), $\mathcal{H}_{p_2} = \frac{1}{2}\iiint v_2^2 f(x, v, t)\ \mathrm{d}x\mathrm{d}v_1\mathrm{d}v_2$

$$\begin{aligned} \partial_t f(x, v_1, v_2, t) + \frac{q}{m} v_2 B_3(x, t)\partial_{v_1} f(x, v_1, v_2, t) = 0 \\ \partial_t E_2(x, t) = -q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_2 f(x, v_1, v_2, t)\mathrm{d}v_1\mathrm{d}v_2 \end{aligned} \tag{4.29}$$

Since there is no advection in x we know that the transport in v_1 averages out by

$$\int_{v_1^{\min}}^{v_1^{\max}} f(x, v_1, v_2, \tau)\mathrm{d}v_1 = \int_{v_1^{\min}}^{v_1^{\max}} f(x, v_1, v_2, 0)\mathrm{d}v_1 \quad \forall \tau \in [0, t], \tag{4.30}$$

such that $\mathcal{H}_{p_2}$ can be integrated exactly in a single step yielding the final discretization

$$\hat{\mathcal{H}}_{p_2} \begin{cases} \hat{f}(x, k_{v_1}, v_2, t) = \hat{f}(x, k_{v_1}, v_2, 0)\mathrm{e}^{-\mathrm{i}k_{v_1} v_2 \frac{q}{m} B_3(x,0)t} \\ \hat{E}_2(k_x, t) = \hat{E}_2(k_x, 0) - t \cdot q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} v_2 \hat{f}(k_x, v_1, v_2, 0)\mathrm{d}v_1\mathrm{d}v_2. \end{cases} \tag{4.31}$$

- Electric energy, $\mathcal{H}_E = \frac{1}{2}\int |E(x, t)\|^2\ \mathrm{d}x$

$$\begin{aligned} \partial_t f + \frac{q}{m} E_1(x, t)\partial_{v_1} f(x, v_1, v_2, t) + \frac{q}{m} E_2(x, t)\partial_{v_2} f(x, v_1, v_2, t) = 0 \\ \partial_t B_3(x, t) = -\partial_x E_2(x, t) \\ \partial_t E(x, t) = 0 \end{aligned} \tag{4.32}$$

The advection is constant in (v_1, v_2) and varies only in x, such that the constant coefficient advection can be solved exactly in Fourier space.

$$\hat{\mathcal{H}}_E \begin{cases} \hat{f}(x, k_{v_1}, k_{v_2}, t) = \hat{f}(x, k_{v_1}, k_{v_2}, 0)\mathrm{e}^{-\mathrm{i}\frac{q}{m}\left(E_1(x,0)k_{v_1} + E_2(x,0)k_{v_2}\right)t} \\ \hat{B}_3(k_x, t) = \hat{B}_3(k_x, 0) - t \cdot \mathrm{i}k_x \hat{E}_2(k_x, t) \end{cases} \tag{4.33}$$

- Magnetic energy, $\mathcal{H}_B = \frac{1}{2}\int \|B(x, t)^2\|\ \mathrm{d}x$

$$\begin{aligned} \partial_t E_2(x, t) = -\partial_x B_3(x, t) \\ \partial_t E_1(x, t) = \partial_t B_3(x, t) = 0 \end{aligned} \tag{4.34}$$

$$\hat{\mathcal{H}}_B \left\{ \hat{E}_2(k_x, t) = \hat{E}_2(k_x, 0) - t\ \mathrm{i}k_x \hat{B}(k_x, 0) \right. \tag{4.35}$$

For the initialization of the simulation the electric field E_1 is obtained by the Poisson equation, which reduces in one dimension to Gauss' law. In Fourier space Gauss' law reads

$$\hat{E}_1(k_x, t) = \frac{1}{\mathrm{i}k_x} \underbrace{q \int_{v_2^{\min}}^{v_2^{\max}} \int_{v_1^{\min}}^{v_1^{\max}} \hat{f}(k_x, v_1, v_2, t)\mathrm{d}v_1\mathrm{d}v_2}_{:=\hat{\rho}(k_x,t)} \quad \text{for } k_x \neq 0. \tag{4.36}$$

Gauss' law is preserved during the entire simulation, such that we denote the error on eqn. (4.36) at final time as $\mathcal{P}_\epsilon$, which should be close to machine precision. Instead of the standard second order Strang splitting using two Lie steps, we prefer a second order method which has less than half the error constant of the Strang splitting [216]. It requires four Lie steps and is given by symmetric composition of a flux φ with its adjoint φ^* as

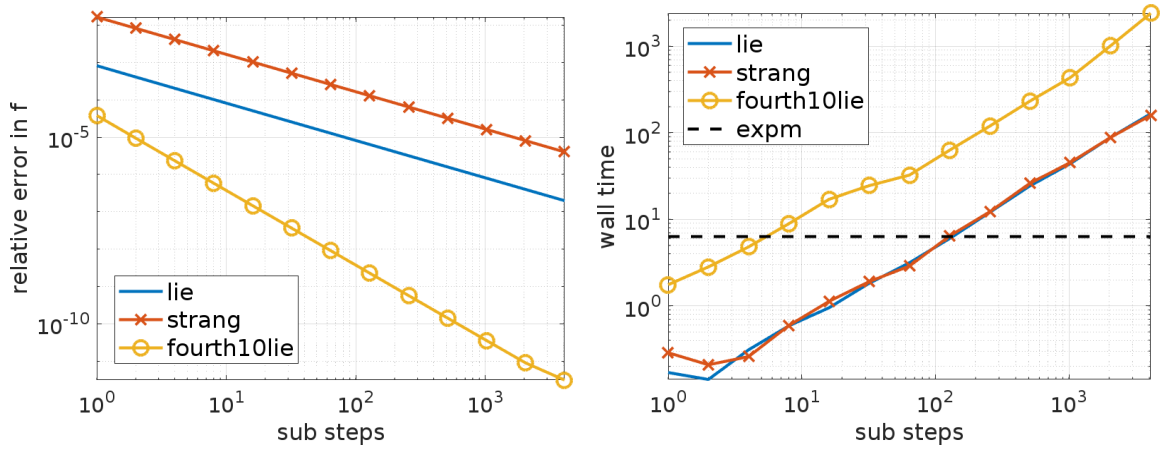
$$\varphi_{\alpha\Delta t} \circ \varphi^*_{(1/2-\alpha)\Delta t} \circ \varphi_{(1/2-\alpha)\Delta t} \circ \varphi^*_{\alpha\Delta t},\ y_2 = (2\sqrt{326} - 36)^{1/3},\ \alpha = \frac{y_2^2 + 6y_2 - 2}{12y_2}. \tag{4.37}$$

In the following four tests with varying initial conditions resulting in nonlinear Landau damping, the Weibel and the Weibel streaming instability with parameters according to [19, 217] are performed. The second order splitting in eqn. (4.37) is used for the time discretization. In most cases the energy error is taken as a measure of correctness, yet the strength of the presented scheme is the preservation of structure, such that the energy error can be misleading, because choice of a small enough time step, short simulation time and a sufficient resolution can mimic conservation. If the structure preserving method is implemented correctly a simulation will exhibit long term stability, despite an insufficient resolution in time and space. Here we also want to point out that the perfect energy conservation in [19] for the Weibel instability was only achieved by high order integrators and all presented results presented perfectly coincide with the PIC and PIF, except that they do not exhibit noise and are from the authors experience much cheaper to obtain. In general, the PIF still performs better for simulations containing only very few modes. Stable results for low resolution are found in figs. 4.4, 4.5, and for better resolution in fig. 4.6, 4.7, 4.8. The default parameters are denoted in eqn. (4.38) along with the initial condition (4.39), which were adapted from [217]. We conclude that it is not hard to obtain a geometric Eulerian method for the Vlasov–Maxwell simulations. Given the affinity of the presented scheme to PIF, it fits perfectly in the scope of this thesis since parallel development of Eulerian and Lagrangian codes creates confidence in the obtained results.

$$
\begin{aligned}
&\text{default} && \epsilon, \beta_r, \beta_i, v_{0,1}, v_{0,2}, \delta, B_0 = 0,\ c = 1, \sigma_1, \sigma_2 = 1 \\
& && N = N_x = N_{v_1} = N_{v_2} = 32,\ \Delta t = 0.05 \\
&\text{Strong Landau} && \epsilon_e = 0.5,\ k = 0.5, \\
& && v_{1,\max} = 4.5\sigma_1,\ v_{1,\min} = -v_{1,\max},\ v_{2,\max} = 4.5\sigma_2,\ v_{2,\min} = -v_{2,\max} \\
&\text{Weibel} && \beta_r = -10^{-3},\ k = 1.25,\ \sigma_1 = \tfrac{0.02}{\sqrt{2}},\ \sigma_2 = \sqrt{12}\sigma_1, \\
& && v_{1,\max} = 4.5\sigma_1,\ v_{1,\min} = -v_{1,\max},\ v_{2,\max} = 4.5\sigma_2,\ v_{2,\min} = -v_{2,\max} \\
&\text{Weibel streaming sym.} && \sigma_1 = \sigma_2 = \tfrac{0.1}{\sqrt{2}},\ k = 0.2,\ \beta_i = 10^{-3},\ v_{0,1} = 0.3,\ v_{0,2} = -0.3,\ \delta = \tfrac{1}{2} \\
& && v_{1,\max} = 0.9,\ v_{1,\min} = -v_{1,\max},\ v_{2,\max} = 0.9,\ v_{2,\min} = -v_{2,\max} \\
&\text{Weibel streaming asym.} && \sigma_1 = \sigma_2 = \tfrac{0.1}{\sqrt{2}},\ k = 0.2,\ \beta_i = 10^{-3},\ v_{0,1} = 0.5,\ v_{0,2} = -0.1,\ \delta = \tfrac{1}{6} \\
& && v_{1,\max} = 0.3\ (or\ 0.7),\ v_{1,\min} = -v_{1,\max},\ v_{2,\max} = 1.05,\ v_{2,\min} = -0.55
\end{aligned}
\tag{4.38}
$$

$$
\begin{aligned}
f(x, v_1, v_2, t = 0) &= \frac{1 + \epsilon \cos(kx)}{2\pi\sigma_1\sigma_2^2} \mathrm{e}^{-\frac{v_1^2}{2\sigma_1^2}} \left(\delta \mathrm{e}^{-\frac{(v_2 - v_{0,1})^2}{2\sigma_2^2}} + (1 - \delta) \mathrm{e}^{-\frac{(v_2 - v_{0,2})^2}{2\sigma_2^2}} \right) \\
B_3(x, t = 0) &= \beta_r \cos(kx) + \beta_i \sin(kx) \\
E_2(x, t = 0) &= \alpha_r \cos(kx) + \alpha_i \sin(kx) \\
\partial_x E_1(x, t = 0) &= 1 - \int_{\mathbb{R}^2} f(x, v_1, v_2, t)\ \mathrm{d}v
\end{aligned}
\tag{4.39}
$$

Figure 4.2.: Parameters and corresponding initial conditions for different Vlasov–Maxwell (1d2v) test-cases. The most challenging cases are the symmetric and asymmetric Weibel streaming instability.



(a) L^2 error (b) wall time

Figure 4.3.: By use of the matrix exponential *expm* the split step $\mathcal{H}_{p_1}$ can be integrated exactly, but it is not matrix free and does at the moment not take advantage of the fast Fourier transform. But it can also be approximated by a sub stepped splitting, which is shown here for the asymmetric Weibel streaming instability at $t = t_{\max} = 300$ in the fully nonlinear phase for $N_x = N_v = 128$. Many sub steps are required to approximate $\mathcal{H}_{p_1}$, such that high order methods are required (a) since the matrix exponential is comparably efficient (b). Nevertheless experiments have shown that there was no visible difference in the fields for the presented test-cases when only two sub-steps where chosen.

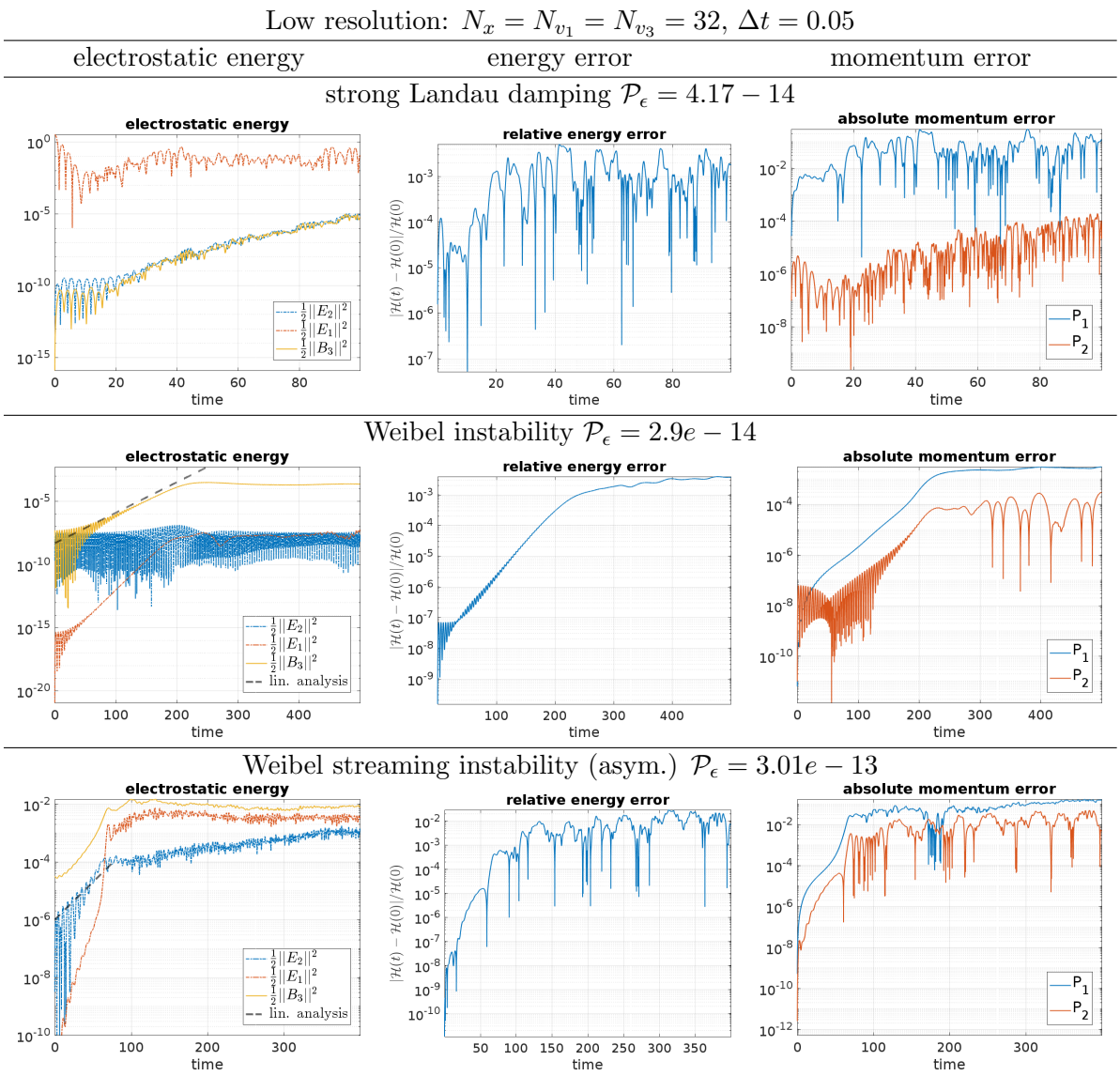


Figure 4.4.: Electrostatic energy, relative energy error and the momentum error in the two velocity components for different test cases of the Vlasov–Maxwell 1d2v geometric pseudo-spectral solver. The time discretization is performed by a second order Strang splitting. Although the resolution with just 32 grid points per dimension is very low, the solver appears to be stable over longer times.

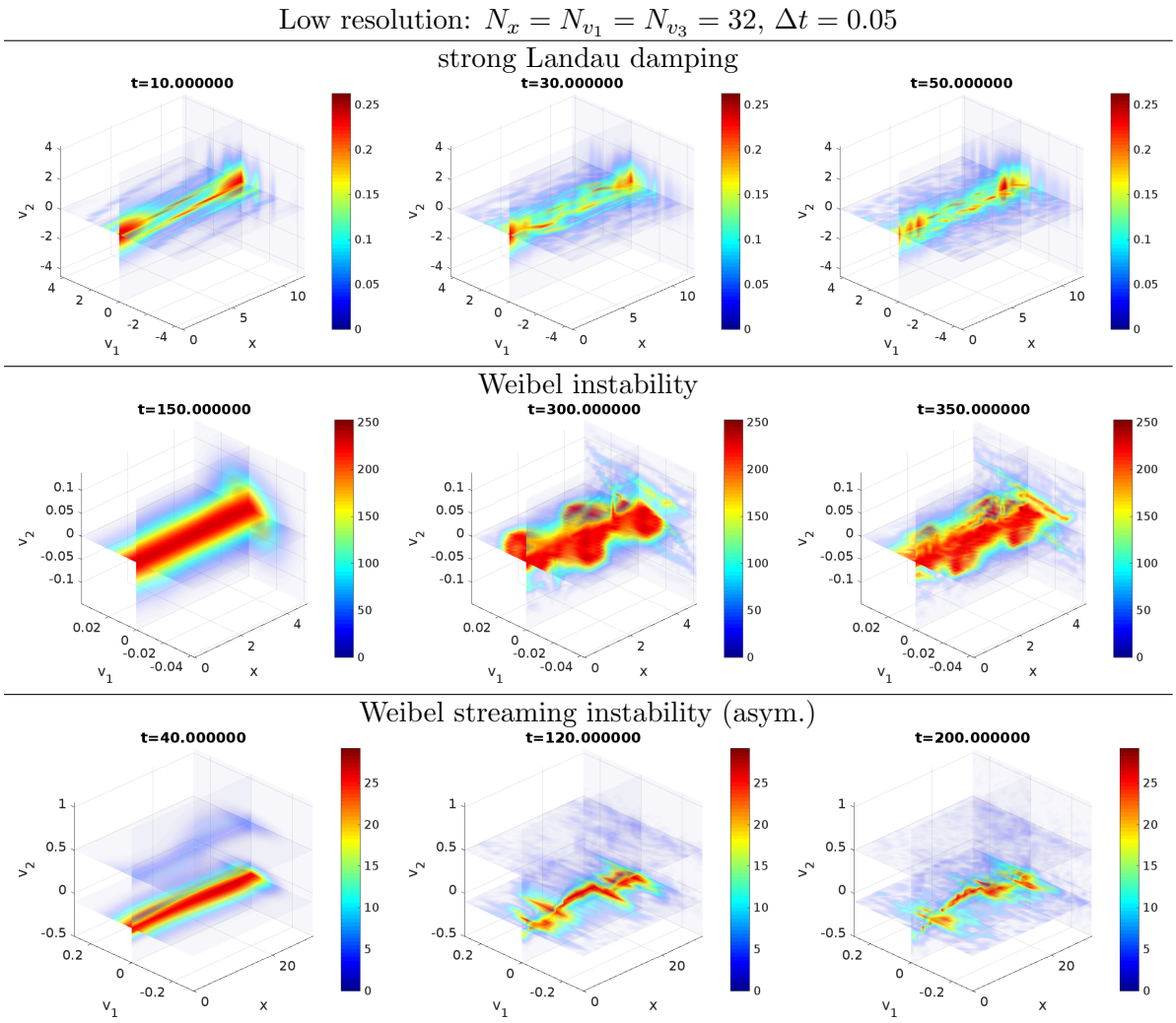


Figure 4.5.: Phase space densities for Vlasov–Maxwell 1d2v simulations under low resolution.

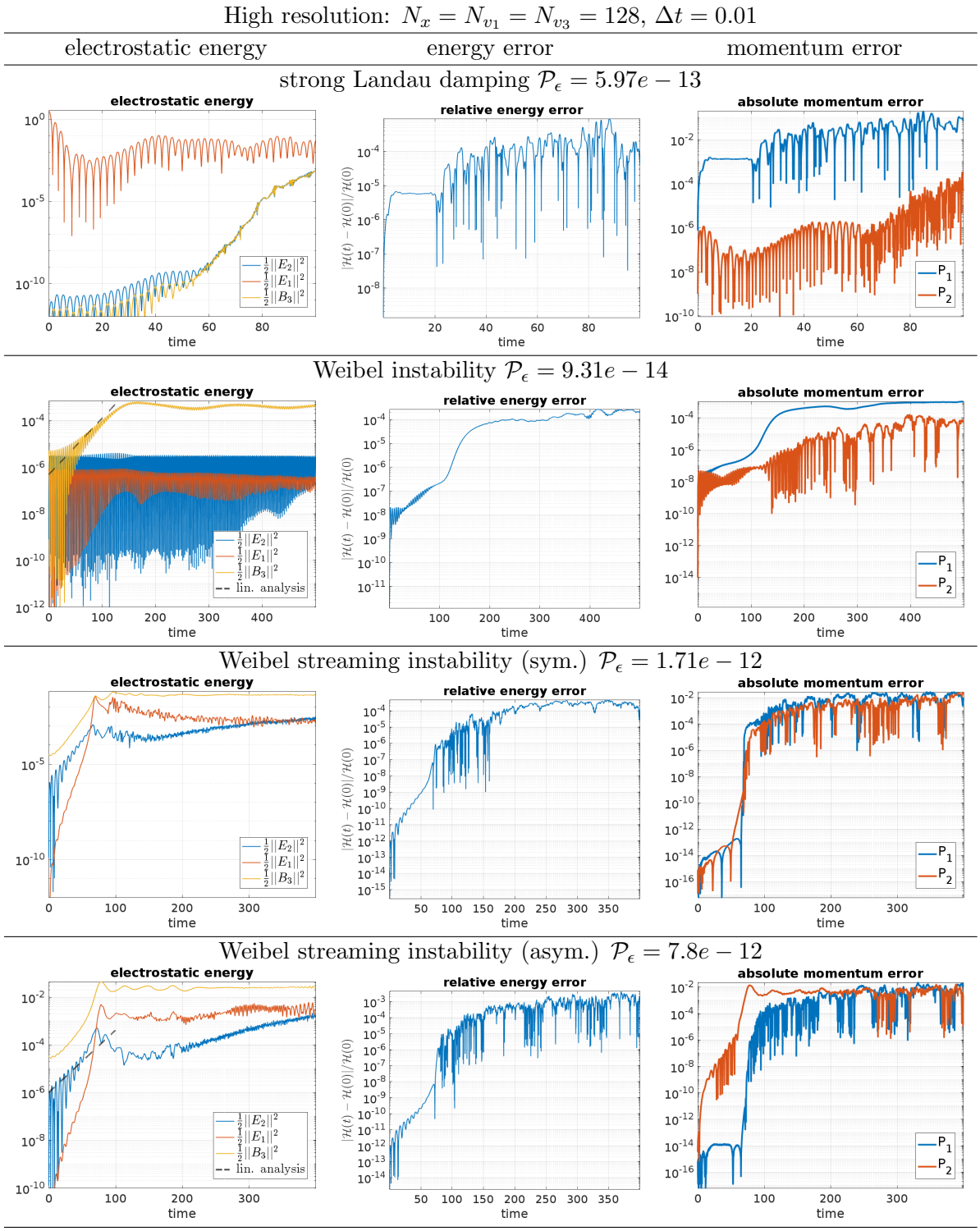


Figure 4.6.: High resolution results for three Vlasov–Maxwell 1d2v simulations with the geometric pseudo-spectral solver. The energy error is smaller than in the low resolution but remains at a high level, which is comparable to the GEMPIC[19] results, where a smaller energy error was only achieved with a high order splitting.

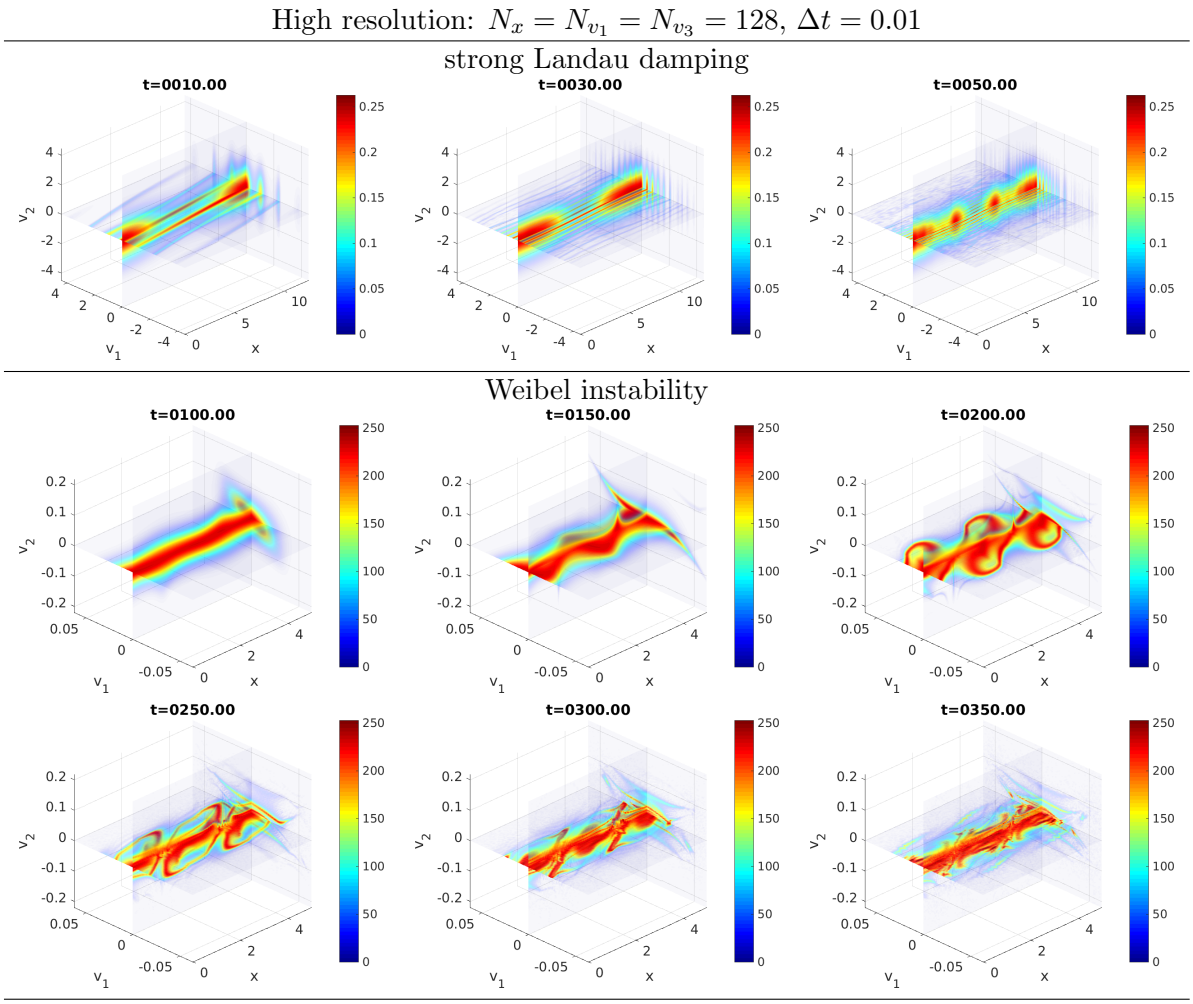
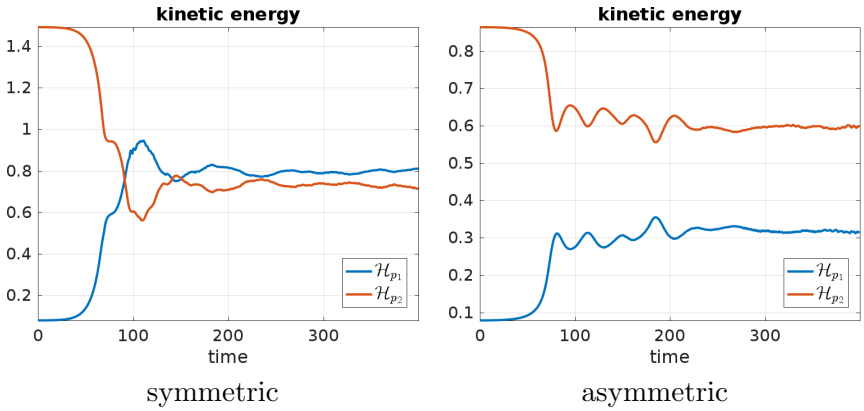


Figure 4.7.: Phase space densities for Vlasov–Maxwell 1d2v simulations under high resolution.



symmetric asymmetric

Figure 4.8.: Kinetic energy for the symmetric and asymmetric Weibel streaming instability at high resolution.

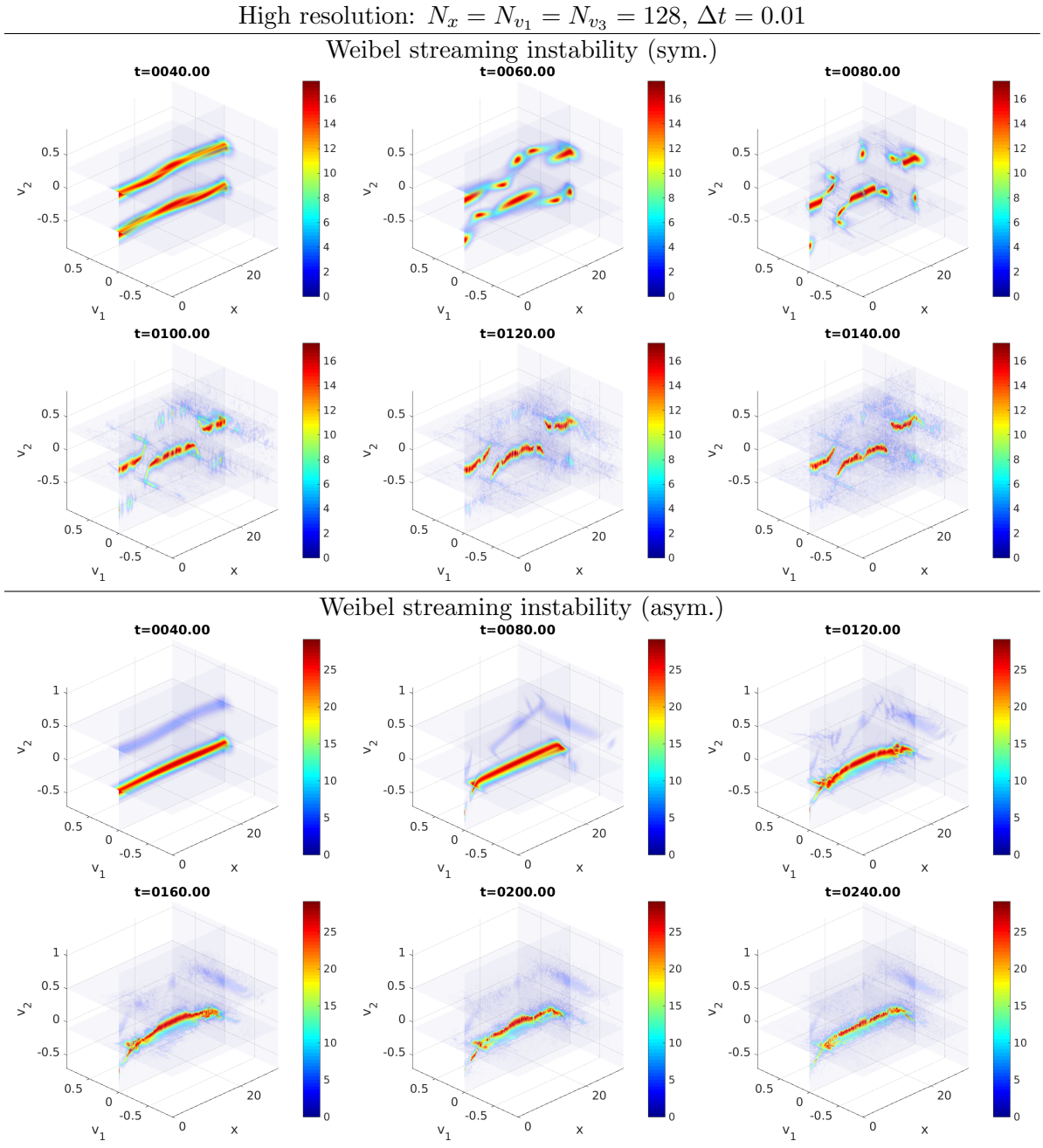


Figure 4.9.: Phase space densities for Vlasov–Maxwell 1d2v simulations under high resolution.

Chapter 5.

Conclusion and Outlook

The goal of this work is to lay a foundation for stochastic and spectral particle methods and to demonstrate the application of new methods for the Vlasov equation. Starting from this basic work some worthwhile paths are already clear to see for the future. But, as always in numerics, it is fruitful to provide some code to build upon.

5.1. Exemplatory codes

For reproducibility a github repository[1] is provided containing all MATLAB and julia codes used in this thesis. Almost all figures containing comparisons have their own script that produces these plots and studies precisely, and which is also independent of the machine. Only comparisons using extensive computations will not run out of the box. This repository is most useful if you want to e.g. investigate the presented variance reduction methods for other test-cases or modify them. Digging in foreign codes is labor intensive, such that clean, simple and performant examples suited for educational purposes or sparking future work are provided for free[2].

- Vlasov-Poisson (1d1v) Micro-benchmark (MATLAB, julia, python)

MATLAB:

- Vlasov-Poisson-Fokker-Planck (1d1v) PIC and PIF with δf
- Vlasov-Maxwell (3d3v) PIF
- Vlasov-Maxwell (1d2v) Multi-Species PIF
- Vlasov-Poisson (1d1v) Pseudo-Spectral
- Vlasov-Maxwell (1d2v) Pseudo-Spectral
- Vlasov-Poisson (1d1v) Dispersion Relations
- Guiding-center (2d) PIC with Triangles

julia:

- Vlasov-Poisson (3d3v) PIF
- Vlasov-Maxwell (3d3v) PIF
- Vlasov-Poisson (1d1v) Particle in Chebyshev (with *ApproxFun.jl*) and Legendre

Although the MATLAB codes are quite performant and can be used on Nvidia GPUs using MATLAB's GpuArrays, the author strongly encourages everyone to port the MATLAB codes to julia because the particle methods can easily be made scalable using MPI, and GPU support is also becoming more and more attractive.

[1] `https://github.com/ameresj/StochasticSpectralParticles`
[2] `http://jakobameres.com`

5.2. Stochastics

A unifying approach on the existing stochastic particle simulations in plasma physics was presented. Variance as a measure of error for the fields was established, allowing for better diagnostics. It then became clear that PIC suffers from the fact that small amplitudes are obscured by the high level of the particle noise. The control variate mechanism proved to be a versatile technique to reduce the variance which shadows the small amplitudes. In the linear phase it is reasonably efficient to use the initial condition directly as control variate, which allowed us to obtain dispersion relations with the help of the matrix pencil method. Given the issues of PIC with small amplitudes it is certainly not a good idea to obtain dispersion relations in that way as a primary objective, but one should use a spectrally perturbed Ansatz, see [218]. It is also interesting how to obtain a dispersion relation by linearization around the current state in the nonlinear phase in order to assess stability properties, since this requires density estimation of the full distribution. For density reconstruction OSDE has only little overhead, is easy to implement and can also provide control variates in the nonlinear phase. The control variate can even be viewed as a projection, which [219, 104] might make a combination with symplectic integrators possible. This is necessary, because long term stability is required in the nonlinear phase. The situation looks worse for conditional Monte Carlo, which is most useful when the values transported by the markers are subject to a change, as it is the case for collisions. Properties can be baked into the distribution but the changes are mostly so violent that hardly any improvements are made as they destroy the dynamic. Standard Monte Carlo could be improved but it does not work for QMC. Thus, the best option so far is to combine uniform sampling, QMC and an OSDE control variate. Multilevel Monte-Carlo is an interesting extension of the control variate mechanism, which is most promising for the combination of models constructed on different time scales, but is not a remedy for the small amplitude noise. It is also clear, that every moment-guided simulation should rather use the control variate to enforce constraints than the crude moment matching. The control variate could be applied in other particle simulations [144, 220] where additional information about certain moments come from another set of equations.
The particle noise catches on to anything unstable in the system, and therefore it is so intuitive to find new physics with PIC. When the transport along a complex magnetic field dominates over the effects by the self consistent field, accurate results can be obtained with few particles compared to the high dimensionality of the problem. Monte Carlo in two dimensions does not make much sense and we have learned that the variance in PIC also increases with the dimension. In the simplest scenario the one dimensional problem is extended into three dimensions by an outer product. Therefore, a particle method has to be sufficiently accurate already in one dimension in order to be competitive. Restricting the variance by filtering or directly the dimensionality of the spatial space can, of course, turn the situation in favor to PIC. Therefore, using PIC for a 1d2v or 2d3v Vlasov–Maxwell model is more appropriate than discretizing a 3d2v gyrokinetic density with Monte-Carlo markers, and performing a 3d density estimate with the additional dimension for the quadrature used in the gyro-average on top, which drags a four dimensional bias and a three dimensional variance along.

5.3. Spectral methods

PIF codes are even easier to implement than PIC, but when optimized they are better vectorized along the modes than along the particles. The same applies for Chebyshev and Legendre discretization for the fields. Fourier modes as eigenfunctions of the Laplace operator provide a natural way of filtering thus reducing the variance, which is the largest obstacle

for Monte-Carlo particle methods. For any given geometry the Laplace operator guides us to an appropriate coordinate system and suitable basis functions, which definitely should be used for PIC if available. This becomes more complex once an arbitrary possibly field aligned geometry is needed, but it is still possible by combining PIF and PIC. Given the fact that the filtered Fourier modes are no eigenfunctions it is questionable whether the field alignment is appropriate. Global spectral methods in general curvilinear coordinates lead mostly to dense and ill-conditioned matrices [54]. Additionally it might not even be the best option to describe the domain by one single global mapping but simply cut out the D-shaped poloidal plane from a square, see [221]. For the Poisson equation, cylinder coordinates (R, Z, φ) and a Legendre-Fourier Ansatz for (R, φ) result in a tridiagonal matrix such that only the poloidal plane (R, Z) has to be taken care of. For future work we suggest using the fictitious domain method with internal forcing, which was successfully applied to the Poisson equation in [222]. In general using PIF in the toroidal direction is always a good idea.
Fourier space is beautiful, hence the flawless properties of PIF for Vlasov–Poisson and Vlasov–Maxwell, but there are even more opportunities. The gyro-average, which corresponds to a circular motion, can be described by Bessel functions in Fourier space, but an exponential integration scheme can also benefit from the Fourier structure in the same way. We have showed that the 1d2v Monte Carlo can be extended into the time domain for a semi implicit Hamiltonian splitting. In future work, together with exponential integration the gyromotion can be resolved implicitly at low costs and it is even possible to overcome the plasma frequency as well, as other experiments have shown [223]. Yet this requires a fully implicit particle methods, such that it might be easier to follow the Fourier spectral approach presented in this work.

Appendix

Appendix A.

Mixed stochastic and deterministic methods

A.1. Randomizing deterministic quadrature

In previous examples considering the gyroaverage operator we saw that the periodic midpoint rule can easily be randomized yielding an unbiased estimator that is as least as performant as the plain quadrature rule. But this is of course only possible in periodic domains. Therefore, we treat two forms of variance reduction by quadrature rules in bounded domains namely Chebyshev and Gauss-Lobatto quadrature.

A.1.1. Chebyshev

Another quadrature rule for bounded domains, but also based on equidistant points is the Chebyshev quadrature on $[-1,1]$. For this the interval $[-1,1]$ is bended onto a half circle, the integrand mirrored onto the full circle and since the circle is periodic equidistant quadrature nodes are the method of choice. Therefore, define the transformation from the circle $\theta \in [0, 2\pi]$ to the domain $x \in [-1,1]$ by $x = \cos(\theta)$. Note that the corresponding Jacobian is given as $\frac{\mathrm{d}}{\mathrm{d}\theta}\cos(\theta) = -\sin(\theta) = -\sqrt{1-\cos(\theta)^2}$ and can also be expressed in terms of x as $-\sqrt{1-x^2}$. Chebyshev quadrature uses the equidistant points on the unit circle, whic The randomized Chebyshev quadrature is then derived by

$$\begin{aligned}
\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,\mathrm{d}x &= \int_{\cos^{-1}(-1)}^{\cos^{-1}(1)} \frac{f(\cos(\theta))}{\sqrt{1-\cos(\theta)^2}}(-\sin(\theta))\,\mathrm{d}\theta \\
&= \int_0^{\pi} f(\cos(\theta))\,\mathrm{d}\theta = \frac{1}{2}\int_0^{2\pi} f(\cos(\theta))\,\mathrm{d}\theta \\
&\approx \frac{\pi}{N}\sum_{n=1}^{N} f(\underbrace{\cos(\theta_n)}_{:=x_n}) = \frac{\pi}{N}\sum_{n=1}^{N} f(x_n) \\
\theta_n &:= 2\pi\frac{n-u}{N}, \quad u \sim \mathcal{U}(0,1).
\end{aligned} \tag{A.1}$$

For $u = \frac{1}{2}$ the first half of the nodes x_j coincides with the other half such resulting in the standard Chebyshev quadrature rule. Thus an additional factor of two in the number of quadrature yields spectral convergence and an unbiased Monte Carlo estimator. Thus it is only useful if the number of markers is so large, that it can make up for this factor of two. The weight factor $\frac{1}{\sqrt{1-x^2}}$ can be included in the quadrature weights.
If $f(x,v)$ is smooth in v, like a Maxwellian, the numerical quadrature gains efficiency.

A.1.2. Gauss–Lobatto

For the standard bounded quadrature rules, like Gauss–Legendre and Gauss-Lobatto the equidistant stratification technique cannot by applied anymore. Thus we turn to a different

idea that of uses orthonormal polynomials, which originally stems from [224]. For one dimension examples for the randomization of the trapezoidal and Gauss-Lobatto rule are derived in [225][p.69-47], but overcomplicated and not generalized. Thus we present a new general method of how to combine deterministic quadrature and interpolation in order to reduce the variance of the Monte Carlo quadrature. Actually the control variate is underlying mechanism for the formulas derived in [225]. A Gaussian quadrature rule with quadrature nodes and weights $(x_j, w_j)_{j=0,\dots,N}$ approximates an integral as

$$\mathcal{J} = \int_\Omega f(x)\ \mathrm{d}x \approx \sum_{j=1}^{N} w_j f(x_j). \tag{A.2}$$

For the spectral methods quadrature and interpolation are tightly connected, see [54], such that a numerically well behaving approximation of f can be made by Lagrange interpolation using the quadrature nodes (x_j). The Lagrange polynomials for given nodes (x_j) are defined as

$$\ell_j(x) := \prod_{\substack{1\le n\le N\\ n\neq j}} \frac{x - x_n}{x_j - x_n}. \tag{A.3}$$

The interpolation $\mathcal{I}_f$ of f using the nodes (x_j) and the corresponding Lagrange polynomials ℓ_j reads

$$\mathcal{I}_f(x) = \sum_{j=1}^{N} f(x_j)\ell_j(x), \tag{A.4}$$

and can be exactly integrated by the quadrature rule eqn. (A.2) which yields

$$\int_\Omega \mathcal{I}_f(x)\ \mathrm{d}x = \int_\Omega \sum_{j=0}^{N} f(x_j)\ell_j(x)\ \mathrm{d}x = \sum_{j=1}^{N} f(x_j) \underbrace{\int_\Omega \ell_j(x)\ \mathrm{d}x}_{=w_j} = \sum_{j=1}^{N} w_j f(x_j). \tag{A.5}$$

This connection holds true for all N-point Gaussian quadrature rules which are exact at least up to a polynomial degree N. Then the standard Monte Carlo integral for eqn. (A.2) is rewritten as $\delta f = f - \mathcal{I}_f$ scheme using the interpolation $\mathcal{I}_f$. For a random deviate A distributed according to some probability density p the δf Monte Carlo integral reads

$$\int_\Omega f(x)\ \mathrm{d}x = \mathbb{E}\left[\frac{f(A)}{p(A)}\right] = \mathbb{E}\left[\frac{f(A) - \mathcal{I}_f(A)}{p(A)}\right] + \int_\Omega \mathcal{I}_f(x)\ \mathrm{d}x. \tag{A.6}$$

Inserting eqn. (A.5) into eqn. (A.6) connects the quadrature rule with the Monte Carlo estimator by

$$\int_\Omega f(x)\ \mathrm{d}x = \mathbb{E}\left[\frac{f(A) - \mathcal{I}_f(A)}{p(A)}\right] + \sum_{j=1}^{N} w_j f(x_j). \tag{A.7}$$

Unfortunately, even if the quadrature rule is exact for a higher degree than N the Monte Carlo estimator of eqn. (A.7) is only exact up to degree N, because N-point Lagrange interpolation is only exact up to degree N yielding the δf as $f - \mathcal{I}_f = 0$ to drop out. Nevertheless if there is some freedom in the choice of p we can at least reduce the variance by choosing p close to $\delta f = f - \mathcal{I}_f$. Since $\ell_i(x_j) = \delta_{ij}$ the interpolation is always exact at the quadrature nodes, which means that δf vanishes there.

$$\delta f(x_j) = f(x_j) - \mathcal{I}_f(x_j) = 0 \text{ for all } j = 1, \dots, N. \tag{A.8}$$

Since we do not know more about δf, we let the sampling density p also vanish at the nodes (x_j) by defining

$$p(x) = \frac{\left|\prod_{j=1}^{N}(x - x_j)\right|}{\int_\Omega \left|\prod_{j=1}^{N}(x - x_j)\right| \, \mathrm{d}x}. \tag{A.9}$$

In one dimension the method of choice for sampling from p is inverse transform sampling, by using the inverse P^{-1} of the cumulative sampling density $P(y) = \int_0^y P(x)dx$. Unfortunately closed expressions for P^{-1} are harder to find with increasing degree N, where we have to fall back to Newtons method. In the following we revise and complement the mechanism given in [225][p.69-47]. We start with the trapezoidal rule for $\Omega = [0, 1]$ with $(x_j) = \{0, 1\}, (w_j) = 0.5$, where the interpolation reads

$$\mathcal{I}_f(x) = f(0) + (f(1) - f(0))\, x = f(0)(1 - x) + f(1)x. \tag{A.10}$$

$$p(\alpha) = 6\alpha(1 - \alpha), \alpha \in [0, 1] \tag{A.11}$$

We can sample the random deviate A from p by inverse transform sampling for a uniformly distributed $u \sim \mathcal{U}(0, 1)$

$$A = P^{-1}(u) = \mathcal{R}\left\{\frac{1}{4}\left(-(1 + \mathrm{i}\sqrt{3})\sqrt[3]{-2u + 2\sqrt{(u-1)u} + 1} + \frac{\mathrm{i}(\sqrt{(}3) + \mathrm{i})}{\sqrt[3]{-2u + 2\sqrt{(u-1)u} + 1}} + 2\right)\right\} \tag{A.12}$$

Inserting in (A.7) yields an estimator exact for quadratic polynomials, see eqn. (A.13).

$$\mathcal{J} = \mathbb{E}\left[\frac{f(A) - (1 - A)f(0) - Af(1)}{6A(1 - A)}\right] + \frac{f(0) + f(1)}{2} \tag{A.13}$$

An interesting trick from [225], is to introduce antithetic sampling by the random deviate $1 - A$ resulting in an estimator accurate for cubic polynomials in eqn. (A.14).

$$\begin{aligned}\mathcal{J} &= \frac{1}{2}\mathbb{E}\left[\frac{f(A) - (1 - A)f(0) - Af(1)}{6A(1 - A)} + \frac{f(1 - A) - (A)f(0) - (1 - A)f(1)}{6(1 - A)A}\right] + \frac{f(0) + f(1)}{2} \\ &= \mathbb{E}\left[\frac{f(A) + f(1 - A) - f(0) - f(1)}{12A(1 - A)}\right] + \frac{f(0) + f(1)}{2}\end{aligned} \tag{A.14}$$

Note that for eqn. (A.13) the special case $A = \frac{1}{2}$ yields the fifth order Simpsons rule:

$$\mathcal{J} \approx \frac{1}{6}\left\{f(0) + 4f\left(\frac{1}{2}\right) + f(1)\right\}. \tag{A.15}$$

Next the order is increased by using Simpson's rule with $(x_j) = \left\{0, \frac{1}{2}, 1\right\}, \ (w_j) = \left\{\frac{1}{6}, \frac{r}{6}, \frac{1}{6}\right\}$ and the quadratic interpolation

$$\mathcal{I}_f(x) = f(0)(1 - x)(1 - 2x) + 4x(1 - x)f\left(\frac{1}{2}\right) - x(1 - 2x)f(1). \tag{A.16}$$

The canonical sampling density along with its inverse cumulative distribution function reads

$$\begin{aligned} p(x) &= 16\|x(1-x)(2x-1)| \\ P(x) &= \begin{cases} 8x^2(x-1)^2 & \text{for } x \in [0, \frac{1}{2}] \\ 1 - 8x^2(x-1)^2 & \text{for } x \in (\frac{1}{2}, 1] \end{cases} \\ P^{-1}(u) &= \begin{cases} \frac{1}{2}\left(1 - \sqrt{1 - 2\sqrt{\frac{u}{2}}}\right) & \text{for } u \in [0, \frac{1}{2}) \cdot \\ \frac{1}{2} & \text{for } u = \frac{1}{2} \\ 1 - \frac{1}{2}\left(1 - \sqrt{1 - 2\sqrt{\frac{1-u}{2}}}\right) & \text{for } x \in (\frac{1}{2}, 1] \end{cases} \end{aligned} \tag{A.17}$$

Here Hammersley wanted to avoid the absolute value and took

$$p(x) = 30x(1-x)(1-x)^2, x \in [0,1], \tag{A.18}$$

yielding a quartic (A.19) a quintic (A.20) estimator.

$$\mathcal{J} = \mathbb{E}\left[\frac{f(A) - (1-A)(1-2A)f(0)}{10A(1-A)(1-2A)^2} + A(1-2A)f(1) - 4A(1-A)f\left(\frac{1}{2}\right)\right] + \frac{f(0) + 4f\left(\frac{1}{2}\right) + f(1)}{6} \tag{A.19}$$

$$\mathcal{J} = \mathbb{E}\left[+\frac{f(A) + f(1-A) - f(0) - f(1)}{60A(1-A)} + \frac{f(A) + f(1-A) - 2f\left(\frac{1}{2}\right)}{15(1-2A)^2}\right] + \frac{f(0) + 4f\left(\frac{1}{2}\right) + f(1)}{6} \tag{A.20}$$

Nevertheless, similar to the Chebyshev case before, a factor two in the polynomial exactness of the quadrature rule is lost. So there seems to be some penalty on randomizing quadrature rules.

A.2. Enforcing constraints

A.2.1. Moment matching techniques

Moment matching during the simulation might seem quite attractive as it enforces the conservation of certain moments. In this context, Owen [69][p.33] explains the connection to control variates quite well and he also gives an interesting alternative re-weighting method that enforces positive weights [69][p.38]. Most important here is that in a survey of variance reduction method the connection between moment matching and control variates is made, see [226]. In plasma physics moment matching in simulations is also referred to as a moment guided simulation [144]. If the phase space positions of the markers were not modified but only their weights, it is straightforward to only manipulate the weights. This happens for example when applying a control variate h.

$$\delta w_k := \frac{f(x_k, v_k) - h(x_k, v_k)}{g(x_k, v_k)} = w_k - \frac{h(x_k, v_k)}{g(x_k, v_k)} \tag{A.21}$$

We define some additional constants as

$$\hat{\delta\mu}_n = \frac{1}{N_p}\sum_{k=1}^{N_p} \delta w_k (v_k)^n, \;\; \hat{v}_n = \frac{1}{N_p}\sum_{k=1}^{N_p} (v_k)^n, \;\; \hat{\delta\lambda}_n = \frac{1}{N_p}\sum_{k=1}^{N_p} (\delta w_k)^n.$$

A linear Ansatz for matching $\delta\mu_1, \delta\mu_2$ by only manipulating the weights yields

$$\delta f w_k^* = T(w_k) := \frac{\delta\mu_1 \hat{v}_2 - \delta\mu_2 \hat{v}_1}{\hat{\delta\mu_1} \hat{v}_2 - \hat{\delta\mu_2} \hat{v}_2} w_k - \frac{\delta\mu_1 \hat{\delta\mu_2} - \delta\mu_2 \hat{\delta\mu_1}}{\hat{\delta\mu_1} \hat{v}_2 - \hat{\delta\mu_2} \hat{v}_1}.$$

The velocity moments mean $\delta\mu_1$ and variance $\delta\mu_2$ can be set or kept, but mass conservation $\hat{\delta\lambda_1} = \delta\lambda_1 = 0$ is lost.

A.2.2. Constraints by control variates for full f and δf

We leave the intuitive setting of δf - sampling the difference - behind us, but keep the control variate in order to employ constraints. This is the crucial point, because work in this area are dominated by the δf idea [71][p.569],[68, 125, 47], subtracting a large enough analytical known part of the density to improve the moment estimators on the density. This idea stems from the days of linearization, that made it to a statistic method. Which is why control variate PIC is a much more accurate description than δf-PIC, what constantly leads to confusion.
We broaden our view, and follow essentially the key idea of [227]. The control variate idea was to improve the estimation of the mean of one random deviate by using the mean of another. In our familiar setting we have two moments described by the mean of $\Theta(X, V)$ and $\Phi(X, V)$. We are interested in

$$\theta = \mathbb{E}[\Theta] \approx \hat{\theta} = \frac{1}{N_p} \sum_{k=1}^{N_p} \Theta(x_k, v_k) \tag{A.22}$$

and we know a-priori the exact value of

$$\phi = \mathbb{E}[\Phi]. \tag{A.23}$$

We define a new random variable

$$\Theta^* := \Theta - \alpha\Phi + \alpha\phi \tag{A.24}$$

with same expectation as Θ,

$$\theta = \mathbb{E}[\Theta^*] = \mathbb{E}[\Theta] \underbrace{-\alpha\mathbb{E}[\phi] + \alpha\phi}_{=0}. \tag{A.25}$$

But its variance is reduced by

$$\mathbb{V}[\Theta^*] = (1 - \rho)\,\mathbb{V}[\Theta], \quad \rho := \frac{\mathbb{COV}[\Theta, \Phi]}{\mathbb{V}[\Theta]\,\mathbb{V}[\Phi]} \tag{A.26}$$

for the coefficient α set to

$$\alpha = \frac{\mathbb{COV}[\Theta, \Phi]}{\mathbb{V}[\Phi]}. \tag{A.27}$$

We can always estimate α by estimating variances and covariances. But now instead of focusing on finding a suitable control variate Φ for Θ by constructing Φ in ways such that it is correlated to Θ, we think a different way following [227].
By interpreting equation (A.23) as a constraint, we can impose with the control variate estimator $\hat{\Theta}^*$ the constraint (A.23) onto the original estimator $\hat{\theta}$. Potentially this has a broad application, as there are many constraints in Vlasov simulation, namely all preserved quantities.

	error	variance
$\hat{\theta}$	3.604366E-03	2.194456E+02
$\hat{\theta}^*$	1.858271E-04	3.541589E+01
moment matched $\hat{\theta}$	4.708416E-16	2.210685E+02

Figure A.1.: Initial condition for bump-on-tail instability with control variate and moment matching for $N_p = 10^4$.

Kinetic energy

An exampled for a conserved quantity is the momentum, here without mass normalization,

$$\phi := \iint v f(x,v,t)\ \mathrm{d}v = \iint v f(x,v,t=0)\ \mathrm{d}v \tag{A.28}$$

This is a constraint that can be put into a random variable with $\Phi := V(t)W(t)$

$$\mathbb{E}\left[\Phi\right] = \mathbb{E}\left[V(t)W(t)\right] = \phi = \iint v f(x,v,t=0)\ \mathrm{d}v. \tag{A.29}$$

We now want to estimate another quantity, the kinetic energy $\theta = \iint v^2 f(x,v,t)$ associated with the random variable $\Theta := V(t)^2 W(t)$.

$$\theta = \mathbb{E}\left[\Theta\right] = \mathbb{E}\left[V(t)^2 W(t)\right] \tag{A.30}$$

By using Φ as a control variate for Θ we incorporate some prior knowledge - namely the conserved quantity - into the Monte Carlo estimator

$$\Theta^* = V(t)^2 W(t) - \alpha V(t)W(t) + \alpha\phi, \tag{A.31}$$

$$\hat{\theta}^* = \left[\frac{1}{N_p}\sum_{k=1}^{N_p} v_k(t)^2 w_k^t + \hat{\alpha} v_k(t) w_k^t\right] + \hat{\alpha}\phi. \tag{A.32}$$

The optimization parameter α is estimated by

$$\hat{\alpha} = \frac{\mathbb{C}\hat{\mathbb{O}}\mathbb{V}\left[V(t)^2 W(t), V(t)W(t)\right]}{\hat{\mathbb{V}}\left[V(t)W(t)\right]}. \tag{A.33}$$

This is most effective when the mean velocity is much greater than zero. We consider the initial density of a Bump-on-tail (2.368), where we know the mean velocity at every point in time, as momentum is conserved.

For the moment matching we also matched the second moment - the energy - analytically by the formulas provided in (2.134),(2.135),(2.136). This is a crucial point, as of course the error is down to machine precision because we enforced it. But the variance remains unchanged, whereas the control variate reduces the variance. We desire variance reduction in our simulation. This might also be the reason why approaches in [144] and [220], where only matching of moments is done and actual optimization of variance reduction is missing, lack greater impact. It should be pointed out, that for a centered Maxwellian and a symmetric two-stream this constraints have no impact, but especially no negative one. Thus it really depends on the velocity distribution, which is probably more interesting at the plasma edge.

Mass conservation

Now we want to give a simple example of how the kernel density estimation of the charge density via the finite elements can be improved. Throughout the literature [15],[68] one is often concerned with spurious effects stemming from the loss of mass conservation in the δf scheme. For example [15][p.408, eqn. (39)] just averages the weights, where [68][p.1011,eqn. (45)-(46)] is more elegant and uses moment matching, but both induce completely unknown bias.
But these effects stem rather from the finite element estimate $\hat{\rho}_h$ not having the correct mass $\theta = \int_{\Omega_x} \rho(x,t)\,\mathrm{d}x$, which is a priori known as a conserved quantity. Define $\mathbb{1}_h = \int_{\Omega_x} \psi(x)\,\mathrm{d}x$ [1]. Then the mass of the discrete charge density shall coincide with the a priori known analytical value, which reads

$$\begin{aligned}\theta = \int_{\Omega_x} \rho(x,t)\,\mathrm{d}x &= \mathbb{E}\left[W(t)\right] \\ &= \int_{\Omega_x} (\mathcal{M}\mathbb{E}\left[\psi(X(t))W(t)\right])^t \psi(x)\,\mathrm{d}x = (\mathbb{1}_h^t \mathcal{M})\,\mathbb{E}\left[\psi(X(t))W(t)\right]. \quad \text{(A.34)}\end{aligned}$$

In the case of the Galerkin discretization we know exactly

$$\int_{\Omega_x} \rho(x,t)\,\mathrm{d}x = \int_{\Omega_x} \rho_h(x,t)\,\mathrm{d}x, \tag{A.35}$$

hence no additional bias from the spatial discretization is introduced. To reduce the complexity for this example, we limit ourselves to the estimation of one coefficient $b_n(t) = \mathbb{E}\left[\psi_n(X(t))W(t)\right]$ where the constraint is

$$\theta = (\mathbb{1}_h^t \mathcal{M})\,\mathbb{E}\left[\psi(X(t))W(t)\right] \tag{A.36}$$

For the control variate - constrained Monte Carlo - estimator we define the random deviate

$$B_n^*(t) = \psi_n(X(t))W(t) \;-\; \beta_n\,(\mathbb{1}_h^t \mathcal{M})\psi(X(t))W(t) \;+\; \beta_n \underbrace{\int_{\Omega_x} \rho(x,t)}_{=\theta}, \tag{A.37}$$

with the optimization coefficient β_n

$$\beta_n = \frac{\mathbb{COV}\left[\psi_n(X(t))W(t), (\mathbb{1}_h^t \mathcal{M})\psi(X(t))W(t)\right]}{\mathbb{V}\left[(\mathbb{1}_h^t \mathcal{M})\psi(X(t))W(t)\right]}. \tag{A.38}$$

We want to emphasize that rather complex constraints involving calculations in the Galerkin space can be used. It is also straightforward to replace $W(t)$ with $\delta W(t)$ from the δf part. Here we give a complete example of how to do this but first we change the constrained to the simple mass defined as

$$\mathbb{E}\left[W(t)\right] = \theta, \tag{A.39}$$

leading to the random deviate

$$B_n^*(t) := \psi_n(X(t))W(t) \quad - \beta_n\,W(t) + \beta_n \underbrace{\int_{\Omega_x} \rho(x,t)}_{=\theta}. \tag{A.40}$$

This results in the standard Monte Carlo estimator

$$\hat{b}_n^*(t) := \beta_n\theta \;+\; \frac{1}{N_p}\sum_{k=1}^{N_p} \psi_n(x_k^t)w_k^t - \beta_n w_k^t. \tag{A.41}$$

[1] Note that for uniform B-splines, $\mathcal{M}^t \mathbb{1}_h = (1,\ldots,1)^t$ holds.

The optimization coefficient β_n has to be estimated for every $n = 1, \dots, N_h$ separately

$$\beta_n = \frac{\mathbb{COV}\left[\psi_n(X(t))W(t), W(t)\right]}{\mathbb{V}\left[W(t)\right]} \tag{A.42}$$

This improves conservation of $\int \hat{\rho}_h(x,t)\ \mathrm{d}x\mathrm{d}v = \int \rho_h(x,t)\ \mathrm{d}x\mathrm{d}v$. Now, one can show that mass is not only conserved in expectation but also in the realization. Let us incorporate this with the δf control variate for given h. The first idea is to optimize the variance with respect to the δf control variate as usual resulting in the optimization coefficient α and

$$B_n^*(t) = \psi_n(X(t)) \underbrace{\frac{f(X,V,t) - \alpha h(X,V,t)}{g(X,V,t)}}_{:=\delta W(t)} + \alpha \iint_\Omega \psi_n(x,v) h(x,v)\ \mathrm{d}x\mathrm{d}v. \tag{A.43}$$

$$\alpha_n := \frac{\mathbb{COV}\left[\psi_n(X)\frac{f(X,V,t)}{g(X,V,t)}, \psi_n(X)\frac{h(X,V,t)}{g(X,V,t)}\right]}{\mathbb{V}\left[\psi_n(X)\frac{h(X,V,t)}{g(X,V,t)}\right]} \tag{A.44}$$

For this fixed α we define the mass conserving estimator via the random variable

$$\begin{aligned} B_n^{**}(t) := \psi_n(X(t))\delta W(t) \quad & - \beta_n\ \delta W(t) \\ & + \beta_n \left(\int_{\Omega_x} \rho(x,t)\ \mathrm{d}x - \alpha_n \int_\Omega h(x,v,t)\ \mathrm{d}x\mathrm{d}v \right) \\ & + \alpha_n \iint_\Omega \psi_n(x,v) h(x,v)\ \mathrm{d}x\mathrm{d}v \end{aligned} \tag{A.45}$$

and the optimization coefficient

$$\beta_n := \frac{\mathbb{COV}\left[\psi_n(X(t))W(t), W(t)\right]}{\mathbb{V}\left[W(t)\right]}. \tag{A.46}$$

This marginal optimization coefficient has the advantage of being easy to implement in existing δf codes. Yet for the sake of completeness we solve the rather lengthy problem adapting the notation $Z_t = (X(t), V(t)),\ X_t = X(t)$,

$$\begin{aligned} &\min_{\alpha,\beta\in\mathbb{R}} \frac{1}{2}\mathbb{V}\left[B_n^{**}(t)\right] = \\ &\qquad \min_{\alpha,\beta\in\mathbb{R}} \frac{1}{2}\mathbb{V}\left[\underbrace{\psi_n(X)\frac{f(Z,t)}{g(Z,t)} - \alpha\psi_n(X)\frac{h(Z,t)}{g(Z,t)}}_{\delta f} \underbrace{-\beta\frac{f(Z,t)}{g(Z,t)} + \beta\alpha\frac{h(Z,t)}{g(Z,t)}}_{\text{mass conservation constrain}} \right] \end{aligned} \tag{A.47}$$

with the first derivatives

$$\begin{aligned} \frac{d}{d\alpha}B_n^{**}(t) = \mathbb{COV}\Bigg[& \psi_n(X)\frac{f(Z,t)}{g(Z,t)} - \alpha\psi_n(X)\frac{h(Z,t)}{g(Z,t)} - \beta\frac{f(Z,t)}{g(Z,t)} + \beta\alpha\frac{h(Z,t)}{g(Z,t)}, \\ & - \psi_n(X)\frac{h(Z,t)}{g(Z,t)} + \beta\frac{h(Z,t)}{g(Z,t)} \Bigg] \end{aligned} \tag{A.48}$$

and

$$\begin{aligned} \frac{d}{d\beta}B_n^{**}(t) = \mathbb{COV}\Bigg[& \psi_n(X)\frac{f(Z,t)}{g(Z,t)} - \alpha\psi_n(X)\frac{h(Z,t)}{g(Z,t)} - \beta\frac{f(Z,t)}{g(Z,t)} + \beta\alpha\frac{h(Z,t)}{g(Z,t)}, \\ & - \frac{f(Z,t)}{g(Z,t)} + \alpha\frac{h(Z,t)}{g(Z,t)} \Bigg]. \end{aligned} \tag{A.49}$$

Adding the mass constrain will not reduce the δf variance by a great deal, therefore, simultaneous optimization of α and β is mostly not worth the effort and it is fine to work with the ready δ-weights and then enforce the optimization. It is recommended to also calculate the correlation coefficient, and set β to zero when the correlation coefficient is small, relative to machine precision. In a MATLAB implementation we use 10^{-14}. But this, of course, again depends on the application, therefore, we present these two options. The bias introduced by estimating the optimization coefficients, can for sure be neglected [65],[28]. Everything above reduces the error on the constraint in the estimators, which [227] shows.

L^p-norm conservation for δf

In the full f version, the L^p norms are conserved by default, as the weights remain constant.

$$\iint_\Omega f(x,v,t)^p \,\mathrm{d}x\mathrm{d}v = \mathbb{E}\left[\frac{f(Z(t),t)^p}{g(Z(t),t)}\right] = \text{const. for all } t \geq 0 \tag{A.50}$$

Suppose $p \geq 2$. For $\delta f = f - \alpha h$ the density can be rewritten such that

$$\iint_\Omega \delta f(x,v,t)^p \,\mathrm{d}x\mathrm{d}v = \iint_\Omega f(x,v,t)^p + \sum_{n=1}^{p-1} f(x,v,t)^n \left(-\alpha h(x,v,t)\right)^{p-n} + \left(-\alpha h(x,v,t)\right)^p \,\mathrm{d}x\mathrm{d}v. \tag{A.51}$$

On our way to designing a control variate to enforce the L^p-norm conservation, we substitute the integrands in eqn. (A.51) by expected values that will later be approximated by the particles.

$$\begin{aligned}\mathbb{E}\left[\frac{(f(Z,t)-\alpha h(Z,t))^p}{g(Z,t)}\right] - \sum_{n=1}^{p-1}\binom{p}{n}\mathbb{E}\left[\frac{f(Z,t)^n\left(-\alpha h(Z,t)\right)^{p-n}}{g(Z,t)}\right] \\ = \iint_\Omega f(x,v,t)^p \,\mathrm{d}x\mathrm{d}v + \iint_\Omega \left(-\alpha h(x,v,t)\right)^p \,\mathrm{d}x\mathrm{d}v\end{aligned} \tag{A.52}$$

The L^p-norm of f is a conserved quantity of the Vlasov equation. (2.1). Since h is available in analytic form, its L^p-norm is also at hand. Now, as before, we translate the constraint (A.52) to a control variate

$$\begin{aligned}B_n^{**}(t) := \psi_n(X(t))\frac{f(Z,t)-\alpha h(Z,t)}{g(Z,t)} \\ - \beta\left[\frac{(f(Z,t)-\alpha h(Z,t))^p}{g(Z,t)} - \sum_{n=1}^{p-1}\binom{p}{n}\frac{f(Z,t)^n\left(-\alpha h(Z,t)\right)^{p-n}}{g(Z,t)}\right] \\ + \beta\left[\iint_\Omega f(x,v,t)^p \,\mathrm{d}x\mathrm{d}v + \iint_\Omega \left(-\alpha h(x,v,t)\right)^p \,\mathrm{d}x\mathrm{d}v\right],\end{aligned} \tag{A.53}$$

which for $p = 2$ reduces to

$$\begin{aligned}B_n^{**}(t) := \psi_n(X(t))\frac{f(Z,t)-\alpha h(Z,t)}{g(Z,t)} \\ - \beta\left[\frac{\left(f(Z,t)-\alpha h(Z,t)\right)^2}{g(Z,t)} + 2\alpha\frac{f(Z,t)h(Z,t)}{g(Z,t)}\right] \\ + \beta\left[\iint_\Omega f(x,v,t)^2 \,\mathrm{d}x\mathrm{d}v + \alpha^2\iint_\Omega h(x,v,t)^2 \,\mathrm{d}x\mathrm{d}v\right].\end{aligned} \tag{A.54}$$

Appendix B.

Vlasov models and geometries

B.1. Systems and parameters

In order to perform simulations within a physics context we introduce the Vlasov–Maxwell and Vlasov–Poisson system along with some standard physical effects for single and multiple species. More simplified models include a guiding center and a drift kinetic model stemming partially from gyrokinetic theory in the zero Larmor radius case.

B.1.1. Multi-species Vlasov–Maxwell and Vlasov–Poisson

Before treating multiple species, we have to acquire different test cases and scenarios which are closer to the physics in nature. An example of how a physicist may describe a simulation is: “we take thermal ions and electrons considering collisions by electrostatic space charge kicks and look at the electron scale”. In this case the initial condition “thermal” describes a Gaussian, “space charge” refers to the self consistent electric field“ and the ”electron scale“ only means that the ions are treated as fixed. The translation is ”one species electron Vlasov–Poisson with constant ion background and Gaussian/Maxwellian initial condition“. So the Poisson equation models the ”kicks“ by the species itself. From a mathematical point of view this is a distorted picture, and mostly you will not encounter a set of equations and an initial condition, but rather vague descriptions. Nevertheless with some basic knowledge it is quite simple to ”translate“ such descriptions into something easy accessible.
We define some physical quantities, which we use later in the normalization where we interpret everything relative to electrons. Not all quantities are actually needed, some cancel out or are being replaced by a constant relative to something depending on the electrons. The thermal velocity for a species s (electrons e or ions i) is denoted by

$$v_{th,s} := \sqrt{k_B \frac{T_s}{m_s}}, \tag{B.1}$$

where T_s is the temperature, m_s the mass and $k_B = 1.3806485210^{-23} JK^{-1}$ the Boltzmann's constant. Here the first pitfall is that the temperature in plasma physics is often given in electron Volts (eV), which is not a temperature but an energy. Division by the constant $\frac{k_B}{e} = 8.617330310^{-5} eVK^{-1}$ yields the desired temperature in Kelvin. For a temperature $\hat{T}_s$ given in eV it holds that $v_{th,s} = \sqrt{\frac{\hat{T}_s e}{m_s}}$. Given temperature and mass ratio between a species s and the electrons e results in the following useful ratio of thermal velocities.

$$\frac{v_{th,s}}{v_{th,e}} = \sqrt{\frac{T_s}{T_e}\frac{m_e}{m_s}} \tag{B.2}$$

s	species	e, i
T_s	temperature	K
m_s	mass	kg
e	unit charge	C
q_s	charge	C
n_s	density of species (constant)	m^{-d}

The (electron) plasma frequency defines the characteristic time scale by ω_p^{-1}

$$\omega_p := \omega_{pe} = \sqrt{\frac{n_e e^2}{m_e \epsilon_0}}. \tag{B.3}$$

This frequency is defined for an arbitrary species s with charge as

$$\omega_{ps} := \sqrt{\frac{n_s e^2}{m_s \epsilon_0}}. \tag{B.4}$$

The electron Debye length reads

$$\lambda_D := \sqrt{\frac{\epsilon_0 k_B T_e}{n_e e}} = \frac{v_{th,e}}{\omega_p}. \tag{B.5}$$

Note that we chose ϵ_0 and μ_0 for vacuum, but in principle any other nonlinear operator is also possible.
When working with a magnetic field one often stumbles upon the dimensionless quantity β, which is the ratio between the pressure of a species s at a certain temperature and the magnetic pressure for a given magnetic field strength B.

$$\beta_s = \frac{p_s}{p_{magnetic}} = \frac{n_s k_B T_s}{\frac{B^2}{2\mu_0}} = \frac{2 n_s k_B T_s}{B^2 c^2 \epsilon_0} \tag{B.6}$$

Since in most cases β is given for the electrons, the strength of the magnetic field B can be obtained from β directly with respect to the normalization we chose here yielding the normalized field strength $\tilde{B}$. Another option to describe the magnetic field strength is the *cyclotron frequency* $\omega_{c,s}$ for a species s.

$$\omega_{c,s} := \frac{|q_s| B}{m_s}, \quad \omega_{ce} := \frac{eB}{m_e} \tag{B.7}$$

The radial gyromotion of the particles around a field line has frequency $\omega_{c,s}$, thus, it is also called the gyro frequency. The electron cyclotron frequency is commonly denoted as ω_{ce}. Note that the factor $\frac{v_{th,e}}{c}$ also appears in normalized Maxwell equations.

$$B = \sqrt{\frac{2 n_s k_B T_s}{\beta c^2 \epsilon_0}} = \left(\frac{v_{th,e}}{c} \sqrt{\frac{2}{\beta}} \right) \left(\omega_p \frac{m_e}{e} \right) \Rightarrow \tilde{B} = \frac{v_{th,e}}{c} \sqrt{\frac{2}{\beta}}. \tag{B.8}$$

Sometimes B is denoted by B_0 and then oscillations or profiles are given with respect to B_0 originating from a fixed β. We continue with an example for nondimensionalization in the case of the Poisson equation.

$$\nabla \cdot E(x,t) = \frac{1}{\epsilon_0} \sum_s q_s \int_{\mathbb{R}^d} f(x,v) \, \mathrm{d}v \tag{B.9}$$

		normalization	SI unit
dimension	d	-	1
time	t	$\frac{1}{\omega_p}$	s
length	x	λ_D	m
velocity	v	$v_{th,e}$	$\frac{m}{s}$
number density	n_s	n_e	m^{-d}
phase space density	f_s	$\frac{n_e}{(v_{th,e})^d}$	$\left(s \cdot m^{-2}\right)^d$
charge density	ρ_s	$e\ n_e$	$A \cdot s \cdot m^{-d}$
current density	j_s	$e\ n_e\ v_{th,e}$	$A \cdot m^{-(d-1)}$
electric field	E	$\frac{m_e}{e} v_{th,e} \omega_p$	$V \cdot m^{-1}$
magnetic field	B	$\frac{m_e}{e} \omega_p$	T
electric potential	Φ	$\frac{m_e}{e} (v_{th,e})^2$	V
charge	q	e	$A \cdot s$
mass	m	m_e	kg

Table B.1.: Normalization of commonly used quantities with respect to the electron denoted by $[\ ]_e$.

We make the substitutions $\tilde{t} = t\omega_p$, $\tilde{v} = \frac{v}{v_{th,e}}$ and $\tilde{x} = \frac{x}{\lambda_D}$ defining the dimensionless electric field $\tilde{E}$. We know the unknown normalization constant C already according to table B.1, but we want to derive it again. Therefore, we make the following Ansatz for the dimensionless the electric field $\tilde{E}$:

$$\tilde{E}(\tilde{x}, \tilde{t}) \cdot C = E(\tilde{x}\lambda_D, \tilde{v}\ v_{th,e}). \tag{B.10}$$

Inserting the known $\frac{n_e}{(v_{th,e})^d} \tilde{f}(\tilde{x}, \tilde{v}) f\left(\tilde{x}\lambda_D, \tilde{v}\ v_{th,e}, \frac{\tilde{t}}{\omega_p}\right) = f(x, v, t)$ for the phase space density the substitution reads

$$\tilde{\nabla} \cdot \tilde{E}(\tilde{x}, \tilde{t}) \frac{1}{\lambda_D} C = \frac{1}{\epsilon_0} \sum_s q_s \int_{\mathbb{R}^d} \frac{n_e}{(v_{th,e})^d} \tilde{f}(\tilde{x}, \tilde{v})\ (v_{th,e})^d \mathrm{d}\tilde{v}. \tag{B.11}$$

We non-dimensionalize the charge by $q_s = e\frac{q_s}{e}$ and bring all quantities with dimension to the left hand side.

$$\tilde{\nabla} \cdot \tilde{E}(\tilde{x}, \tilde{t}) \frac{C\epsilon_0}{\lambda_D n_e e} C = \sum_s \frac{q_s}{e} \int_{\mathbb{R}^d} \tilde{f}(\tilde{x}, \tilde{v})\ \mathrm{d}\tilde{v}. \tag{B.12}$$

Now, the right hand side is non-dimensional and therefore, C has to be chosen such that the left hand side is also non-dimensional, yielding $\frac{C\epsilon_0}{\lambda_D n_e e} = 1$. It is useful to express $\epsilon_0 = \frac{n_e e^2}{\omega_p^2 m_e}$. We would like to express everything normalized to the electron scale which means The same result is obtained as in table B.1.

$$C = \frac{\lambda_D n_e e}{\epsilon_0} = v_{th,e} \frac{n_e e}{\epsilon_0 \omega_p} = v_{th,e} \frac{n_e e \omega_p^2 m_e}{n_e e^2 \omega_p} = \frac{m_e}{e} v_{th,e} \omega_p. \tag{B.13}$$

With the above definitions we state our typical equations relative to electrons. This allows us to take parameters relative to the *electron scale* into account, where they are available or normalize values from the real world.
For a species s in d dimensions the typical initial condition is a Gaussian velocity distribution (also called Maxwellian).

$$f_s(x, v, t = 0) = \frac{n_s}{n_e} \left(\sqrt{2\pi}\ \frac{v_{th,s}}{v_{th,e}}\right)^{-d} \exp\left(-\frac{\|v\|^2}{2\left(\frac{v_{th,s}}{v_{th,e}}\right)^2}\right) \tag{B.14}$$

Here n_s and n_e denote the mean density and, therefore, a perturbation can be added on top of this. We obtain the normalized Vlasov equation (B.15) and the characteristics (B.16) for a species s relative to electrons.

$$\partial_t f_s(x,v,t) + v \cdot \nabla_x f_s(x,v,t) + \frac{q_s}{e}\frac{m_e}{m_s}\left[E(x,t) + v \times B(x,t)\right] \cdot \nabla_v f_s(x,v,t) = 0 \tag{B.15}$$

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} V_s(t) &= \frac{q_s}{e}\frac{m_e}{m_s}\left[E(X_s(t),t) + V_s(t) \times B(X_s(t),t)\right] \\ \frac{\mathrm{d}}{\mathrm{d}t} X_s(t) &= V_s(t) \end{aligned} \tag{B.16}$$

For the Maxwell equations (B.17)-(B.20) the vacuum permeability can be expressed as $\mu_0\epsilon_0 = \frac{1}{c^2}$, which leaves the speed of light as the only natural constant in the equations. We can normalize c with respect to the thermal electron velocity, but $\tilde{c}$ will remain a very large quantity.

$$\partial_t E(x,t) = \underbrace{\left(\frac{c}{v_{th,e}}\right)^2}_{=:\tilde{c}^2} \nabla \times B(x,t) - \sum_s \frac{q_s}{e}\int_{\mathbb{R}^d} v f_s(x,v,t)\,\mathrm{d}v \qquad \text{Ampère} \tag{B.17}$$

$$\nabla \times E(x,t) = -\partial_t B(x,t) \qquad \text{Faraday} \tag{B.18}$$

$$\nabla \cdot E(x,t) = \sum_s \frac{q_s}{e}\int f_s(x,v,t)\,\mathrm{d}v \qquad \text{(electrostatic) Gauss} \tag{B.19}$$

$$\nabla \cdot B(x,t) = 0 \qquad \text{magnetic Gauss} \tag{B.20}$$

The speed of light is normalized to the electron thermal velocity which reads $\tilde{c} := \frac{c}{v_{th}}$. In many cases for the sake of simplicity the speed of light is artificially set to one, $\tilde{c} = 1$.
We continue with simplified models derived from the electromagnetic Vlasov–Maxwell system. Suppose the magnetic field is constant in time, the electric field E is obtained by solving Faraday's and Gauss' law which read

$$\nabla \times E(x,t) = 0 \text{ and } \nabla \cdot E(x,t) = \sum_s \underbrace{\frac{q_s}{e}\int f_s(x,v,t)}_{=\rho_s(x,t)}. \tag{B.21}$$

Some elementary math tells us that the curl of E always vanishes when E is the gradient of a scalar field, which then turns out to be the electric potential Φ such that

$$E(x,t) = -\nabla\Phi(x,t) \Rightarrow \nabla \times E(x,t) = \nabla \times (-\nabla\Phi(x,t)) = 0. \tag{B.22}$$

Since the Faraday equation is now satisfied, inserting the electric potential into Gauss' law yields the Poisson equation:

$$E(x,t) = (-\nabla\Phi(x,t)) = -\Delta\Phi(x,t) = \sum_s \rho_s(x,t). \tag{B.23}$$

This means for a purely electrostatic model the electric field E is obtained from the Poisson equation (B.24).

$$-\Delta\Phi(x,t) = \sum_s \frac{q_s}{e}\int_{\mathbb{R}^d} f_s(x,v,t)\,\mathrm{d}v, \quad E(x,t) = -\nabla\Phi(x,t) \tag{B.24}$$

For the initialization of an Vlasov–Maxwell solver at $t = 0$ the electric field is also obtained by the Poisson eqn. (B.24). A suitable discretization conserves the electrostatic and magnetic Gauss laws, which can be verified at the end of a simulation and is often referred to as the Poisson error although eqn. (B.24) is not valid over time, but Gauss law is.
Depending on the choice of the ion background Vlasov–Poisson is equivalent or at least very similar to Vlasov-Ampère, where the Poisson equation is replaced by the Ampère equation (B.25).

$$\partial_t E(x,t) = -\sum_s \frac{q_s}{e} \int_{\mathbb{R}^d} v f_s(x,v,t) \, \mathrm{d}v \tag{B.25}$$

If we chose to simulate only electrons and set $s = e$ then the only natural constant appearing in the Vlasov–Maxwell system is the speed of light c in vacuum.
Now, what is actually the mathematicians favorite model? First we set the only natural constant to one $\tilde{c} = 1$! For the standard initial condition (B.14), we do not need to specify n_e because everything is normalized with respect to the electrons, and the only choice is to set $v_{th} = 1$. For the one species model, one simply neglects the Vlasov equation for the ions f_i and defines mostly the standard initial condition (B.14) with $n_i = n_e$ and $v_{th,i} = v_{th,e} = 1$. The mass m_i is not needed, but the charge $q_i = -e$. Thus the simulation will be valid for electrons in a universe with $c = 1$ for different temperatures T_e and densities n_e.
Note that the total energy $\mathcal{H}$ in the Vlasov–Maxwell system consisting of the kinetic energy $\mathcal{H}_p$, electric energy $\mathcal{H}_E$ and magnetic energy $\mathcal{H}_B$ is conserved with the energies defined as

$$\begin{aligned}
\mathcal{H} &= \mathcal{H}_p + \mathcal{H}_E + \mathcal{H}_B, \\
\mathcal{H}_p &= \sum_s \frac{1}{2} m_s \iint |v|^2 f_s(x,v,t) \, \mathrm{d}x\mathrm{d}v, \\
\mathcal{H}_B &= \frac{1}{2} \int |B(x,t)|^2 \, \mathrm{d}x, \\
\mathcal{H}_E &= \frac{1}{2} \int |E(x,t)|^2 \, \mathrm{d}x.
\end{aligned} \tag{B.26}$$

The total momentum

$$\mathcal{P} = \int E(x,t) \times B(x,t) \, \mathrm{d}x + \underbrace{\sum_s \iint m_s v f_s(x,v,t) \, \mathrm{d}x\mathrm{d}v}_{\text{particle momentum}} \tag{B.27}$$

is also among the conserved physical quantities.

Linearized Vlasov–Maxwell

Sometimes the non-linear dynamics is too difficult to resolve such that one would like to start with a simpler, linearized model. Here the species index s is only dropped for the introduction. First a time independent state $f^0(x,v)$ as a equilibrium solution to the Vlasov–Maxwell equations (B.15),(B.17)-(B.20) is chosen. In one dimension this is typically a Maxwellian or just the spatially unperturbed initial condition, thus, it is often denoted as $f^0(v)$ without any spatial dependence. However, in higher dimensional problems the temperature and drift - variance and mean - often depend on the position. In order to keep the generality, the spatial dependence is allowed in the equilibrium state $f^0(x,v)$. We then can approximate the true solution f by an asymptotic expansion around the equilibrium state f^0 for a small $\epsilon > 0$ in eqn. (B.28).

$$f(x,v,t) = f^0(x,v) + \epsilon f^1(x,v,t) \tag{B.28}$$

The time dependent $f^1(x,v,t)$ contains the difference between the equilibrium f^0 and the full solution f and is, therefore, often referred to as $\delta f(x,v,t)$. This can cause confusion with the control variate δf method for variance reduction, such that we keep the notation f^1 here. Again, variance reduction does not equal linearization. Since one also have to collect coefficients of powers of ϵ for the Maxwell equations, expansions for the electric and magnetic field are introduced in the same manner in eqn. (B.29).

$$E(x,t) = E^0(x) + \epsilon E^1(x,t), \qquad B(x,t) = B^0(x) + \epsilon B^1(x,t) \tag{B.29}$$

Inserting eqns. (B.28),(B.29) into the unnormalized Vlasov equation and ordering the coefficients yields

$$\begin{aligned}\partial_t f^0(x,v) + \underbrace{v\cdot\nabla_x f^0(x,v,t) + \frac{q}{m}\left[E^0(x) + v\times B^0(x)\right]\cdot\nabla_v f^0(x,v)}_{=0} \\ + \epsilon\Big\{\partial_t f^1(x,v,t) + v\cdot\nabla_x f^1(x,v,t) \\ + \frac{q}{m}\left[E^0(x) + v\times B^0(x)\right]\cdot\nabla_v f^1(x,v,t) + \frac{q}{m}\left[E^1(x,t) + v\times B^1(x,t)\right]\cdot\nabla_v f^0(x,v)\Big\} \\ + \epsilon^2\left[\frac{q}{m}\left[E^1(x,t) + v\times B^1(x,t)\right]\cdot\nabla_v f^1(x,v,t)\right] = 0.\end{aligned} \tag{B.30}$$

Note that $\partial_t f^0(x,v) = 0$ and that the nonlinear interaction between the f^1 and the field E^1, B^1 are in the ϵ^2 term which is going to be neglected. Since the Maxwell equations are linear in (E,B,f) (superposition principle), equating coefficients yields a set of Maxwell equations for the pair (f^0,E^0,B^0) and a separate one for (f^1,E^1,B^1). The first pair is already solved as condition to the equilibrium f_0 such that only the second one is left. Thus, the linearization does not affect the Maxwell solver.
Reintroducing multiple species and the correct normalization yields the linearized Vlasov equation (B.31) and the corresponding Maxwell equations (B.32).

$$\begin{aligned}&\partial_t f_s^1(x,v,t) + v\cdot\nabla_x f_s^1(x,v,t) \\ &+ \frac{q_s}{e}\frac{m_e}{m_s}\Big\{\left[E^0(x) + v\times B^0(x)\right]\cdot\nabla_v f_s^1(x,v,t) + \left[E^1(x,t) + v\times B^1(x,t)\right]\cdot\nabla_v f_s^0(x,v)\Big\} = 0\end{aligned} \tag{B.31}$$

$$\begin{aligned}\partial_t E^1(x,t) &= \left(\frac{c}{v_{th,e}}\right)^2 \nabla\times B^1(x,t) - \sum_s \frac{q_s}{e}\int_{\mathbb{R}^d} v f_s^1(x,v,t)\ \mathrm{d}v \\ \nabla\times E^1(x,t) &= -\partial_t B^1(x,t) \\ \nabla\cdot E^1(x,t) &= \sum_s \frac{q_s}{e}\int f_s^1(x,v,t)\ \mathrm{d}v \\ \nabla\cdot B^1(x,t) &= 0\end{aligned} \tag{B.32}$$

It is problematic that eqn. (B.31) cannot be solved anymore with the method of characteristics. Therefore, a weight for every characteristic is introduced by the likelihood $F(t) := f(X(t),V(t),t)$. The equations of motion for each species s are then given in eqn. (B.33).

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}X(t) &= V(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}V_s(t) &= \frac{q_s}{e}\frac{m_e}{m_s}\left[E^0(X_s(t)) + V_s(t)\times B^0(X_s(t))\right] \\ \frac{\mathrm{d}}{\mathrm{d}t}F_s(t) &= -\frac{q_s}{e}\frac{m_e}{m_s}\left[E^1(X_s(t)) + V_s(t)\times B^1(X_s(t))\right]\end{aligned} \tag{B.33}$$

Plasma waves

For code verification - did we implement the algorithm correctly? - one is often faced with the problem of finding suitable test-cases. The standard approach consists of a perturbation analysis of the linearized system which gives some results to compare with. Instead of just citing some scenarios, this section should teach the non-physicist reader how to identify suitable scenarios from the available literature. Here kinetic equations are solved, yet MHD theory seems to be much more evolved such that we chose to translate MHD scenarios to kinetic test-cases by multiplying a Maxwellian velocity distribution with the MHD density. We start with the first phenomenon in a plasma which is a plasma wave. MHD waves can be found in [228], but most kinetic theory should be taken from [229, 109]. The latter one can be hard to read, thus, it is advised to consult [229] first.
A plasma wave with velocity V in direction of the wave vector $\vec{k}$ in a periodic box of the length L passes the box in one period of time $T = \frac{L}{V}$. Therefore, in most cases the length of the box is set to $L = \frac{2\pi}{k} = \frac{2\pi}{\vec{k}}$ in order for the frequency to be easily determined from the wave vector.

$$\omega = 2\pi f = \frac{2\pi}{T} = V \cdot k \Leftrightarrow \frac{\omega}{k} = V \tag{B.34}$$

In some relations, the frequency ω has an imaginary part such that a description of the wave as $t \mapsto A_0 \exp(\mathrm{i}\omega t)$ yields an oscillation with the real part of ω and a damping of the initial wave amplitude A_0 over time. Note also that the wave vector $\vec{k}$ is in one dimensional models only denoted by k. Depending on the reduced system there are only some wave vectors that actually make sense, thus if not noted otherwise we suppose that $\vec{k}$ points along a unit vector.
Plasma oscillations - Langmuir waves ($\vec{k} \parallel \vec{E}$)
Until now, we repeatedly treated the simple Landau damping test-case which acts only on the electrons, thus requiring only a neutralizing background. It is electrostatic, hence there is no constant magnetic field $\vec{B}_0 = 0$. It can also travel along the magnetic field which means the wave vector is parallel to the magnetic background field $\vec{k} \parallel \vec{B}_0$. The Bohm-Gross relation reads

$$\omega = \sqrt{\omega_{p,e} + 3kv_{th,e}}. \tag{B.35}$$

Much more accurate roots are found by numerical resolution of the dispersion relation for the Vlasov–Poisson system.
Ion acoustic wave ($\vec{k} \parallel \vec{E}$)
This starts out as a two species (ions and electrons) Vlasov–Poisson test case that can, of course, also be used in Vlasov–Maxwell. The ion acoustic wave is a plane plasma wave, where a perturbation of the ions leads to a wave traveling at the sound speed c_s, given in the approximated dispersion relation, see eqn. (B.36), which we obtained from [228][p.458], where also "electron acoustic" waves are described. The factor $\gamma = 1 + \frac{2}{d}$ describes the dimensionality of the wave, see [228][p.454] and is here $\gamma = 3$. In this case we suppose that the electrons are much hotter than the ions $T_e \gg T_i$, see [109].

$$\begin{aligned} V_s = \frac{\omega}{k} = \sqrt{\frac{k_B(T_e + \gamma T_i)}{m_i}} = v_{th,e}\sqrt{\left(\frac{m_e}{m_i} + \gamma\left(\frac{v_{th,i}}{v_{th,e}}\right)^2\right)} = v_{th,e}\sqrt{\frac{m_e}{m_i}\left(1 + \gamma\frac{T_i}{T_e}\right)} \\ \text{for } \frac{V_s}{k}\omega \ll \omega_{pi}\left(1 + \frac{T_i}{T_e}\right) = \omega_p\sqrt{\frac{m_e}{m_i}\frac{n_i}{n_e}}\left(1 + \left(\frac{v_{th,i}}{v_{th,e}}\right)^2\right) \end{aligned} \tag{B.36}$$

Here eqn. (B.36) is only valid for low frequencies. We adapt the test case described in [230] and refer to the similar tests in [158, 231].
An electromagnetic wave - also called light wave ($B_0 = 0, \vec{E} \perp \vec{B}$)

The first real electromagnetic scenario is the simplest electromagnetic wave, also called a light wave or a photon. Without background field a weak initial excitation of the magnetic field yields an electromagnetic wave, which is characterized by a phase shifted oscillation of the magnetic and electric field. This corresponds to light traveling through a medium and, therefore, being slightly slower than in vacuum. Although this is a rather simple phenomenon, experimentalists use microwave diagnostics for comparison to theoretical predictions [232] and even for tomography of the velocity distribution, see [233]. Therefore, one has to be able to simulate also the effects the plasma has on external electromagnetic waves in order to compare experiment to theory. In the following only electrons are required such that we do not need the ion time scale.

$$\begin{aligned} \omega &= \sqrt{\omega_{p,e} + k^2 c^2} \\ \tilde{\omega} &= \sqrt{1 + \frac{\tilde{k}^2 \lambda_D^2 \tilde{c}^2}{v_{th,e}^2 \omega_{p,e}^2}} = \sqrt{1 + \tilde{k}^2 \tilde{c}^2} \end{aligned} \tag{B.37}$$

Since this wave has a higher frequency ω then the plasma frequency ω_p, the simulation time should be chosen short. The speed of light should be significantly greater than the thermal velocity of the electrons $c \gg v_{th,e}$ implying $\tilde{c} \gg 1$, e.g. we chose $\tilde{c} = 10$. The frequency ω should then be larger than the plasma frequency such that we can observe the wave. A very large ω only tests the Maxwell solver, but here we want to measure the slower speed of light, hence we set $\omega = 3$ resulting in $k = 0.4$. As a test the frequency ω or the 90 degree phase shift between E and B can be measured.

The exotic electron X-Wave ($\vec{k} \perp \vec{B}_0,\ \vec{k} \parallel \vec{E}$)

We continue with an electron wave requiring electromagnetic effects but only an ion background. Thus, the dispersion relation comes from cold plasma theory and the ions should constitute a constant background. A detailed treatment of these wave phenomenons in a Vlasov–Maxwell plasma by a completely spectral and, thus, highly accurate discretization can be found in [207, 113]. The frequency for electrons is obtained by

$$\begin{aligned} \frac{c^2 k^2}{\omega^2} &= 1 - \frac{\omega_{p,e}^2(\omega^2 - \omega_{p,e}^2)}{\omega^2(\omega^2 - \omega_{p,e}^2 - \omega_{c,e}^2)} \\ \frac{\tilde{c}^2 \tilde{k}^2}{\tilde{\omega}^2} &= 1 - \frac{\tilde{\omega}^2 - 1}{\tilde{\omega}^2(\tilde{\omega}^2 - \tilde{\omega}_{c,e}^2 - 1)}. \end{aligned} \tag{B.38}$$

The electron X-wave dispersion relation (B.38) admits two solutions, the slow and the fast X-wave. The solution in normalized form is given in eqn. (B.39).

$$\omega^2 = \frac{2(c^2 k^2 (1 + \omega_{ce}^2) + 1))}{\sqrt{(c^2 k^2 - \omega_{ce}^2)^2 + 4\omega_{ce}^2} + \omega_{ce}^2 + c^2 k^2 + 2} \tag{B.39}$$

A test-case for the nonlinear $X - B$ mode conversion from exotic waves to Bernstein waves for non-periodic 1d2v Vlasov–Maxwell with electrons can be found in [234].

Magnetoacoustic (magnetosonic) wave - the compressional Alfvén wave ($\vec{k} \perp \vec{B}_0,\ \vec{k} \parallel \vec{E}$)

The Alfvén wave is a transverse wave traveling along the magnetic field $\vec{k} \parallel$ with the Alfvén velocity V_A.

$$\begin{aligned} V_A &= \frac{B_0}{\mu_0 \sum_s m_s n_s} \\ \tilde{V}_A &= \frac{c}{v_{th,e}} \tilde{B}_0 \sqrt{\frac{m_e n_e}{\sum_s m_s n_s}} = \tilde{c}\ \tilde{B}_0\ \sqrt{\frac{m_e n_e}{\sum_s m_s n_s}} \end{aligned} \tag{B.40}$$

Yet in the 1d2v Vlasov–Maxwell example only one dimensional longitudinal waves are possible such as the ion acoustic wave. The ion acoustic wave perpendicular to a nonzero background magnetic field $\vec{k} \perp B_0$ is called the magnetoacoustic wave. Since it is a longitudinal sound wave compressing and decompressing the ions, it is also referred to as the compressional Alfvén wave. The dispersion relation obtained from the cold plasma fluid description reads

$$\omega = ck\sqrt{\frac{V_s + V_A}{c^2 + V_A^2}}, \tag{B.41}$$

where V_A is the Alfvén velocity and V_s the sound speed known from the electrostatic ion acoustic wave before. The wave is called magnetoacoustic for $\omega < \omega_{c,i}$ and the fast Alfvén wave for $\omega_{c,i} < \omega < \omega_{c,e}$.

B.1.2. Vlasov–Maxwell in 1d2v

We consider a reduction of the full Vlasov–Maxwell model onto one spatial and two velocity components. Elimination of the second and third spatial component, leaves us with two components of the electric field and one component of the magnetic field. Here the single magnetic component in z-direction is denoted by B.

$$x = x_1,\ v = (v_1, v_2),\ E = (E_1, E_2),\ B = B_3 \tag{B.42}$$

For a density $f(x, v_1, v_2, t)$, the two components of the electric field $E_1(x,t), E_2(x,t)$ and the magnetic field $B(x,t)$ the reduced Vlasov equation is given in eqn. (B.43).

$$\partial_t f_s + v_1 \partial_x f_s + \frac{q_s}{e}\frac{m_e}{m_s}\left[E_1 \partial_{v_1} f_s + E_2 \partial_{v_2} f_s + B\left(v_2 \partial_{v_1} f_s - v_1 \partial_{v_2} f_s\right)\right] = 0 \tag{B.43}$$

Dropping the species index s yields the corresponding characteristics in eqn. (B.44).

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} V_1(t) &= \frac{q}{e}\frac{m_e}{m}\left[E_1(X_s(t), t) + V_2(t) B(X(t), t)\right] \\
\frac{\mathrm{d}}{\mathrm{d}t} V_2(t) &= \frac{q}{e}\frac{m_e}{m}\left[E_2(X_s(t), t) - V_1(t) B(X(t), t)\right] \\
\frac{\mathrm{d}}{\mathrm{d}t} X(t) &= V_1(t)
\end{aligned} \tag{B.44}$$

The time dependent Maxwell equations reduce to a system of three equations (B.45).

$$\begin{aligned}
\partial_t E_1(x,t) &= -\sum_s \frac{q_s}{e}\int v_1 f_s(x, v_1, v_2, t)\mathrm{d}v \\
\partial_t E_2(x,t) &= -\left(\frac{c}{v_{th,e}}\right)^2 \partial_x B(x,t) - \sum_s \frac{q_s}{e}\int v_2 f_s(x, v_1, v_2, t)\mathrm{d}v, \\
\partial_t B(x,t) &= -\partial_x E_2(x,t)
\end{aligned} \tag{B.45}$$

At the initialization for $t = 0$ the Poisson eqn. (B.46) needs to be solved in order to obtain the first component E_1 of the electric field. The second component is always initialized as zero, $E_2(x, 0) = 0$.

$$-\partial_{xx}\Phi(x,t) = \sum_s \frac{q_s}{e}\int_{\mathbb{R}^d} f_s(x, v_1, v_2, t)\ \mathrm{d}v, \quad E_1(x,t) = -\partial_x \Phi(x,t) \tag{B.46}$$

We consider the Hamiltonian splitting described in [19], which yields three Hamiltonians $\mathcal{H}_{p_1}, \mathcal{H}_{p_2}, \mathcal{H}_B, \mathcal{H}_E$. Later the following discretization is performed by Lagrangian particles such that we do not need the Vlasov equation but its characteristics in the splitting.

- Kinetic energy ($d = 1$), $\mathcal{H}_{p_1} = \frac{1}{2}\int v_1^2 f(x, v, t)\ \mathrm{d}v$

$$\begin{aligned}
\partial_t B(x,t) &= \partial_t E_2(x,t) = 0 \\
\frac{\mathrm{d}}{\mathrm{d}t} V_1(t) &= 0 \\
\frac{\mathrm{d}}{\mathrm{d}t} X(t) &= V_1(t) \\
\frac{\mathrm{d}}{\mathrm{d}t} V_2(t) &= -\frac{q}{e}\frac{m_e}{m} V_1(t) B(X(t), t) \\
\partial_t E_1(x,t) &= -\sum_s \frac{q_s}{e} \int v_1 f_s(x, v_1, v_2, t)\mathrm{d}v
\end{aligned} \tag{B.47}$$

- Kinetic energy ($d = 2$), $\mathcal{H}_{p_v} = \frac{1}{2}\int v_2^2 f(x, v, t)\ \mathrm{d}v$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} X(t) &= \frac{\mathrm{d}}{\mathrm{d}t} V_2(t) = 0 \\
\partial_t B(x,t) &= \partial_t E_1(x,t) = 0 \\
\frac{\mathrm{d}}{\mathrm{d}t} V_1(t) &= \frac{q}{e}\frac{m_e}{m} V_2(t) B(X(t), t) \\
\partial_t E_2(x,t) &= -\sum_s \frac{q_s}{e} \int v_2 f_s(x, v_1, v_2, t)\mathrm{d}v
\end{aligned} \tag{B.48}$$

- Electric energy $\mathcal{H}_E = \frac{1}{2}\int |E(x,t)|^2\ \mathrm{d}x$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} X(t) &= 0 \\
\partial_t E_1(x,t) &= \partial_t E_2(x,t) = 0 \\
\frac{\mathrm{d}}{\mathrm{d}t} V_1(t) &= \frac{q}{e}\frac{m_e}{m} E_1(X_s(t), t) \\
\frac{\mathrm{d}}{\mathrm{d}t} V_2(t) &= \frac{q}{e}\frac{m_e}{m} E_2(X_s(t), t) \\
\partial_t B(x,t) &= -\partial_x E_2(x,t)
\end{aligned} \tag{B.49}$$

- Magnetic energy $\mathcal{H}_B = \frac{1}{2}\int |B(x,t)|^2\ \mathrm{d}x$

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} X(t) &= \frac{\mathrm{d}}{\mathrm{d}t} V_1(t) = \frac{\mathrm{d}}{\mathrm{d}t} V_2(t) = 0 \\
\partial_x B(x,t) &= \partial_x E_1(x,t) = 0 \\
\partial_t E_2(x,t) &= -\left(\frac{c}{v_{th,e}}\right)^2 \partial_x B(x,t)
\end{aligned} \tag{B.50}$$

B.1.3. Drift kinetic and guiding center model

We extend the two dimensional guiding center model to a four dimensional electrostatic drift kinetic model. Here drift kinetic refers to gyrokinetic in the zero Lamor radius limit. For us the gyrokinetic equations as approximation to the Vlasov–Maxwell equations are only valid in the large aspect ratio, because of the $W = 0$ approximation in [25]. It holds true for scenarios with the homogeneous magnetic fields used here. Essentially we solve equations of gyrokinetic type and neglect the gyroaverage. This (3d1v) model problem is spatially three dimensional and regards only the parallel velocity $v_\parallel$. Such a drift kinetic model is treated in many occasions [42, 190, 192, 193] especially because it has a Hamiltonian structure $\partial_t f = \{f, H\}$,

see also [21]. Let $B(x)$ denote a vector field representing a background magnetic field and denote the normalized direction of the magnetic field by the

$$B : \mathbb{R}^3 \to \mathbb{R}^3, \quad b = \frac{B}{\|B\|}, \quad B_0 = \|B\|. \tag{B.51}$$

The fast gyromotion happens perpendicular to the magnetic field, such that a particle travels parallel with velocity $v_\parallel$ along a magnetic field line. In order to describe parallel and orthogonal motions the $\perp$ operator is introduced as a projection into the space orthogonal to the magnetic field. Thus for $v, w \in \mathbb{R}^3$ and the normalized $\|b\| = 1$ the parallel operator $(\ )_\parallel$ is defined by the scalar product as

$$(v)_\parallel = \frac{v \cdot B}{B \cdot B} B = (v \cdot b) b. \tag{B.52}$$

The definition of the perpendicular operator $(\ )_\perp$ is based upon $(\ )_\parallel$.

$$(v)_\perp = v - (v)_\parallel = v - \frac{v \cdot B}{B \cdot B} B = v - (v \cdot b) b = v - (v^t b) b = v - (b b^t) v = b \times v \times b. \tag{B.53}$$

Another useful identity reads

$$(v)_\perp \cdot (w)_\perp = (v)_\perp \cdot w = (v) \cdot w_\perp. \tag{B.54}$$

These definitions can be directly extend to the ∇ operator for a scalar function Φ.

$$\nabla_x^\parallel := \vec{b} \cdot \nabla_x \tag{B.55}$$

$$\nabla_x^\perp := \vec{b} \times \nabla_x \times \vec{b} \tag{B.56}$$

$$\nabla^\perp \Phi := \vec{b} \times \nabla \Phi \times \vec{b} = \nabla \Phi - (\nabla \Phi \cdot \vec{b}) \vec{b} \tag{B.57}$$

The drift kinetic equation for a four dimensional density $f(x, v_\parallel, t)$ with $x \in \mathbb{R}^3$ and $v_\parallel \in \mathbb{R}$ reads

$$\partial_t f(x, v_\parallel, t) - (\nabla_x \Phi(x, t))_\perp \cdot \nabla_x^\perp f(x, v_\parallel, t) + v_\parallel \nabla_x^\parallel \cdot f(x, v_\parallel, t) - \nabla^\parallel \Phi(x, t) \cdot \partial_{v_\parallel} f(x, v_\parallel, t) = 0. \tag{B.58}$$

Note that

$$(\nabla_x \Phi(x, t))_\perp \cdot \nabla_x^\perp f = (\nabla_x \Phi(x, t))_\perp \cdot \nabla_x f = \nabla_x^\perp \Phi(x, t) \cdot \nabla_x f, \tag{B.59}$$

which allows us to rewrite eqn. (B.58) as

$$\partial_t f - \nabla_x^\perp \Phi(x, t) \cdot \nabla_x f + v_\parallel \nabla_x^\parallel \cdot f - \nabla_x^\parallel \Phi(x, t) \cdot \partial_{v_\parallel} f = 0, \tag{B.60}$$

and finally

$$\partial_t f + \left(v_\parallel \cdot \vec{b} - \vec{b} \times \nabla \Phi \times \vec{b} \right) \nabla_x f + \left(\vec{b} \cdot \nabla \Phi \right) \partial_{v_\parallel} f = 0. \tag{B.61}$$

The characteristics of eqn. (B.60) read

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} X(t) &= -\nabla_x^\perp \Phi(X(t), t) + \vec{b} \cdot v_\parallel \\ \frac{\mathrm{d}}{\mathrm{d}t} V_\parallel(t) &= \vec{b} \cdot \nabla \Phi(X(t), t). \end{aligned} \tag{B.62}$$

So far we did not specify any species in the above equations, yet the goal is to simulate the ion time scale. The right hand side of the Poisson equation features contributions from ions and electrons represented by the respective number density $n_s(x)$.

$$-\Delta \Phi(x) = q_i \int_{\mathbb{R}} f_i(x, v_\parallel) \, \mathrm{d}v_\parallel + q_e \int_{\mathbb{R}} f_e(x, v_\parallel) \, \mathrm{d}v_\parallel = q_i n_i(x) + q_e n_e(x) \tag{B.63}$$

Assuming that the electrons respond quickly and follow basically the ion distribution, the electrons do not need their own Vlasov equations but can be entirely modeled by the Boltzmann response. This is often referred to as Boltzmann or adiabatic electrons, or adiabatic electron response, see [235] for more theory. Here n_0 is the background plasma density and T_e the electron temperature. These quantities typically vary only perpendicular to the magnetic field such that n_e can be assumed to be constant along the field lines. Inserting the electron Boltzmann response,

$$n_e(x) = n_0 \mathrm{e}^{-q_e \frac{\Phi(x)}{T(x)}} \approx n_0 \left(1 - q_e \frac{\Phi(x)}{T_e(x)}\right) \tag{B.64}$$

using the approximation $\mathrm{e}^x \approx 1 + x$ for small x, into the Poisson equation yields the Poisson equation with adiabatic electron response

$$-\Delta\Phi(x) + q_e \frac{n_0}{T_e(x)} \Phi(x) = q_i n_i(x) + q_e n_0. \tag{B.65}$$

Sometimes this response shall be restricted onto the flux surfaces, or in general a specific dimension such that we define for $x_3 \in [a, b]$ the average $\bar{\Phi}$ as

$$\bar{\Phi}(x, t) = \frac{1}{b-a} \int_a^b \Phi(x, t) \mathrm{d}x_3. \tag{B.66}$$

In this case we restrict the adiabatic response to the third dimension by

$$-\Delta\Phi(x) + q_e \frac{n_0}{T_e(x)} \left(\Phi(x) - \bar{\Phi}(x)\right) = q_i n_i(x) + q_e n_0. \tag{B.67}$$

This already falls into the domain of physical modeling such that there are different variants available. In general the Boltzmann response term can also be used for the fully kinetic models, which helps in testing implementations. Gyrokinetic theory works in front of a background under the assumption of quasi neutrality such that the Poisson equation for the electric potential vanishes and a much more complicated variant appears: the quasi-neutrality equation. Thus, instead of the Poisson equation, we use the quasi-neutrality equation with an adiabatic electron response restricted onto the third dimension

$$-\nabla_x^\perp \cdot \left[\frac{n_0(x)}{B_0 \Omega_i} \nabla_x^\perp \Phi(x, t)\right] + \frac{e n_0(x)}{T_e(x)} (\Phi(x, t) - \bar{\Phi}(x, t)) = \int f(x, v_\parallel, t) \mathrm{d}v_\parallel - n_0(x). \tag{B.68}$$

This model requires the density and temperature profiles to be constant along the magnetic field

$$\nabla_x^\parallel n_0(x) = 0 \text{ and } \nabla_x^\parallel T_e(x) = 0. \tag{B.69}$$

Reduction of eqn. (B.60) to the plane perpendicular to the magnetic field yields the two dimensional guiding center model identical to the vorticity equation. A more comprehensive form of equation (B.160) is given given in Cartesian coordinates with $f(t, x, y)$, $\Phi(x, y)$ for all $(x, y) \in \tilde{\Omega}$

$$\partial_t f + (\nabla\Phi)_y \partial_x f - (\nabla\Phi)_x \partial_y f = 0, \quad t \in [0, T] \tag{B.70}$$

with electric field $E = (E_x, E_y)^t = -\nabla\Phi$ and the characteristics

$$\frac{\mathrm{d}}{\mathrm{d}t} X(t) = -E_y(X(t), V(t), t) \text{ and } \frac{\mathrm{d}}{\mathrm{d}t} Y(t) = E_x(X(t), V(t), t). \tag{B.71}$$

In Cartesian and polar coordinates the total energy is given as the H^1 seminorm of Φ

$$\begin{aligned} \mathcal{E}(t) &= \iint_{\tilde{\Omega}} |\nabla\Phi(t, x, y)|^2 \mathrm{d}(x, y) \\ &= \int_\Omega r \left|\partial_r \Phi(t, r, \theta)\right|^2 + \frac{1}{r} \left|\partial_\theta \Phi(t, r, \theta)\right|^2 \mathrm{d}r\mathrm{d}\theta \end{aligned} \tag{B.72}$$

and the total mass by

$$\mathcal{M}(t) = \iint_{\tilde{\Omega}} \rho(x,y)\mathrm{d}(x,y) = \iint_{\Omega} \rho(t,r,\theta) r \mathrm{d}r\mathrm{d}\theta. \tag{B.73}$$

Later a formulation of the drift kinetic model in curvilinear coordinates is provided.
The drift kinetic model is quite close to the Vlasov equation, especially if we solve a Poisson equation as in [24]. In order to use a in time multilevel Monte Carlo approach a mapping from full Vlasov to drift kinetic has to be provided. Here the magnetic moment $\mu = \frac{v_\perp^2}{2B_0} = 0 \to v_\perp = 0.$ was neglected such that we project v to $v_\parallel$ by

$$v_\parallel = \mathcal{P}(v) = \left(v \cdot \vec{b}\right) \tag{B.74}$$

and we define a quasi-inverse as

$$v = \mathcal{P}^{-1}(v_\parallel) = v_\parallel \vec{b} \tag{B.75}$$

The characteristics then read

$$\begin{aligned} \frac{d}{dt}X(t) &= \mathcal{P}(V(t)) \cdot \vec{b} + \vec{b} \times \nabla\Phi\left(X(t),t\right) \times \vec{b} \\ \frac{d}{dt}V(t) &= \mathcal{P}^{-1}\left(\vec{b} \cdot \nabla\Phi\right) \end{aligned} \tag{B.76}$$

This allows us to advance full kinetic particles according to the drift kinetic equations of motion.

B.2. Coordinate transformations into curvilinear coordinates

Although the periodic box is a comfortable home for investigations concerning numerical schemes we have to leave this setting behind when challenging real world problems. But since the box is so convenient, it is straightforward to take the real world Ω and transform it back into our box $\tilde{\Omega}$. This is done by a coordinate transformation where we aim to introduce a ready to use framework on a very basic level. Ratnani provides more material linked to plasma physics in the appendix of [236]. A coordinate transformation from logical $\tilde{x} \in \tilde{\Omega}$ to physical $x \in \Omega$ coordinates is defined as a diffeomorphism $T \in \mathcal{C}^1(\tilde{\Omega}, \Omega)$ as

$$T : \tilde{\Omega} \subset \mathbb{R}^n \to \Omega \subset \mathbb{R}^m, \quad \tilde{x} \mapsto x = \left(T_1(\tilde{x}), \ldots, T_n(\tilde{x})\right). \tag{B.77}$$

In n dimensions the derivative of T is denoted by the Jacobi matrix J_T

$$J_T(\tilde{x}) := \mathrm{D}T(\tilde{x}) = \begin{pmatrix} \partial_{\tilde{x}_1} T_1(\tilde{x}) & \ldots & \partial_{\tilde{x}_n} T_1(\tilde{x}) \\ \vdots & & \vdots \\ \partial_{\tilde{x}_1} T_m(\tilde{x}) & \ldots & \partial_{\tilde{x}_n} T_m(\tilde{x}) \end{pmatrix}. \tag{B.78}$$

Since T is a diffeomorphism, it has an inverse T^{-1} with Jacobi matrix $J_{T^{-1}}(x) = \mathrm{D}T^{-1}(x)$. The immediate verification test is then $T(T^{-1}(x)) = x$ and $J_{T^{-1}}(T(\tilde{x})) = J_T^{-1}(\tilde{x})$.

$$\begin{aligned} T^{-1} : \Omega \to \tilde{\Omega}, \quad x \mapsto \tilde{x} &= \left(T_1^{-1}(x), \ldots, T_n^{-1}(x)\right) \\ J_{T^{-1}}(x) := \mathrm{D}T^{-1}(x) &= \begin{pmatrix} \partial_{x_1} T_1^{-1}(x) & \ldots & \partial_{x_n} T_1^{-1}(x) \\ \vdots & & \vdots \\ \partial_{x_1} T_m^{-1}(x) & \ldots & \partial_{x_n} T_m^{-1}(x). \end{pmatrix} = \left(J_T(\tilde{x})\right)^{-1} \end{aligned} \tag{B.79}$$

In the following, scalar functions and vector fields are also transformed. For a scalar integrable function f we define the transformation $\tilde{f}$ yielding the integral equation (B.80) with the change of variables $\mathrm{d}x = \det(J_T(\tilde{x}))\mathrm{d}\tilde{x}$ including the Jacobi determinant $\det(J_T(\tilde{x}))$.

$$\begin{aligned} f : \Omega \subset \mathbb{R}^m \to \mathbb{C},\ x \mapsto f(x) \\ \tilde{f} : \tilde{\Omega} \subset \mathbb{R}^n \to \mathbb{C},\ \tilde{x} \mapsto \tilde{f}(\tilde{x}) = f\left(T(\tilde{x})\right) \\ \int_\Omega f(x)\ \mathrm{d}x = \int_{\tilde{\Omega}} \tilde{f}(\tilde{x})\ |\det(J_T(\tilde{x}))|\mathrm{d}\tilde{x} \end{aligned} \tag{B.80}$$

For a vector field F the natural basis for the transformation $\tilde{F}$ is the *covariant* basis given by the *covariant transform* in eqn. (B.81). It emerges naturally from the multidimensional chain rule for $x = T(\tilde{x})$.

$$\begin{aligned} F : \Omega \subset \mathbb{R}^m \to \mathbb{R}^m,\ x \mapsto F(x) = J_T(\tilde{x})^{-t}\tilde{F}(\tilde{x}) \\ \tilde{F} : \tilde{\Omega} \to \mathbb{R}^m,\ \tilde{x} \mapsto \tilde{F}(\tilde{x}) = F\left(T(\tilde{x})\right) = \tilde{F}(\tilde{x}) \\ \mathrm{D}_{\tilde{x}}\left[F\left(T(\tilde{x})\right)\right] = \mathrm{D}_{\mathrm{x}}F(T(\tilde{x}))J_T(\tilde{x})^t \Leftrightarrow \mathrm{D}_{\mathrm{x}}F(T(\tilde{x})) = J_T(\tilde{x})^{-t}\mathrm{D}_{\tilde{x}}\tilde{F}\left(T(\tilde{x})\right) \end{aligned} \tag{B.81}$$

Note that the same transform is made for a constant vector $v \in \mathbb{R}^m$ by $\tilde{v} = J_T^{-t}(\tilde{x})v$. The *covariant* transformations for the gradient and the curl are given in eqn. (B.82).

$$\begin{aligned} \nabla f(x) = J_T(\tilde{x})^{-t}\tilde{\nabla}\tilde{f}(\tilde{x}) \\ \nabla \times F(x) = \frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\ \tilde{\nabla} \times \tilde{F}(\tilde{x}) \end{aligned} \tag{B.82}$$

For every vector field F the *contravariant* transform $\hat{F}$ provides a natural way to calculate the divergence, see eqn. (B.83)

$$\begin{aligned} F(x) = \frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{F}(\tilde{x}) \\ \nabla \cdot F(x) = \frac{1}{\det(J_T(\tilde{x}))}\tilde{\nabla} \cdot \hat{F}(\tilde{x}) \end{aligned} \tag{B.83}$$

We can transform between the *covariant* $\tilde{F}$ and *contravariant* $\hat{F}$ representation by

$$\tilde{F}(\tilde{x}) = \frac{J_T(\tilde{x})^t J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{F}(\tilde{x}) \text{ and } \hat{F}(\tilde{x}) = \det(J_T(\tilde{x}))J_T(\tilde{x})^{-1}J_T(\tilde{x})^{-t}\tilde{F}(\tilde{x}). \tag{B.84}$$

By use of the backtransform the *curl* for the contravariant basis is found as

$$\nabla \times F(x) = \frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\ \tilde{\nabla} \times \left[\frac{J_T(\tilde{x})^t J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{F}(\tilde{x})\right]. \tag{B.85}$$

This expression is rather hard to evaluate, such that we conclude that depending on which kind of differential operator are being used one should chose between the covariant and contravariant basis.

B.2.1. Vlasov–Maxwell and Poisson

The GEMPIC framework [19] is used for discretization and, therefore, $(E, B) \in H(curl, \Omega) \times H(div, \Omega)$. This setting has to be respected under the change of variables, thus we chose the covariant Piola transform for the electric field E and the contravariant transform for the magnetic field B.

$$\text{covariant} \qquad E(x,t) = J_T(\tilde{x})^{-t}\tilde{E}(\tilde{x},t), \quad E \in H(curl,\Omega),\ \tilde{E} \in H(curl,\tilde{\Omega}) \tag{B.86}$$

$$\text{contravariant} \qquad B(x,t) = \frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{B}(\tilde{x},t), \quad B \in H(div,\Omega),\ \hat{B} \in H(div,\tilde{\Omega}) \tag{B.87}$$

With this change of variable the Faraday eqn. (B.18) can be solved in the strong form and directly with the differential operators in the logical coordinates, see also eqn. (B.81).

$$\begin{aligned} \partial_t B(x,t) &= -\nabla \times E(x,t) \\ \Leftrightarrow \quad \frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\partial_t \hat{B}(\tilde{x},t) &= -\frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\ \tilde{\nabla} \times \tilde{E}(\tilde{x},t) \\ \Leftrightarrow \quad \partial_t \hat{B}(\tilde{x},t) &= -\tilde{\nabla} \times \tilde{E}(\tilde{x},t) \end{aligned} \tag{B.88}$$

We proceed with the weak form of the Ampère equation for a test function $\varphi \in H(curl,\Omega)$ in eqn. (B.89) that we adapted from [19][p.11].

$$\begin{aligned} \int_\Omega \partial_t E(x,t)\cdot\varphi(x)\ \mathrm{d}x = \left(\frac{c}{v_{th,e}}\right)^2 \int_\Omega B(x,t)\cdot\nabla\times\varphi(x)\ \mathrm{d}x \\ -\sum_s \frac{q_s}{e}\int_\Omega \left(\int_{\mathbb{R}^d} v f_s(x,v,t)\mathrm{dv}\right)\cdot\varphi(x)\ \mathrm{d}x \quad \forall\varphi\in H(curl,\Omega) \end{aligned} \tag{B.89}$$

Because of $\partial_t E = -j$ the current density j has to be in the same coordinate system as E, which means $j \in H(curl,\Omega)$. Hence the covariant transform is used for the current density $j(x,t)$ and the velocity $v = J_T(\tilde{x})^{-t}\tilde{v}$. The basis transform of the velocity $\tilde{v} = J_T(\tilde{x})v$ has to incorporate the change of variables $\mathrm{d}v = \det\left(J_T(\tilde{x})^{-t}\right)\mathrm{d}\tilde{v} = \frac{\mathrm{d}\tilde{v}}{\det(J_T(\tilde{x}))}$. The charge density ρ is defined as $\rho \in L^2(\Omega)$. Since $\varphi \in H(curl,\Omega)$, $\tilde{\varphi}$ is transformed with the covariant transform. We define $\tilde{f}_s(\tilde{x},\tilde{v},t) := f(T(\tilde{x}),v,t)$.

$$\begin{aligned} &\int_{\tilde{\Omega}} J_T(\tilde{x})^{-t}\partial_t\tilde{E}(\tilde{x},t)\cdot J_T(\tilde{x})^{-t}\tilde{\varphi}(\tilde{x})\ |\det(J_T(\tilde{x}))|\mathrm{d}\tilde{x} = \\ &\qquad \left(\frac{c}{v_{th,e}}\right)^2 \int_{\tilde{\Omega}} \frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{B}(\tilde{x},t)\cdot\frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\tilde{\nabla}\times\tilde{\varphi}(\tilde{x})\ |\det(J_T(\tilde{x}))|\mathrm{d}\tilde{x} \\ &-\sum_s \frac{q_s}{e}\int_{\tilde{\Omega}}\left(\int_{\mathbb{R}^n} J_T^{-t}(\tilde{x})\tilde{v}\tilde{f}_s(\tilde{x},\tilde{v},t)\frac{\mathrm{d}\tilde{v}}{\det\left(J_T(\tilde{x})\right)}\right)\cdot J_T(\tilde{x})^{-t}\tilde{\varphi}(\tilde{x})\,|\det(J_T(\tilde{x}))|\mathrm{d}\tilde{x} \quad \forall\tilde{\varphi}\in H(curl,\tilde{\Omega}) \end{aligned} \tag{B.90}$$

Rearranging terms in eqn. (B.90) yields eqn. (B.91) where unfortunately the current density contains now a metric.

$$\begin{aligned} &\int_{\tilde{\Omega}} \partial_t\tilde{E}(\tilde{x},t)^t\left[J_T(\tilde{x})^{-1}J_T(\tilde{x})^{-t}\right]\tilde{\varphi}(\tilde{x})\ |\det(J_T(\tilde{x}))|\mathrm{d}\tilde{x} = \\ &\qquad \left(\frac{c}{v_{th,e}}\right)^2 \int_{\tilde{\Omega}} \hat{B}(\tilde{x},t)^t\cdot\frac{J_T(\tilde{x})^tJ_T(\tilde{x})}{|\det(J_T(\tilde{x}))|}\tilde{\nabla}\times\tilde{\varphi}(\tilde{x})\ \mathrm{d}\tilde{x} \\ &-\sum_s \frac{q_s}{e}\int_{\mathbb{R}^n}\int_{\tilde{\Omega}} \tilde{f}_s(\tilde{x},\tilde{v},t)\tilde{v}_s\cdot\left(J_T(\tilde{x})^{-1}J_T(\tilde{x})^{-t}\tilde{\varphi}(\tilde{x})\right)\mathrm{d}\tilde{x}\mathrm{d}\tilde{v} \quad \forall\tilde{\varphi}\in H(curl,\tilde{\Omega}) \end{aligned} \tag{B.91}$$

We chose the transform such that Faraday's law could be solved exactly. In general we have to fall back on the weak form for the Ampère equation, because transforming the Maxwell part of the strong Ampère equation in the given setting yields

$$\partial_t E(x,t) = \frac{J_T(\tilde{x})^tJ_T(\tilde{x})}{\det(J_T(\tilde{x}))}\ \tilde{\nabla}\times\left[\frac{J_T(\tilde{x})^tJ_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{B}(\tilde{x})\right], \tag{B.92}$$

which has not necessarily a closed form.
For Vlasov–Maxwell or Vlasov–Poisson a transformation of the weak Poisson equation is needed. In both cases one chooses covariant transform for E and a scalar test function

$\varphi \in H^1(\Omega)$. The weak form eqn. (B.93) of the Poisson equation (B.24) contains already the potential as a scalar function, therefore, $\Phi \in H^1(\Omega)$ and consequently for the gradient $-\nabla\Phi = E \in H(curl, \Omega)$.

$$\int_\Omega \nabla\Phi(x,t)\cdot\nabla\varphi(x)\ \mathrm{d}x = \sum_s \frac{q_s}{e}\int_{\mathbb{R}^n}\int_\Omega f_s(x,v,t)\varphi(x)\ \mathrm{d}x\mathrm{d}v \quad \forall\varphi\in H^1(\Omega) \tag{B.93}$$

$$E(x,t) = -\nabla\Phi(x,t) \tag{B.94}$$

Since the transformed electric field is given in the covariant basis, it can be calculated in strong form from the gradient of the potential $\tilde{\Phi}$ by $-J_T(\tilde{x})^{-t}\tilde{E}(\tilde{x},t) = -J_T(\tilde{x})^{-t}\tilde{\nabla}\tilde{\Phi}(\tilde{x},t)$. This results in the transformed Poisson equation, see eqn. (B.95).

$$\int_{\tilde{\Omega}} J_T(\tilde{x})^{-t}\tilde{\nabla}\tilde{\Phi}(\tilde{x},t)\cdot J_T(\tilde{x})^{-t}\tilde{\nabla}\tilde{\varphi}(\tilde{x})\ \det(J_T(\tilde{x}))\mathrm{d}\tilde{x} \tag{B.95}$$

$$= \sum_s \frac{q_s}{e}\int_{\mathbb{R}^n}\int_{\tilde{\Omega}} \tilde{f}_s(\tilde{x},\tilde{v},t)\tilde{\varphi}(\tilde{x})\ \det(J_T(\tilde{x}))\mathrm{d}\tilde{x}\mathrm{d}\tilde{v} \quad \forall\tilde{\varphi}\in H^1(\tilde{\Omega}) \tag{B.96}$$

$$\tilde{E}(\tilde{x},t) = -\tilde{\nabla}\tilde{\Phi}(\tilde{x},t) \tag{B.97}$$

Suppose the particles live in the logical coordinates, the transformed Vlasov equation in logical coordinates is needed. For the transformation of the Vlasov eqn. (B.15) we note that the cross product is invariant under basis transform up to the sign of the Jacobi determinant. Since the coordinate transformation is independent of v, the change of variables in v is only a linear one, which yields

$$\nabla_v f_s(x,v,t) = J_T(\tilde{x})\tilde{\nabla}_{\tilde{v}}\tilde{f}_s(\tilde{x},\tilde{v},t). \tag{B.98}$$

Using the covariant and the contravariant representations of the fields result in eqn. (B.99), where some tensors cancel out yielding the final eqn. (B.100).

$$\begin{aligned}\partial_t\tilde{f}_s(\tilde{x},\tilde{v},t) &+ J_T^{-t}(\tilde{x})\tilde{v}\cdot J_T(\tilde{x})^{-t}\tilde{\nabla}_{\tilde{x}}\tilde{f}_s(\tilde{x},\tilde{v},t)\\ &+ \frac{q_s}{e}\frac{m_e}{m_s}\left[J_T(\tilde{x})^{-t}\tilde{E}(\tilde{x},t) + J_T^{-t}(\tilde{x})\tilde{v}\times\frac{J_T(\tilde{x})}{\det(J_T(\tilde{x}))}\hat{B}(\tilde{x},t)\right]\\ &\cdot J_T(\tilde{x})\tilde{f}_s(\tilde{x},\tilde{v},t) = 0\end{aligned} \tag{B.99}$$

$$\begin{aligned}\partial_t\tilde{f}_s(\tilde{x},\tilde{v},t) &+ \frac{\tilde{v}\cdot\tilde{\nabla}_{\tilde{x}}\tilde{f}_s(\tilde{x},\tilde{v},t)}{\det(J_T(\tilde{x}))}\\ &+ \frac{q_s}{e}\frac{m_e}{m_s}\left[\det(J_T(\tilde{x}))J_T(\tilde{x})^{-1}J_T(\tilde{x})^{-t}\tilde{E}(\tilde{x},t) + \tilde{v}\times\hat{B}(\tilde{x},t)\right]\cdot\tilde{\nabla}_{\tilde{v}}\tilde{f}_s(\tilde{x},\tilde{v},t) = 0\end{aligned} \tag{B.100}$$

From eqn. (B.100) the characteristics are extracted into eqn. (B.101).

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{V}(t) &= \frac{q_s}{e}\frac{m_e}{m_s}\left[\tilde{E}(\tilde{X}(t),t) + V(t)\times\hat{B}(\tilde{X}(t),t)\right]\\ \frac{\mathrm{d}}{\mathrm{d}t}\tilde{X}(t) &= \frac{\tilde{V}(t)}{\det(J_T(\tilde{X}(t)))}\end{aligned} \tag{B.101}$$

From eqn. (B.101) we see that the ODE describing the characteristic $X(t)$ is nonlinear due to the Jacobi determinant, even for constant $V(t)$. This complicates the exact integration in

the Hamiltonian splitting, hence we chose the natural basis for the velocity by

$$\dot{\tilde{X}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\underbrace{T^{-1}(X(t))}_{\tilde{X}(t)} = J_T^{-1}(\tilde{X}(t))\underbrace{\dot{X}(t)}_{=V(t)} =: \tilde{V}(t) \tag{B.102}$$

yielding the transform

$$v = J_T(\tilde{x})\tilde{v}, \quad \tilde{v} = J_T(\tilde{x})^{-1}v, \quad \mathrm{d}v = \det(J_T(\tilde{x}))\mathrm{d}\tilde{v}. \tag{B.103}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{V} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(J_T(\tilde{X}(t))^{-1}V(t)\right) = J_T(\tilde{X}(t))^{-1}\dot{V}(t) + \left[\sum_{d=1}^{3}\underbrace{\dot{\tilde{X}}_d(t)}_{=\tilde{V}_d(t)}\partial_{\tilde{x}_d}\left(J_T(\tilde{X}(t))^{-1}\right)\right]V(t)\\ &= J_T(\tilde{X}(t))^{-1}\dot{V}(t) + \left[\sum_{d=1}^{3}\tilde{V}_d(t)\partial_{\tilde{x}_d}\left(J_T(\tilde{X}(t))^{-1}\right)\right]J_T(\tilde{X}(t))\tilde{V}(t)\end{aligned} \tag{B.104}$$

The equations of motions in eqn. (B.105) are then more suitable for exact integration under constant $V(t)$.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{V}(t) &= \frac{q_s}{e}\frac{m_e}{m_s}J_T(\tilde{X}(t))^{-1}J_T(\tilde{X}(t))^{-t}\left[\tilde{E}(\tilde{X}(t),t) + \tilde{V}(t)\times\hat{B}(\tilde{X}(t),t)\right]\\ &\quad + \left[\sum_{d=1}^{3}\tilde{V}_d(t)\partial_{\tilde{x}_d}\left(J_T(\tilde{X}(t))^{-1}\right)\right]J_T(\tilde{X}(t))\tilde{V}(t)\\ \frac{\mathrm{d}}{\mathrm{d}t}\tilde{X}(t) &= \tilde{V}(t)\end{aligned} \tag{B.105}$$

Therefore, the velocity transformation (B.103) is suited for a Hamiltonian splitting along the three spatial dimensions. We observe that the current in the Ampère equation contains a double Jacobi determinant:

$$\begin{aligned}\sum_s \frac{q_s}{e}\int_{\tilde{\Omega}}\left(\int_{\mathbb{R}^n} J_T(\tilde{x})\tilde{v}\tilde{f}_s(\tilde{x},\tilde{v},t)\det(J_T(\tilde{x}))\mathrm{d}\tilde{v}\right)\cdot J_T(\tilde{x})^{-t}\tilde{\varphi}(\tilde{x})\;|\det(J_T(\tilde{x}))|\mathrm{d}\tilde{x} =\\ \sum_s \frac{q_s}{e}\int_{\tilde{\Omega}}\int_{\mathbb{R}^n}\tilde{f}_s(\tilde{x},\tilde{v},t)\tilde{v}\cdot\tilde{\varphi}(\tilde{x})\;|\det(J_T(\tilde{x}))|^2\;\mathrm{d}\tilde{v}\mathrm{d}\tilde{x}.\end{aligned} \tag{B.106}$$

Fortunately, since both characteristics $X = T(\tilde{X})$ and $V = J_T(\tilde{X})\tilde{V}$ have been transformed the particle density[1] reads

$$\tilde{f}_{p,s}(\tilde{x},\tilde{v}) = \frac{1}{N_p}\sum_{n=1}^{N_p}\delta\big(\tilde{x}-\tilde{X}_n\big)\delta\big(\tilde{v}-\tilde{V}_n\big)\frac{W_n}{\det\left(J_T\big(\tilde{X}_n\big)\right)^2}, \tag{B.107}$$

such that the double Jacobi determinant in eqn. (B.106) cancels out upon insertion of $\tilde{f}_{p,s}$. This means that it is in general possible to integrate the current over time using an anti-derivative of the given basis function if both x and v are transformed.

[1] See also B.2.5 and eqn. (B.177).

B.2.2. Back-transform by a Newton method

Suppose $x \in \mathbb{R}^3$ is given and we want to find a $\xi \in [0,1]^3$ such that

$$T(\xi) = x, T^{-1}(x) = \xi. \tag{B.108}$$

If the inverse transform T^{-1} cannot be directly calculated or tabulated by an interpolation (we recommend Chebyshev), an inversion by the Newton method is the most straightforward way to proceed. This only requires the Jacobian J_T or, if available, the inverse J_T^{-1}. The corresponding iteration for $n = 1, 2, \dots$ is given in eqn. (B.109) and should be terminated when the increment $\delta\xi$ drops below a tolerance near machine precision.

$$\xi_{n+1} = \xi_n - \underbrace{J_T(\xi_n)^{-1}\left(T(\xi_n) - x\right)}_{=\delta\xi} \tag{B.109}$$

Such are numerical back-transform should always be implemented in order to test analytical versions and vice versa.

B.2.3. Common coordinate systems

The following section contains an overview of various coordinate transformations that are commonly used. Apart from the standard bodies cylinder, torus and sphere we provide the fusion like geometries such as the D-Shaped Torus representing a Tokamaks core and also a flux surface of a Stellarator. Since it is very important to test the implementation of PDE solvers under this coordinate transformation, analytical expressions of the gradient and the Laplace operator are provided or can be found in [191] such that, e.g. a Poisson solver can be easily tested by the method of manufactured solutions.
Most of these geometries approximate the flux surfaces of a MHD equilibrium in order to obtain a flux surface aligned description. The fusion devices Tokamak and Stellarator have a toroidal shape such that coordinates (r, θ, φ) seem more natural to handle. Here $r \in [r_{\min}, r_{\max}]$ describes the flux surface label, which in the literature is also often denoted by s or ψ. The other two periodic coordinates $\theta \in [0, L_\theta]$ and $\varphi \in [0, L_\varphi]$ parametrize a flux surface. If not denoted otherwise a 2π-periodicity is naturally assumed defining $L_\varphi = L_\theta = 2\pi$. The transform to logical coordinates (ξ_1, ξ_2, ξ_3) is then merely a scalar multiplication. The general form of the coordinate transformation is given in eqn. (B.110) with the corresponding Jacobi matrix in eqn. (B.111).

$$\begin{aligned} T : [r_{\min}, r_{\max}] \times [0, L_\theta] \times [0, L_\varphi] &\to \mathbb{R}^3 \\ (r, \theta, \varphi) &\mapsto \begin{pmatrix} T_x(r,\theta,\varphi) \\ T_y(r,\theta,\varphi) \\ T_z(r,\theta,\varphi) \end{pmatrix} \end{aligned} \tag{B.110}$$

$$\begin{aligned} J_T : [r_{\min}, r_{\max}] \times [0, L_\theta] \times [0, L_\varphi] &\to \mathbb{R}^{3\times 3} \\ (r, \theta, \varphi) \mapsto \begin{pmatrix} \partial_r T_x(r,\theta,\varphi) & \partial_\theta T_x(r,\theta,\varphi) & \partial_\varphi T_x(r,\theta,\varphi) \\ \partial_r T_y(r,\theta,\varphi) & \partial_\theta T_y(r,\theta,\varphi) & \partial_\varphi T_y(r,\theta,\varphi) \\ \partial_r T_z(r,\theta,\varphi) & \partial_\theta T_z(r,\theta,\varphi) & \partial_\varphi T_z(r,\theta,\varphi) \end{pmatrix} \end{aligned} \tag{B.111}$$

Sometimes, contrary to the covariant transform, the normalized Jacobian matrix J_T is used for transformation. For this we define the unit vectors e_r, e_θ, e_φ as the normalized columns of J_T. This also yields the definition of the normalized Jacobi $\bar{J}_T$ matrix in eqn. (B.112).

$$\bar{J}_T \Rightarrow v = \bar{J}_T \tilde{v} \qquad \Rightarrow \tilde{v} = \bar{J}_T^{-1} v \tag{B.112}$$

Any vector field v in coefficient form $\tilde{v} = (v_r, v_\theta, v_\varphi)$ can then be transfered to Cartesian coordinates by

$$v_r(r,\theta,\varphi)e_r(r,\theta,\varphi) + v_\theta(r,\theta,\varphi)e_\theta(r,\theta,\varphi) + v_\varphi(r,\theta,\varphi)e_\varphi(r,\theta,\varphi). \tag{B.113}$$

This approach is mostly needed when magnetic field components are already given as unit vectors in the respective coordinate system, e.g.,

$$B = B_\theta e_\theta + B_\varphi e_\varphi = B_{poloidal}e_\theta + B_{toroidal}e_\varphi. \tag{B.114}$$

For simplified models $B_{poloidal}$ and $B_{toroidal}$ are then only constants. But it is always better to stay in the natural covariant transform, thus, the transformation of scalar fields $\Phi(x,y,z)$ reads

$$\tilde{\Phi}(r,\theta,\varphi) := \Phi(T(r,\theta,\varphi)). \tag{B.115}$$

The same also applies for the gradient of the respective scalar field.

$$\Rightarrow \nabla_{(r,\theta,\varphi)}\tilde{\Phi}(r,\theta,\varphi) = J_T^t(\nabla_{(x,y,z)}\Phi)\,(T(r,\theta,\varphi)) \tag{B.116}$$

$$\Rightarrow \nabla_{(x,y,z)}\Phi = J_T^{-t}\nabla_{(r,\theta,\varphi)}\tilde{\Phi} \tag{B.117}$$

The possible confusion here originates from the fact that the coordinate transformation is always linked to the magnetic equilibrium. In the modeling of an experiment it is then straightforward to use the normalized basis. Since our framework builds upon the covariant and contravariant transform it is strongly advised to translate given values B_θ, B_φ for vector fields B into the contravariant basis.

Polar and cylinder coordinates

When dealing with curvilinear coordinates the first encounter are the two dimensional polar coordinates which become the cylinder coordinates in three dimensions. Here the z axis remains unchanged yielding the substitution $\varphi = z$.

$$T(r,\theta,\varphi) = \begin{pmatrix} r\cos(\theta) \\ r\sin(\theta) \\ \varphi \end{pmatrix},\ J_T(r,\theta,\varphi) = \begin{pmatrix} \cos(\theta) & -r\sin(\theta) & 0 \\ \sin(\theta) & r\cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix},\ \det(J_T(r,\theta,\varphi)) = r \tag{B.118}$$

Recall that $\text{hypot}(x,y) = \sqrt{x^2+y^2}$ and the definition of the atan2 which matches the periodicity of the arctan correctly and defines values for $x = 0$. The atan2 is found in all modern computing language as an intrinsic function and is preferred over the arctan.

$$\text{atan2}(y,x) := \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ \frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \end{cases} \tag{B.119}$$

With these intrinsic functions available, the inverse coordinate transformation can be efficiently computed directly by

$$T^{-1}(x,y,z) = \begin{pmatrix} \text{hypot}(x,y) \\ \text{atan2}(y,x) \\ \varphi \end{pmatrix}. \tag{B.120}$$

For a Particle-In-Fourier method the most expensive operation is the evaluation of the first Fourier mode. When the particles live in physical space, the first Fourier mode of θ is obtained directly by a single division.

$$\mathrm{e}^{\mathrm{i}\theta} = \cos(\theta) + \mathrm{i}\sin(\theta) = \frac{x + \mathrm{i}y}{\sqrt{x^2 + y^2}} = \frac{x + \mathrm{i}y}{r} \tag{B.121}$$

Although this is rather obvious, similar equations can be obtained for other geometries. The combined mapping from the normalized logical coordinates (ξ_1, ξ_2, ξ_3) to the physical space (x_1, x_2, x_3) with the Jacobi matrix is given in eqn. (B.122).

$$T(\xi_1, \xi_2, \xi_3) = \begin{pmatrix} [\xi_1(r_{\max} - r_{\min}) + r_{\min}]\cos(\xi_2 2\pi) \\ [\xi_1(r_{\max} - r_{\min}) + r_{\min}]\sin(\xi_2 2\pi) \\ \xi_3 L_z \end{pmatrix} \tag{B.122}$$

$$T^{-1}(x_1, x_2, x_3) = \begin{pmatrix} (\mathrm{hypot}(x_1, x_2) - r_{\min})\frac{1}{r_{\max} - r_{\min}} \\ \arctan\left(\frac{x_2}{x_1}\right)\frac{1}{2\pi} \\ \frac{x_3}{L_z} \end{pmatrix} \tag{B.123}$$

$$J_T(\xi) = \begin{pmatrix} \cos(2\pi\xi_2)(r_{\max} - r_{\min}) & -2\pi\sin(2\pi\xi_2)(r_{\min} + \xi_1(r_{\max} - r_{\min})) & 0 \\ \sin(2\pi\xi_2)(r_{\max} - r_{\min}) & 2\pi\cos(2\pi\xi_2)(r_{\min} + \xi_1(r_{\max} - r_{\min})) & 0 \\ 0 & 0 & L_z \end{pmatrix} \tag{B.124}$$

A variant of describing a toroidal device are also cylindrical coordinates, mostly referred to as (R, Z, φ) coordinates.

$$T(R, Z, \varphi) = \begin{pmatrix} R\cos(\varphi) \\ R\sin(\varphi) \\ Z \end{pmatrix}, \; J_T(R, Z, \varphi) = \begin{pmatrix} \cos(\varphi) & -R\sin(\varphi) & 0 \\ \sin(\varphi) & -R\sin(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; \det(J_T(R, Z, \varphi)) = R \tag{B.125}$$

$$J_T^{-1} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\frac{\sin\theta}{R} & \frac{\cos\theta}{R} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; J_T^{-1}J_T^{-t} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{R^2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; J_T^t J_T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & R^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.126}$$

Using eqn. (B.85), the curl for a vector $\hat{F}(R, Z, \varphi)$ in the contravariant basis reads

$$\begin{aligned} \nabla \times F &= \frac{1}{R}\begin{pmatrix} \cos(\varphi) & -R\sin(\varphi) & 0 \\ \sin(\varphi) & -R\sin(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \partial_R \\ \partial_\varphi \\ \partial_Z \end{pmatrix} \times \left[\frac{1}{R}\begin{pmatrix} 1 & 0 & 0 \\ 0 & R^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \hat{F}_R \\ \hat{F}_\varphi \\ \hat{F}_Z \end{pmatrix}\right] \\ &= \frac{1}{R}\begin{pmatrix} \cos(\varphi) & -R\sin(\varphi) & 0 \\ \sin(\varphi) & -R\sin(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \frac{\partial_\varphi \hat{F}_Z}{R} - R\,\partial_Z \hat{F}_\varphi \\ \frac{\partial_Z \hat{F}_R}{R} - \frac{1}{R}\partial_R \hat{F}_Z + \frac{\hat{F}_Z}{R^2} \\ R\partial_R \hat{F}_\varphi + \hat{F}_\varphi - \frac{\partial_\varphi \hat{F}_R}{R} \end{pmatrix} \end{aligned} \tag{B.127}$$

The Laplace operator is given as

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial R^2} + \frac{1}{R}\frac{\partial\Phi}{\partial R} + \frac{1}{R^2}\frac{\partial\Phi}{\partial\varphi^2} + \frac{\partial^2\Phi}{\partial Z^2}. \tag{B.128}$$

As we can see from eqn. (B.128) the Laplace operator differs from the Cartesian ones as it includes $\frac{1}{R}$ terms. A more natural transformation is achieved in the log-polar coordinates, where the radial component is chosen as $\rho = \log(r)$.

$$T(\rho, \theta, \varphi) = \begin{pmatrix} \mathrm{e}^\rho\cos(\theta) \\ \mathrm{e}^\rho\sin(\theta) \\ \varphi \end{pmatrix}, \; J_T(\rho, \theta, \varphi) = \begin{pmatrix} \mathrm{e}^\rho\cos(\theta) & -\mathrm{e}^\rho\sin(\theta) & 0 \\ \mathrm{e}^\rho\sin(\theta) & \mathrm{e}^\rho\cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; \det(J_T(\rho, \theta, \varphi)) = \mathrm{e}^\rho \tag{B.129}$$

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial\rho} + \frac{\partial^2\Phi}{\partial\theta} + \frac{\partial^2\Phi}{\partial\varphi} \tag{B.130}$$

Toroidal coordinate system (r, φ, θ) (toroidal cylinder)

Let R_0 be the major radius of the torus. Define the toroidal angle $\varphi \in (0, 2\pi)$ and the poloidal angle $\theta \in (0, 2\pi)$. The radius in the poloidal plane is defined as $r \geq 0$ and due to the torus geometry limited by the major radius $r \leq r_{\max} < R_0$ thus $r \in (0, r_{\max})$, where $r_{\max}$ is also called the minor radius.
Transformation to Cartesian coordinates $T : (r, \varphi, \theta) \mapsto (x, y, z)$ is given by

$$T(r, \theta, \varphi) = \begin{pmatrix} (R_0 + r\cos(\theta))\cos(\varphi) \\ (R_0 + r\cos(\theta))\sin(\varphi) \\ r\sin(\theta) \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{B.131}$$

along with the inverse transform

$$T^{-1}(x, y, z) = \begin{pmatrix} \sqrt{\left(\sqrt{x^2+y^2} - R_0\right)^2 + z^2} \\ \operatorname{atan}\left(\frac{z}{\sqrt{x^2+y^2}-R_0}\right) \\ \operatorname{atan}\left(\frac{y}{x}\right) \end{pmatrix} = \begin{pmatrix} r \\ \theta \\ \varphi \end{pmatrix} \tag{B.132}$$

and the Jacobi matrix

$$J_{T(r,\varphi,\theta)} = \nabla T(r, \varphi, \theta)^t = \begin{pmatrix} \cos\theta\cos\varphi & -(R_0 + r\cos\theta)\sin\varphi & -r\sin\theta\cos\varphi \\ \cos\theta\sin\varphi & (R_0 + r\cos\theta)\cos\varphi & -r\sin\theta\sin\varphi \\ \sin\theta & 0 & r\cos\theta \end{pmatrix}, \tag{B.133}$$

and Jacobi determinant

$$\det(J_{T(r,\varphi,\theta)}) = r\left(R_0 + r\cos(\theta)\right) = rR_0 + r^2\cos(\theta). \tag{B.134}$$

Since we assumed $R_0 > r > 0$ we can drop the absolute value in the Jacobi determinant by

$$|\det(J_{T(r,\varphi,\theta)})|\left|rR_0 + r^2\cos(\theta)\right| = r^2\left|\underbrace{\frac{R_0}{r}}_{<1} + \cos(\theta)\right| = rR_0 + r^2\cos(\theta). \tag{B.135}$$

The Laplace operator for $\Phi(r, \theta, \varphi)$ is

$$\begin{aligned}\Delta\Phi = \frac{1}{r^2}\partial_{\theta\theta}\Phi + \partial_{rr}\Phi + \frac{1}{(R_0 + r\cos\theta)^2}\partial_{\varphi\varphi}\Phi \\ + \left(\frac{1}{r} + \frac{\cos\theta}{R_0 + r\cos\theta}\right)\partial_r\Phi - \frac{\sin\theta}{r(R_0 + r\cos\theta)}\partial_\theta\Phi.\end{aligned} \tag{B.136}$$

With the inverse Jacobi matrix

$$J_T(r, \varphi, \theta)^{-1} = \begin{pmatrix} \cos\varphi\cos\theta & \cos\theta\sin\phi & \sin\theta \\ -\frac{\sin\varphi}{R_0+r\cos\theta} & \frac{\cos\varphi}{R_0+r\cos\theta} & 0 \\ -\frac{\cos\varphi\sin\theta}{r} & -\frac{\sin\varphi\sin\theta}{r} & \frac{\cos\theta}{r} \end{pmatrix}, \tag{B.137}$$

useful expressions concerning the Poisson equation the are

$$J_T^{-1}J_T^{-t} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{(R_0+r\cos\theta)^2} & 0 \\ 0 & 0 & \frac{1}{r^2} \end{pmatrix} \tag{B.138}$$

and

$$J_T^{-1} J_T^{-t} \det(J_T) = \begin{pmatrix} \cos(\theta) r^2 + R_0 r & 0 & 0 \\ 0 & \frac{r}{R_0 + r\cos\theta} & 0 \\ 0 & 0 & \frac{R_0 + r\cos\theta}{r} \end{pmatrix} \tag{B.139}$$

Torus (toroidal harmonics)

The natural coordinates for a torus do not describe the poloidal plane starting from cylindrical coordinates and bending them around in order to obtain a torus. Yet they have a less intuitive, rather complicated transformation for a major radius $R_0 > 0$ and $\tau \geq 0,\ \theta, \phi \in [0, 2\pi]$.

$$T(\tau, \theta, \varphi) = \begin{pmatrix} R_0 \frac{\sinh(\tau)}{\cosh(\tau)+\cos(\theta)} \cos(\varphi) \\ R_0 \frac{\sinh(\tau)}{\cosh(\tau)+\cos(\theta)} \sin(\varphi) \\ -R_0 \frac{\sin(\theta)}{\cosh(\tau)+\cos(\theta)} \end{pmatrix} \tag{B.140}$$

The advantage of these coordinates is that the Laplace operator can be separated in Fourier modes and associated Legendre functions of first and second kind [237]. Thus, they are more suitable for highly accurate spectral methods. Although they are not fairly widespread, it can be shown that they have superior qualities in expanding the vacuum magnetic field in a Tokamak [238, 239] even up to the X-point [240]. Also Shafranovs original work on the Shafranov shift uses these coordinates [241]. Since they are not at all field aligned, it is not feasible to use them in any form of advection related to a magnetic field in a semi-Lagrangian or Eulerian method. Because our Lagrangian particles live anyhow in physical space the coordinates are only needed by the fields for which they seem to be an unintuitive yet beautiful candidate. We recall some definitions in eqn. (B.141).

$$\cosh(x) := \frac{e^x + e^{-x}}{2},\ \sinh(x) := \frac{e^x - e^{-x}}{2},$$
$$\sinh^{-1}(z) = \log(z + \sqrt{1+z^2}),\ \cosh\left(\sinh^{-1}(z)\right) = \sqrt{z^2+1} \tag{B.141}$$

For constant τ, the resulting surfaces form tori of radius r, defining a transformation between r and τ in eqn. (B.142).

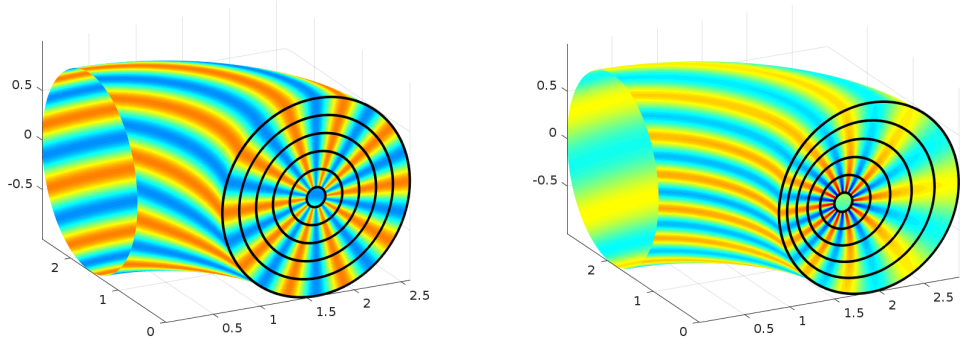
$$r = \frac{R_0}{\sinh(\tau)} \ \Leftrightarrow\ \tau = \sinh^{-1}\left(\frac{R_0}{r}\right) = \log\left(\frac{R_0}{r} + \sqrt{1 + \frac{R_0^2}{r^2}}\right) \tag{B.142}$$

$$\frac{\mathrm{d}r}{\mathrm{d}\tau} = -\frac{R_0 \cosh(\tau)}{\sinh(\tau)^2} = \frac{R_0 \cosh(\tau)}{1 - \cosh(\tau)^2}, \quad \frac{\mathrm{d}\tau}{\mathrm{d}r} = -\frac{R_0}{r\sqrt{R_0^2 + r^2}} \tag{B.143}$$

Note also that $\cosh(\tau) = \sqrt{\frac{R_0^2}{r^2} + 1}$. The major radius R of the toroidal surface with radius r or label τ can be found by $R(\tau) = R_0 \coth(\tau)$ or $R(r) = \sqrt{R_0^2 + r^2}$. Inserting eqn. (B.142) back into the original transform yields a transform where r can be seen again as a flux surface label describing tori of radius r.

$$T(r, \theta, \varphi) = \begin{pmatrix} \frac{R_0^2}{\sqrt{r^2+R_0^2}+r\cos(\theta)} \cos(\varphi) \\ \frac{R_0^2}{\sqrt{r^2+R_0^2}+r\cos(\theta)} \sin(\varphi) \\ r\frac{-R_0 \sin(\theta)}{\sqrt{r^2+R_0^2}+r\cos(\theta)} \end{pmatrix} \tag{B.144}$$

Obviously the transformation given in eqn. (B.144) does not have a singularity at $r = 0$ and is, therefore, a diffeomorphism. The inverse transforms for (B.140) and (B.144) are given in



(a) toroidal-cylinder (b) toroidal

Figure B.1.: Segment of a Torus in the toroidal and toroidal-cylinder coordinates $R_0 = 1.6, r_{\max} = 1$. The $\exp(\mathrm{i}(10\theta + 3\varphi))$ mode weighted by $\sqrt{\cosh(\tau) + \cos(\theta)}$ is depicted on every surface of constant τ. Note the compression of the nested toroidal surfaces and the Fourier modes on the inner side of the torus.

eqn. (B.145).

$$
\begin{aligned}
\tau &= \frac{1}{2} \ln \left(\frac{\left(\sqrt{x^2+y^2}+R_0\right)^2 + z^2}{\left(\sqrt{x^2+y^2}-R_0\right)^2 + z^2} \right) \\
r &= 2 \left[\frac{\left(\sqrt{x^2+y^2}+R_0\right)^2 + z^2}{\left(\sqrt{x^2+y^2}-R_0\right)^2 + z^2} \right]^{\frac{1}{2}} \left[\frac{\left(\sqrt{x^2+y^2}+R_0\right)^2 + z^2}{\left(\sqrt{x^2+y^2}-R_0\right)^2 + z^2} - 1 \right]^{-1} \\
\theta &= \cos^{-1} \left[\frac{\left(R_0^2 - (x^2+y^2+z^2)\right)}{\sqrt{\left(\sqrt{x^2+y^2}+R_0\right)^2 + z^2} \sqrt{\left(\sqrt{x^2+y^2}-R_0\right)^2 + z^2}} \right] \\
\varphi &= \operatorname{atan}\left(\frac{y}{x}\right)
\end{aligned}
\tag{B.145}
$$

For testing the expressions for many differential operators including the Laplace operator can be found in [191][pp.112-114]. In the toroidal coordinates, a solution Φ to the Laplace equation $\Delta\Phi = 0$ decomposes into the product of four orthogonal functions [237, 191].

$$
\Phi(\tau, \theta, \varphi) = \sqrt{\cosh(\tau) + \cos(\theta)} \cdot \begin{cases} P^n_{m-\frac{1}{2}}(\cosh(\tau)) \cdot \mathrm{e}^{\mathrm{i}m\theta} \cdot \mathrm{e}^{\mathrm{i}n\varphi} \\ Q^n_{m-\frac{1}{2}}(\cosh(\tau)) \cdot \mathrm{e}^{\mathrm{i}m\theta} \cdot \mathrm{e}^{\mathrm{i}n\varphi} \end{cases} \tag{B.146}
$$

The associated Legendre functions P and Q of first and second kind can be evaluated efficiently by using the "DTHOR3 2.0" algorithm, see [242, 243, 244]. With this separation the Poisson equation in toroidal coordinates can again be solved very efficiently in spectral space [245, 246]. Thus, a Particle in Toroidal Harmonics (PITH) algorithm is possible and would provide extreme scalability for the Poisson solver. But for a Vlasov simulation the corresponding MHD equilibrium has to be solved using the same spectral method, which has not yet been done. Note that the difference to the cylindrical harmonics is the additional weighting factor $\sqrt{\cosh(\tau) + \cos(\theta)}$, which compresses the modes at the inner side of the torus, see fig.B.1. The Laplace operator, among a full description of the coordinate system

can be found in [191][pp.112-115].

$$\begin{aligned}\Delta\Phi(\tau,\theta,\varphi) = &\left[\frac{\partial}{\partial\tau}\left(\frac{\sinh(\tau)}{\cosh(\tau)+\cos(\theta)}\right)\frac{\partial\Phi}{\partial\tau} + \sinh(\tau)\frac{\partial}{\partial\theta}\left(\frac{1}{\cosh(\tau)+\cos(\theta)}\frac{\partial\Phi}{\partial\theta}\right)\right] \\ &\cdot\left(\frac{(\cosh(\tau)+\cos(\theta))^2}{R_0^2\sinh(\tau)}\right) + \frac{(\cosh(\tau)+\cos(\theta))^2}{R_0^2\sinh(\tau)^2}\frac{\partial^2\Phi}{\partial\varphi^2}\end{aligned} \tag{B.147}$$

We learned that the cylindrical toroidal coordinates are not the *true* harmonics in the toroidal geometry but they are good to to use, since their coordinate transformation itself is not singular for $r = 0$.

Parametrization of MHD equilibria

Equilibrium configurations for Tokamaks can be simply described by incorporating an ellipticity $\kappa(r)$ and triangularity $\delta(r)$ into the pseudo-toroidal coordinates, see eqn. (B.148). This is known as Soloviev equilibrium [247, 248], but actually better descriptions including divergence free magnetic fields are available but more involved [249].

$$T(r,\theta,\varphi) = \begin{pmatrix}\left[R_0(r) + r\cos\left(\theta + \sin^{-1}(\delta)\sin(\theta)\right)\right]\cos(\varphi) \\ \left[R_0(r) + r\cos\left(\theta + \sin^{-1}(\delta)\sin(\theta)\right)\right]\sin(\varphi) \\ \kappa(r) r\sin(\theta)\end{pmatrix} \tag{B.148}$$

The r dependent profiles $R_0(r), \kappa(r)$ and $\delta(r)$ are obtained by solving ODEs for the given initial values at the outermost flux surface, involving additional parameters see [247]. For the cases used in this work we assume constant profiles with $\kappa = 1.44$, $\delta = 0.416$ and the major radius is set to $R_0 = \frac{L_\varphi}{2\pi}$.

Helical Coordinates

Stellarator equilibria are found by minimize the total MHD plasma energy consisting of magnetic and thermal contribution using some additional regularization [250]. Such an equilibrium can be described by a Fourier series over poloidal and toroidal modes [251], according to eqn. (B.149).

$$T(r,\theta,\varphi) = \begin{pmatrix}\left[\sum_{m=0}^{m_{\text{pol}}}\sum_{n=-n_{\text{tor}}}^{n_{\text{tor}}} R_{m,n}(r)\cos(m\theta - n\varphi)\right]\cos(\varphi) \\ \left[\sum_{m=0}^{m_{\text{pol}}}\sum_{n=-n_{\text{tor}}}^{n_{\text{tor}}} R_{m,n}(r)\cos(m\theta - n\varphi)\right]\sin(\varphi) \\ \sum_{m=0}^{m_{\text{pol}}}\sum_{n=-n_{\text{tor}}}^{n_{\text{tor}}} Z_{m,n}(r)\cos(m\theta - n\varphi)\end{pmatrix} \tag{B.149}$$

The Fourier coefficients $R_{m,n}$ and $Z_{m,n}$ with $R_{0,n} = Z_{0,n} = 0$ for $n < 0$, depend on the flux surface label r, but are also subject to the famous Stellarator symmetry

$$R_{-m,-n}(r) = R_{m,n}(r) \text{ and } Z_{-m,-n}(r) = -Z_{m,n}(r), \tag{B.150}$$

which is already incorporated into the representation in eqn. (B.149). Such Fourier geometries with many coefficients are in general hard to invert and require high resolution but yield astonishing geometries, see fig. B.2. If one does not have access to a code providing the coefficients, one is restricted to use published information for single flux surfaces, see e.g. [252]. Thus we either solve the entire problem directly or construct a simplified toy model that incorporates the basic numerical difficulty. For this we rotate an ellipse in the poloidal plane s times around the magnetic axis yielding a helical shape for eqn. (B.151).

$$T(r,\theta,\varphi) = \begin{pmatrix}(R_0 + ar\cos(\varphi s)\cos(\theta) - br\sin(\varphi s)\sin(\theta))\cos(\varphi) \\ (R_0 + ar\cos(\varphi s)\cos(\theta) - br\sin(\varphi s)\sin(\theta))\sin(\varphi) \\ ar\sin(\varphi s)\cos(\theta) + br\cos(\varphi s)\sin(\theta)\end{pmatrix} \tag{B.151}$$

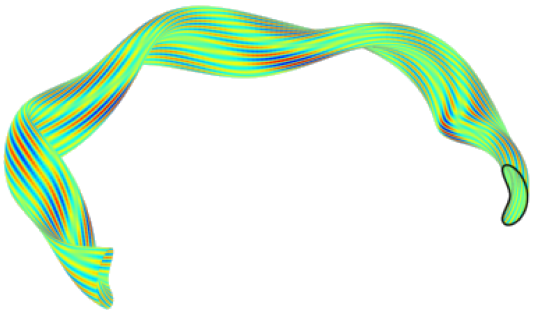


Figure B.2.: Quasi-Helical flux surface with parameters from [252] filled by a randomly synthesized structure.

$$\det\left(J_T(r,\theta,\varphi)\right) = rab\left[R_0 + ar\cos(\varphi s)\cos(\theta) - br\sin(\varphi s)\sin(\theta)\right] \tag{B.152}$$

With given ellipticity κ of an equilibrium we set $a = \kappa, b = \frac{1}{\kappa}$.

B.2.4. Guiding center (2d) and drift kinetic (3d1v)

We begin with the coordinate transformation of the guiding center and drift kinetic model and finish with examples for polar and cylinder coordinates. As already mentioned before, for the polar and cylinder coordinates it is customary to not use the covariant transform but the normalized covariant transform using the normalized Jacobi matrix $\bar{J}$. After the coordinate transformation $\tilde{f}(r,\theta,\varphi,v_\parallel,t) = f(T(r,\theta,\varphi),v_\parallel,t)$ the drift kinetic eqn. (B.61) reads

$$\partial_t \tilde{f} + \left(v_\parallel \cdot \bar{J}_T\tilde{b} + \bar{J}_T\tilde{b} \times J_T^{-t}\tilde{\nabla}\tilde{\Phi} \times \bar{J}_T\tilde{b}\right) \cdot J_T^{-t}\tilde{\nabla}_{\tilde{x}}\tilde{f} + \left(\bar{J}_T\tilde{b} \cdot J_T^{-t}\tilde{\nabla}\tilde{\Phi}\right) \cdot \nabla_{v_\parallel}\tilde{f} = 0, \quad \text{(B.153)}$$

and the corresponding characteristics are

$$\begin{aligned} \frac{d}{dt}\tilde{X}(t) &= \left[J_T^{-1}\left(J_T^{-t}\nabla_{(r,\theta,\varphi)}\tilde{\Phi}(\tilde{X}(t),t) - (\nabla_{(r,\theta,\varphi)}\tilde{\Phi}^t(\tilde{X}(t),t)J_T^{-1}\bar{J}_T\tilde{b} - v_\parallel)\cdot\bar{J}_T\tilde{b}\right)\right]^t, \\ \frac{d}{dt}V_\parallel(t) &= \nabla_{(r,\theta,\varphi)}\tilde{\Phi}^t(\tilde{X}(t),t)J_T^{-1}\bar{J}_T\tilde{b}. \end{aligned} \quad \text{(B.154)}$$

The parallel velocity $v_\parallel \| \vec{B}$ depends on the magnetic field. The transformation of the quasi-neutrality equation is discussed separately with the introduction of a general coordinate elliptic solver. The guiding center density evolution along with the characteristic $X(t)$ for a spatial density $f(x)$ reads

$$\begin{aligned} &\partial_t \tilde{f} + \bar{J}_T\tilde{b} \times J_T^{-t}\tilde{\nabla}\tilde{\Phi} \times \bar{J}_T\tilde{b} \cdot J_T^{-t}\tilde{\nabla}_{\tilde{x}}\tilde{f} = 0, \\ &\frac{d}{dt}\tilde{X}(t) = \left[J_T^{-1}\left(J_T^{-t}\nabla_{(r,\theta,\varphi)}\tilde{\Phi}(\tilde{X}(t),t) - (\nabla_{(r,\theta,\varphi)}\tilde{\Phi}^t(\tilde{X}(t),t)J_T^{-1}\bar{J}_T\tilde{b} - v_\parallel)\cdot\bar{J}_T\tilde{b}\right)\right]^t. \end{aligned} \quad \text{(B.155)}$$

For $B = (0,0,1)^t$ the vorticity equation is obtained in Cartesian coordinates. The drift kinetic model in polar coordinates $f(r,\theta,\varphi,t)$ is given in eqns. (B.156) and (B.157).

$$\partial_t f - \frac{\partial_r\Phi}{r}\partial_r f + \frac{\partial_r\Phi}{r}\partial_\theta f + v\partial_\varphi f - \partial_\varphi\Phi\partial_v f = 0, \ t \in [0,T] \quad \text{(B.156)}$$

$$-\left[\partial_r\Phi + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r\Phi + \frac{1}{r^2}\partial_\theta\Phi\right] + \frac{1}{T_e(r)}(\Phi - \bar{\Phi}) = \frac{1}{n_0(r)}\int_{\mathbb{R}} f \ dv - 1 \quad \text{(B.157)}$$

Here the toroidally averaged $\bar{\Phi}$ is defined as

$$\bar{\Phi}(r,\theta) := \frac{1}{L_\varphi}\int_0^{L_\varphi}\Phi(r,\theta,\varphi)\mathrm{d}\varphi. \quad \text{(B.158)}$$

Integrating eqn. (B.157) over φ yields a separate elliptic PDE (B.159), which has to be solved before eqn. (B.157).

$$\begin{aligned} &-\left[L_\varphi\partial_r\bar{\Phi} + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)L_\varphi\partial_r\bar{\Phi} + L_\varphi\frac{1}{r^2}\partial_\theta\bar{\Phi}\right] \\ &\qquad + \frac{1}{T_e(r)}(L_\varphi\bar{\Phi} - L_\varphi\bar{\Phi}) = \frac{L_\varphi}{n_0(r)}\int_0^{L_\varphi}\int_{\mathbb{R}} f \ \ \mathrm{d}v\mathrm{d}\varphi - L_\varphi \\ &\Leftrightarrow -\left[\partial_r\bar{\Phi} + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)}\right)\partial_r\bar{\Phi} + \frac{1}{r^2}\partial_\theta\bar{\Phi}\right] = \frac{1}{n_0(r)}\int_0^{L_\varphi}\int_{\mathbb{R}} f \ \mathrm{d}v\mathrm{d}\varphi - 1 \end{aligned} \quad \text{(B.159)}$$

A guiding center type equation on the polar plane [253] for a density $f(r,\theta,t)$ with $r \in [r_{\min}, r_{\max}]$, $\theta \in [0,2\pi]$ reads

$$\partial_t f - \frac{\partial_\theta\Phi}{r}\partial_r f + \frac{\partial_r\Phi}{r}\partial_\theta f = 0, \quad t \in [0,T] \quad \text{(B.160)}$$

where the potential $\Phi(r,\theta,t)$ is given by the Poisson equation in polar coordinates

$$\begin{aligned} -\partial_r^2\Phi - \frac{1}{r}\partial_r\Phi - \frac{1}{r^2}\Phi &= f, \\ \hat{\Phi}_m(r=0,t) &= 0 \text{ for } m \neq 0, \\ \partial_r\hat{\Phi}_m(r=0,t) &= 0 \text{ for } m = 0, \\ \Phi(r_{\max},t) &= 0. \end{aligned} \tag{B.161}$$

In order to treat the singularity at the origin a boundary conditions for the Fourier transform in θ, $\hat{\Phi}_m(r,t) := \frac{1}{2\pi}\int_0^{2\pi} \mathrm{e}^{-im\theta}\Phi(r,\theta,t)\mathrm{d}\theta$ is required. We define the electric field as usual as

$$E(r,\theta,t) = -\nabla_{(r,\theta)}\Phi(r,\theta,t) = -\left(\partial_r\Phi(t,r,\theta)\ \vec{e}_r\ +\ \frac{1}{r}\partial_\theta\Phi(t,r,\theta)\ \vec{e}_\theta\right),$$
$$E_r = -\partial_r\Phi, \quad E_\theta = -\frac{1}{r}\partial_\theta\Phi. \tag{B.162}$$

Here we chose homogeneous Neumann boundary condition at $r_{\min}$ for the first Fourier mode $\hat{\Phi}_0$ at $r_{\min}$ and homogeneous Dirichlet boundary conditions for the non zero modes. For more details on boundary conditions we refer to [253]. The characteristics of eqn. (B.160) read

$$\begin{aligned} \frac{d}{dt}r(t) &= -\frac{\partial_\theta\Phi(t,r(t),\theta(t))}{r(t)} = E_\theta(t,r,\theta), \\ \frac{d}{dt}\theta(t) &= \frac{\partial_r\Phi(t,r(t),\theta(t))}{r(t)} = -\frac{E_r(t,r(t),\theta(t))}{r(t)}. \end{aligned} \tag{B.163}$$

B.2.5. Coordinate transformations for Monte Carlo characteristics

When implementing a particle method in curvilinear coordinates - something involving a coordinate transformation, like a cylinder - one often hears: "You should put the Jacobian into the weights, but if you really do not want to, you can just evaluate it for every particle." The problem here is the term "weight", because in this framework there is no weight per se, but only a ratio of two likelihoods $w_k = \frac{f_k}{g_k}$. Here we want to sort this out by explaining where the weights come from, where exactly the Jacobian enters and what choices are available. Given a mapping by a $\mathcal{C}^1$ diffeomorphism $T : [0,1]^d \to \Omega \subset \mathbb{R}^d$, $\xi \mapsto T(\xi) = x$ from logical to physical coordinates with $T([0,1]^d) = \Omega$. Let $J_T(\xi) = |\det(\nabla T(\xi))|$ denote the Jacobi determinant of T. Without loss of generality we regard only the spatial coordinate, because this is the typical situation. Let $g(x,v)$ describe a probability density, (X,V) a corresponding random deviate and $f(x,v)$ the Vlasov density. For some function ψ we can calculate a moment in physical and logical coordinates using the coordinate transformation and the Jacobi determinant. In the typical situation a test-function ψ defined in logical coordinates $\xi \mapsto \psi(\xi)$ is given. With a change of coordinates we then obtain the following identity.

$$\int_{\mathbb{R}}\int_{\Omega} f(x,v)\psi(T^{-1}(x))\ \mathrm{d}x\mathrm{d}v = \int_{\mathbb{R}}\int_{[0,1]^d} f(T(\xi),v)\psi(\xi)J_T(\xi)\ \mathrm{d}\xi\mathrm{d}v \tag{B.164}$$

We define the random deviate of the transformed spatial coordinate $\Xi = T^{-1}(X)$ with $T(\Xi) = X$, and obtain the tuple (Ξ, V). Then the Monte Carlo integral, using the sampling density g of x, v yields

$$\mathbb{E}\left[\frac{f(X,V)}{g(X,V)}\psi(T^{-1}(X))\right] = \int_{\mathbb{R}}\int_{\Omega}\frac{f(x,v)}{g(x,v)}\psi(T^{-1}(x))\ g(x,v)\ \mathrm{d}x\mathrm{d}v =$$
$$\int_{\mathbb{R}}\int_{[0,1]^d}\frac{f(T(\xi),v)}{g(T(\xi),v)}\psi(\xi)\ g(T(\xi),v)J_T(\xi)\mathrm{d}\xi\mathrm{d}v = \mathbb{E}\left[\frac{f(T(\Xi),V)}{g(T(\Xi),V)}\psi(\Xi)\right]. \tag{B.165}$$

Here one might wonder why the Jacobian J_T appears in the second integral but not in the second expectation in eqn. (B.165). We know that g is a probability density describing the distribution of (X, V). We seek now the probability distribution $\tilde{g}$ of the transformed random deviate $(\Xi, V) = (T^{-1}(X), V)$. By the change of coordinates in the integral

$$\iint g(x,v)\,\mathrm{d}x\mathrm{d}v = \iint \underbrace{g(T(\xi),v)J_T(\xi)}_{:=\tilde{g}(\xi,v)}\,\mathrm{d}\xi\mathrm{d}v, \tag{B.166}$$

the transformed probability density is obtained as

$$\tilde{g}(\xi,v) = g(T(\xi),v)J_T(\xi). \tag{B.167}$$

In eqn. (B.165) the change of coordinates in the expectation is trivial, we just transform the samples as we need to and the Jacobian is already included. In eqn. (B.168) we explicitly point out the corresponding probability density that is used in the integral form of the expectations.

$$\mathbb{E}_g\left[\frac{f(X,V)}{g(X,V)}\psi(T^{-1}(X))\right] = \mathbb{E}_{\tilde{g}}\left[\frac{f(T(\Xi),V)}{g(T(\Xi),V)}\psi(\Xi)\right] \tag{B.168}$$

Therefore, we do not need the Jacobian J_T when the markers are available in physical space, since the expectation takes care of it. But where does it enter?
We change viewpoint to the time dependent problem solved by the method of characteristics. There a marker $(X_n(t), V_n(t))$, following a characteristic transports the Vlasov density $f_n = f(X_n(t), V(t)_n, t)$ and the sampling density $g_n = f(X_n(t), V_n(t), t)$, which are both constants in time. In different coordinate systems the marker $(X_n(t), V_n(t))$ has different coordinates e.g. (Ξ_n, V_n) but the value it transports is exactly the same. Often the mapping T allows are more elegant description of a density, and therefore, we define the transformed densities $\hat{f}$ and $\hat{g}$ as

$$\hat{f}(\xi,v,t) := f(T(\xi),v,t) \text{ and } \hat{g}(\xi,v,t) := g(T(\xi),v,t). \tag{B.169}$$

Sometimes, for the sake of a simplified notation, the density $f(\xi, v, t) = f(x = T(\xi), v, t)$ is implicitly defined by the change in argument from x to ξ, which can be helpful or confusing. In eqn. (B.170) we stay explicit and again point out the constant values transported by a marker.

$$\begin{aligned} f_n = f(X_n(t),V_n(t),t) = f\left(T(\Xi_n(t)),V_n(t),t\right) = \hat{f}\left(\Xi_n(t),V_n(t),t\right) \\ \text{and } g_n = g(X_n(t),V_n(t),t) = g\left(T(\Xi_n(t)),V_n(t),t\right) = \hat{g}\left(\Xi_n(t),V_n(t),t\right) \end{aligned} \tag{B.170}$$

Losing the time dependence for the sake of notation, the mapped density $\hat{g}(\xi, v, t)$ is not a probability distribution like $\tilde{g}(x, v)$ which with eqn. (B.168) and eqn. (B.169) yields a transform relation, see eqn. (B.171).

$$g(x,v) = \hat{g}(\xi,v) = \frac{\tilde{g}(\xi,v)}{J_T(\xi)}, \quad x = T(\xi) \tag{B.171}$$

Equation (B.171), also valid for $f, \hat{f}$ and $\tilde{f}$, is the most important one for us, because it describes exactly where and when to put the Jacobian. Typically we have a domain $\Omega = T([0,1]^d)$ that is parameterized by T, therefore it is much more comfortable to work in the logical coordinates. Instead of describing the initial condition $f(x, v)$ in physical coordinates we chose the more convenient $\hat{f}(\xi, v)$. We now, at $t = 0$, need to draw a marker (X_n, V_n), but we want to do this in logical coordinates (Ξ_n, V_n) according to a probability density

$\tilde{g}(\xi, v)$. Mapping the marker into physical coordinates $(X_n, V_n) = (T(\Xi_n), V_n)$ yields our desired sample with distribution g given by

$$g(x, v) = \frac{\tilde{g}(T^{-1}(x), v)}{J_T(T^{-1}(x))} = \tilde{g}(T^{-1}(x), v) J_{T^{-1}}(x). \tag{B.172}$$

The likelihoods f_n, g_n are then the same as in eqn. (B.170), where we stay in the comfortable logical space and obtain for the marker (Ξ_n, V_n):

$$f_n = \hat{f}\,(\Xi_n, V_n) \text{ and } g_n = \frac{\tilde{g}(\Xi_n, v)}{J_T(\Xi_n)}. \tag{B.173}$$

From eqn. (B.173) we conclude that only the sampling in logical coordinates makes it necessary for us to consider the Jacobian J_T. For variance reduction we would like the sampling distribution g to be close to f, such that different choices are available: The first one is to chose the sampling distribution $\tilde{g}(\xi, v)$ as a normalized version of $\hat{f}(\xi, v)$ resulting in $g(x, v)$ not being close to $f(x, v)$ because of the division by the Jacobian eqn. (B.173). This is mostly referred to as "putting the Jacobian in the weights". The other option is to "put the Jacobian in the particles". In order to be close to f we define $\hat{g}(\xi, v)$ as a normalized version of $\hat{f}(\xi, v)$ and chose the sampling distribution as

$$\tilde{g}(\xi, v) = \hat{g}(\xi, v) J_T(\xi). \tag{B.174}$$

This results in $g(x, v)$ being a normalized version of $f(x, v)$ yielding a variance reduction. The uniform sampling, choosing $\hat{g}(\xi, v) = 1\hat{g}(v)$ is then "sampling the Jacobian", because the spatial part of the probability density is then only the Jacobian $\tilde{g}(\xi, v) = \hat{g}(v) J_T(\xi)$. We will later give an example for polar coordinates.
Likelihoods f_n, g_n should always be defined in the physical space (Lebesgue measure), as we do here. It remains to note that control variates in physical $h(x, v)$ or logical coordinates $\tilde{h}(\xi, v)$ are used as usual, e.g., the δ-weight is defined as

$$\delta w_n = \frac{f_n - h(\Xi_n, V_n)}{g_n}. \tag{B.175}$$

We have treated the stochastic aspect and proceed with the deterministic Klimontovich density similar to [48]. Hence, the Klimontovich density f_p as a sum of Dirac-δ functions shall replace f in the occurring integrals.

$$f_p(x, v) = \frac{1}{N_p} \sum_{n=1}^{N_p} \delta(x - x_n) \delta(v - v_n) \frac{f_n}{g_n} \tag{B.176}$$

But first we recall the composition rule for delta functions. For a function $h(\xi) = x$ with a single root $h(\xi_0) = 0$ the composition is defined as

$$\delta(x) = (\delta \circ h)(\xi) = (\delta \circ h)(\xi) \frac{1}{\det(\nabla h(\xi_0))}, \tag{B.177}$$

which can also be derived from integration by substitution. Applying eqn. (B.177) onto eqn. (B.176) yields the transformed Klimontovich density $\hat{f}_p$ as

$$\hat{f}_p(\xi, v) = f_p(T(\xi), v) = \frac{1}{N_p} \sum_{n=1}^{N_p} \frac{\delta(T(\xi) - x_n)}{J_T(\xi_n)} \delta(v - v_n) \frac{f_n}{g_n} = \frac{1}{N_p} \sum_{n=1}^{N_p} \frac{\delta(\xi - \xi_n)}{J_T(\xi_n)} \delta(v - v_n) \frac{f_n}{g_n}. \tag{B.178}$$

By the change of coordinates it is known that

$$\int_{\mathbb{R}}\int_{\Omega} f(x,v)\psi(T^{-1}(x))\ \mathrm{d}x\mathrm{d}v = \int_{\mathbb{R}}\int_{[0,1]^d} \hat{f}(\xi,v)\psi(\xi)J_T(\xi)\ \mathrm{d}\xi\mathrm{d}v. \tag{B.179}$$

Hence, the consistency of the definition of $\hat{f}_p(\xi,v)$ is checked by verifying the integral eqn. (B.179) for the Klimontovich densities f_p and $\hat{f}_p$ reading

$$\int_{\mathbb{R}}\int_{\Omega} f_p(x,v)\psi(T^{-1}(x))\ \mathrm{d}x\mathrm{d}v = \int_{\mathbb{R}}\int_{[0,1]^d} \hat{f}_p(\xi,v)\psi(\xi)J_T(\xi)\ \mathrm{d}\xi\mathrm{d}v. \tag{B.180}$$

The right hand side of eqn. (B.180) is

$$\begin{aligned}\int_{\mathbb{R}}\int_{\Omega} f_p(x,v)\psi(T^{-1}(x))\ \mathrm{d}x\mathrm{d}v &= \int_{\mathbb{R}}\int_{\Omega} \frac{1}{N_p}\sum_{n=1}^{N_p}\delta(x-x_n)\delta(v-v_n)\frac{f_n}{g_n}\psi(T^{-1}(x))\ \mathrm{d}x\mathrm{d}v \\ &= \frac{1}{N_p}\sum_{n=1}^{N_p}\frac{f_n}{g_n}\psi(T^{-1}(x_n)).\end{aligned} \tag{B.181}$$

The left hand side of eqn. (B.180) reads

$$\begin{aligned}&\int_{\mathbb{R}}\int_{[0,1]^d} \hat{f}_p(\xi,v)\psi(T(\xi))J_T(\xi)\ \mathrm{d}\xi\mathrm{d}v \\ &\quad= \frac{f_n}{g_n}\int_{\mathbb{R}}\int_{[0,1]^d} \frac{1}{N_p}\sum_{n=1}^{N_p}\frac{\delta(\xi-\xi_n)}{J_T(\xi_n)}\delta(v-v_n)\psi(T(\xi))J_T(\xi)\frac{f_n}{g_n}\ \mathrm{d}\xi\mathrm{d}v \\ &\quad= \frac{1}{N_p}\sum_{n=1}^{N_p}\frac{f_n}{g_n}\psi(\underbrace{\xi_n}_{=T^{-1}(x_n)}) = \int_{\mathbb{R}}\int_{\Omega} f_p(x,v)\psi(x)\ \mathrm{d}x\mathrm{d}v,\end{aligned} \tag{B.182}$$

and coincides with eqn. (B.181) since the additional Jacobian cancels. This means given a set of markers or random deviates, the Monte Carlo approximation of an integral using these markers in a Klimontovich density is independent of the coordinate system, since the markers can be transformed freely and the Jacobian always cancels out. Particles methods get their attractiveness exactly from this property.
At last an example for this curvilinear sampling shall be given using the familiar polar coordinates for a disc or an annulus Ω. Let the logical domain be given as $\Omega_0 = [r_{\min}, r_{\max}]\times[0,2\pi]$.

$$T(r,\theta) = (r\cos(\theta), r\sin(\theta))\,,\ T^{-1}(x,y) = \left(\sqrt{x^2+y^2}, \arctan\left(\frac{y}{x}\right)\right),\ J_T(r,\theta) = r \tag{B.183}$$

We have a nontrivial Jacobian J_T and, therefore, different options of sampling. The volume of the annulus $T(\Omega_0) = \Omega$ is known to be

$$|\Omega| = \left(r_{\max}^2 - r_{\min}^2\right)\pi. \tag{B.184}$$

Hence when the particles should be uniformly distributed in the domain, the sampling density in Cartesian and logical coordinates reads

$$g(x,y) = \frac{1}{|\Omega|} = \hat{g}(r,\theta) \text{ for } (x,y)\in\Omega \text{ and } (r,\theta)\in\Omega_0. \tag{B.185}$$

But as we have seen before, $\hat{g}$ is not the probability density describing the distribution of the markers $(r_n,\theta_n)_{n=1,\dots,N_p}$ such that eqn. (B.174) has to be applied yielding

$$\tilde{g}(r,\theta) = \frac{J_T(r,\theta)}{|\Omega|}\frac{r}{\left(r_{\max}^2 - r_{\min}^2\right)\pi}. \tag{B.186}$$

With $\iint_\Omega \tilde{g}(r,\theta)drd\theta = 1$, $\tilde{g}$ becomes a probability density. We want to draw random numbers $(r_n, \theta_n)_{n=1,\dots,N_p}$ according to $\tilde{g}(r,\theta)$ and use inverse transform sampling since $\tilde{g}(r,\theta) = \tilde{g}(r)$ reduces to one dimension. Define the cumulative distribution function $G : [r_{\min}, r_{\max}] \to [0,1]$ as

$$G(\tau) = \int_0^{2\pi} \int_{r_{\min}}^{\tau} g(r,\theta)drd\theta = \frac{r^2 - r_{\min}^2}{r_{\max}^2 - r_{\min}^2}, \tag{B.187}$$

with the inverse

$$G^{-1}(u) = \sqrt{u(r_{\max}^2 - r_{\min}^2) + r_{\min}^2}, \quad u \in [0,1]. \tag{B.188}$$

Then the markers (r_n, θ_n) for $n = 1, \dots, N_p$ are obtained in the three steps:

1. Draw iid $\theta_n \sim \mathcal{U}(0, 2\pi)$,
2. Draw iid. $u_n \sim \mathcal{U}(0,1)$,
3. Set $r_n := G^{-1}(u_n) = \sqrt{u_n(r_{\max}^2 - r_{\min}^2) + r_{\min}^2}$.

Another option is to draw the markers uniformly in the logical domain according to

$$r_n \sim \mathcal{U}(r_{\min}, r_{\max}) \text{ and } \theta_n \sim \mathcal{U}(0, 2\pi), \tag{B.189}$$

which corresponds to the sampling density

$$\tilde{g}(r,\theta) = \frac{1}{|\Omega_0|} = \frac{1}{(r_{\max} - r_{\min})2\pi}. \tag{B.190}$$

But then the Jacobian enters into the Cartesian sampling density by definition, see eqn. (B.191).

$$\begin{aligned} \hat{g}(r,\theta) = \frac{\tilde{g}(r,\theta)}{J_T(r,\theta)} &= \frac{1}{(r_{\max} - r_{\min})2\pi} \frac{1}{r} \\ \Rightarrow g(x,y) &= \frac{1}{(r_{\max} - r_{\min})2\pi} \frac{1}{\sqrt{x^2 + y^2}} \end{aligned} \tag{B.191}$$

This yields more markers for small r and does not cancel with the Jacobian of the coordinate transformation and is, therefore, a rather unnatural way of sampling.

Appendix C.

Spectral methods and particle discretizations

C.1. Orthogonal Polynomials

For any Vlasov solver, derivatives, anti-derivative and various methods for evaluation are needed, such that the we collect and review essential formulas from the enormous complex of spectral methods that are most useful for implementation.

C.1.1. Chebyshev

The Chebyshev polynomials of second kind are defined by

$$\begin{aligned} U_0(x) &= 1 \\ U_1(x) &= 2x \\ U_{n+1} &= 2xU_n(x) - U_{n-1}(x). \end{aligned} \tag{C.1}$$

The derivative of the first kind Chebyshev polynomial can be obtained by

$$\frac{d}{dx}T_n(x) = nU_{n-1}(x), \quad \forall n \geq 1 \tag{C.2}$$

or directly by ([195][p.47])

$$\frac{d}{dx}T_n(x) = \frac{n}{2}\frac{T_{n-1}(x) - T_{n+1}(x)}{1-x^2}. \tag{C.3}$$

The Ultra-spherical polynomials U_n can be used to obtain efficient pre-conditioners leading to sparse methods, see [171]. We take advantage of this rather involved computations by using the *ApproxFun.jl* package [171]. The two term recurrence relation for the first derivative of T_n' reads

$$\begin{aligned} T_0'(x) &= 0 \\ T_1'(x) &= 1 \\ T_2'(x) &= 4x \\ T_{n+1}'(x) &= 2x\frac{n+1}{n}T_n'(x) - \frac{n+1}{n-1}T_{n-1}'(x). \end{aligned} \tag{C.4}$$

These two term recurrence relations emerge from the Chebyshev identity (3.71) and the substitution $x = \cos(\theta)$, where $\mathrm{d}x = -\sin(\theta)\mathrm{d}\theta$.

$$T_n(\cos(\theta)) = \cos(n\theta), \quad T_n'(\cos(\theta)) = \left[\frac{\mathrm{d}}{\mathrm{d}\theta}T_n(\cos(\theta))\right]\cdot(-\sin(\theta)) = \frac{\sin(n\theta)n}{\sin(\theta)} \tag{C.5}$$

Unfortunately it is not possible to derive a two term recurrence formula for the second derivative. We define the indefinite integral $R_n(x)$ with constant zero to T_n as

$$R_n(x) := \int T_n(x)\mathrm{d}x = \frac{1}{2}\left(\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1}\right) = \frac{nT_{n+1}(x)}{n^2-1} - \frac{xT_n(x)}{n-1}. \tag{C.6}$$

The indefinite integral R_n can be expressed by the Chebyshev polynomials T_n, which allows for integration of a Chebyshev series directly over the coefficients.

$$R_n(\cos(\theta)) = \int \cos(n\theta)(-\sin(\theta))\mathrm{d}\theta = \begin{cases} -\frac{\sin(\theta)^2}{2} & \text{if } n = 1 \\ \frac{\cos(\theta(n+1))}{2(n+1)} - \frac{\cos(\theta(n-1))}{2(n-1)} & \text{else} \end{cases} \tag{C.7}$$

$$\int_{-1}^{1} T_n(x)\ \mathrm{d}x = \begin{cases} 0 & \text{for } n = 1 \\ \frac{(-1)^n+1}{1-n^2} & \text{else} \end{cases} \tag{C.8}$$

In most cases one wants to evaluate a linear combination of Chebyshev polynomials. The standard method for numerically stable evaluation of polynomials is the Horner method, which is a special case of the Clenshaw algorithm. Because of their two term recurrence relation, the Clenshaw algorithm is directly applicable to the Chebyshev polynomials. We proceed to work on the unit interval $[0, 1]$ with the shifted Chebyshev polynomials of the first kind

$$T_n^*(x) = T_n(2x + 1) \text{ and } T_n^{*\prime}(x) = 2T_n'(2x + 1). \tag{C.9}$$

The recurrence relation for their evaluation reads

$$\begin{aligned} T_0^*(x) &= 1 \\ T_1^*(x) &= 2x - 1 \\ T_{n+1}^*(x) &= 2(2x - 1)T_n^*(x) - T_{n-1}^*(x) \\ &\cdot \end{aligned} \tag{C.10}$$

$$\begin{aligned} T_0^{*\prime}(x) &= 0 \\ T_1^{*\prime}(x) &= 2 \\ T_1^{*\prime}(x) &= 16x - 8 \\ T_n^{*\prime}(x) &= 2(2x - 1)\frac{n}{n-1}T_{n-1}^{*\prime}(x) - \frac{n}{n-2}T_{n-2}^{*\prime}(x). \end{aligned} \tag{C.11}$$

Alternative formulas for the first and second derivatives are given in eqn.(C.12).

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x}T_n(x) &= \frac{\frac{1}{2}n\left(T_{n-1}(x) + T_{n+1}(x)\right)}{1 - x^2} \\ \frac{\mathrm{d}^2}{\mathrm{d}x^2}T_n(x) &= \frac{n}{4}\frac{(n+1)T_{n-2}(x) - 2nT_n(x) + (n-1)T_{n+2}(x)}{(1-x^2)^2} \end{aligned} \tag{C.12}$$

When evaluating the indefinite integral or the derivatives of Chebyshev series, it suffices to calculate a new set of coefficients and express the integral again as Chebyshev series.

$$u(x) = \sum_{n=0}^{N} u_n T_n(x) \tag{C.13}$$

For the indefinite integral we set the integration constant $U_0 = 0$ to zero and suppose $u_n = 0\ \forall n > N$.

$$\begin{aligned} U(x) = \int u(x)\ \mathrm{d}x &= \sum_{n=0}^{N+1} U_n T_n(x) \\ U_n &= \frac{u_{n-1} - u_{n+1}}{2n} \text{ for } n > 1 \\ U_N &= \frac{u_{N-1}}{2N} \\ U_{N+1} &= \frac{u_N}{2(N+1)} \\ U_0 &= 0 \end{aligned} \tag{C.14}$$

The same can be done for the first derivative.

$$\begin{aligned} u'(x) &= \sum_{n=0}^{N} u'_n T_n(x) \\ u'_N &= 0 \\ u'_{N-1} &= 2N u_N \\ u'_n &= 2(n+1)u_{n+1} + u'_{n+2} \\ u'_0 &= u_1 + u'_2 \end{aligned} \tag{C.15}$$

Note the additional factor 2 for shifted Chebyshev polynomials. Successive application of the recursion (C.15) yields a formula for the second derivative.
For evaluation of a series of orthogonal polynomials $(\Phi_k)_{k=0,\dots,n}$,

$$S(x) = \sum_{k=0}^{n} c_k \Phi_k(x), \tag{C.16}$$

which follows the two term recurrence

$$\Phi_{k+1}(x) = \alpha_k(x)\Phi_k + \beta_k(x)\Phi_{k-1}(x), \tag{C.17}$$

the Clenshaw algorithm can be used, which was first described in [254].

$$\begin{aligned} b_{n+1} &= b_{n+2} = 0 \\ b_k(x) &= c_k + \alpha_k b_{k+1}(x) + \beta_{k+1} b_{k+2}(x) \quad \forall\ k = 0, \dots, n. \end{aligned} \tag{C.18}$$

$$S(x) = \Phi_0(x)c_0 + \Phi_1(x)b_1(x) + \beta_1\Phi_0(x)b_2(x) \tag{C.19}$$

The Clenshaw algorithm for a Chebyshev series and its first derivative is given in eqn. (C.20) and eqn. (C.21).

$$\begin{aligned} b_{N+1} &:= 0 \text{ and } b_{N+2} := 0 \\ b_n &:= u_n + 2x b_{n+1} - b_{n+2}, \quad n = N, \dots, 1 \\ u(x) &= \sum_{n=0}^{N} u_n T_n(x) = u_0 + x b_1 - b_2 \end{aligned} \tag{C.20}$$

$$\begin{aligned} b_{N+1} &:= 0 \text{ and } b_{N+2} := 0 \\ b_n &:= u_n + 2x\frac{n+1}{n} b_{n+1} - \frac{n+2}{n} b_{n+2}, \quad n = N, \dots, 2 \\ u(x) &= \sum_{n=0}^{N} u_n T'_n(x) = u_1 + 4x b_2 - 3b_3 \end{aligned} \tag{C.21}$$

Now that we treated several possibilities for the efficient evaluation of Chebyshev polynomials, we continue with solving differential equations. The first candidate is, of course, the Poisson equation with Neumann and Dirichlet boundary conditions. The two standard approaches are either the collocation or the Galerkin method, where we chose the latter since it fits perfectly in the variational particle framework.
The Fourier modes satisfy the periodic boundary condition and the B-spline finite elements can satisfy Dirichlet and Neumann boundary conditions upon construction. Hence the boundary conditions are naturally built into the basis functions. The Chebyshev polynomials have different values at the boundary

$$T_k(\pm 1) = (\pm 1)^k \text{ and } T'_k(\pm 1) = (\pm 1)^{k+1} k^2, \tag{C.22}$$

but we can construct, similar to [255, 165], basis functions ψ_k composed of Chebyshev polynomials, which satisfy the inhomogeneous Robin boundary conditions.

$$a_\pm \psi_k(\pm 1) + b_\pm \psi_k'(\pm 1) = c_\pm \tag{C.23}$$

The inhomogeneous Robin boundary conditions are general enough such that homogeneous Dirichlet or Neumann are just special cases. Following [165][pp. 202], we seek coefficients a_k and b_k such that the basis functions ψ_k can be expressed as a linear combination of Chebyshev polynomials

$$\psi_k(x) = T_k(x) + a_k T_{k+1}(x) + b_k T_{k+2}(x). \tag{C.24}$$

Inserting eqn. (C.22) into eqn. (C.24) results in

$$\begin{aligned} \psi_k(\pm 1) &= (\pm 1)^k + a_k(\pm 1)^{k+1} + b_k(\pm 1)^{k+2} \\ \psi_k'(\pm 1) &= (\pm 1)^{k+1}k^2 + a_k(\pm 1)^{k+2}(k+1)^2 + b_k(\pm 1)^{k+3}(k+2)^2. \end{aligned} \tag{C.25}$$

This yields the following linear system for a_k and b_k

$$\begin{aligned} \left(a_\pm(\pm 1)^{k+1} + b_\pm(\pm 1)^{k+2}(k+1)^2\right) a_k + \left(a_\pm(\pm 1)^{k+2} + b_\pm(\pm 1)^{k+3}(k+2)^2\right) b_k \\ = -a_\pm(\pm 1)^k - b_\pm(\pm 1)^{k+1}k^2 + c_\pm, \end{aligned} \tag{C.26}$$

$$\left(\pm a_\pm + b_\pm(k+1)^2\right) a_k + \left(a_\pm \pm b_\pm(k+2)^2\right) b_k = -a_\pm \mp b_\pm k^2 + c_\pm(\pm 1)^{-k}. \tag{C.27}$$

We rewrite the 2×2-system in matrix form and obtain a solution by the direct inverse.

$$\underbrace{\begin{pmatrix} a_+ + b_+(k+1)^2 & a_+ + b_+(k+2)^2 \\ -a_- + b_-(k+1)^2 & a_- - b_-(k+2)^2 \end{pmatrix}}_{:=\Gamma} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} -a_+ - b_+k^2 + c_+ \\ -a_- + b_-k^2 + c_-(-1)^k \end{pmatrix} \tag{C.28}$$

$$\tag{C.29}$$

$$\begin{aligned} \gamma_k &= \det(\Gamma) = 2a_+a_- + (k+1)^2(k+2)^2\left(a_-b_+ - a_+b_- - 2b_-b_+\right) \\ a_k &= -\frac{1}{\gamma_k}\Big[\left(a_+ + b_+(k+2)^2\right)\left(-a_- + b_-k^2 + c_-(-1)^k\right) \\ &\qquad - \left(a_- - b_-(k+2)^2\right)\left(-a_+ - b_+k^2 + c_+\right)\Big] \\ b_k &= \frac{1}{\gamma_k}\Big[\left(a_+ + b_+(k+1)^2\right)\left(-a_- + b_-k^2 + c_-(-1)^k\right) \\ &\qquad + \left(a_- - b_-(k+1)^2\right)\left(-a_+ - b_+k^2 + c_+\right)\Big] \end{aligned} \tag{C.30}$$

We transform this result for shifted Chebyshev polynomials T^* and define the Galerkin basis function as

$$\psi_k(x) = T_k^*(x) + a_k T_{k+1}^*(x) + b_k T_{k+2}^*(x). \tag{C.31}$$

The boundary values of the polynomials only change by a factor 2 in the first derivative, see eqn. (C.32).

$$T_k^*(\{0,1\}) = (\pm 1)^k \text{ and } T_k^{*\prime}(\{0,1\}) = 2(\pm 1)^{k+1}k^2, \tag{C.32}$$

Hence, a modification of eqn. (C.30) yields the coefficients for the basis functions (C.31) on $[0,1]$.

$$\begin{aligned}
\gamma_k &= 2a_+a_- + 2(k+1)^2(k+2)^2\left(a_-b_+ - a_+b_- - 4b_-b_+\right)\\
a_k &= -\frac{1}{\gamma_k}\Big[\left(a_+ + 2b_+(k+2)^2\right)\left(-a_- + 2b_-k^2 + c_-(-1)^k\right)\\
&\qquad - \left(a_- - 2b_-(k+2)^2\right)\left(-a_+ - 2b_+k^2 + c_+\right)\Big]\\
b_k &= \frac{1}{\gamma_k}\Big[\left(a_+ + 2b_+(k+1)^2\right)\left(-a_- + 2b_-k^2 + c_-(-1)^k\right)\\
&\qquad + \left(a_- - 2b_-(k+1)^2\right)\left(-a_+ - 2b_+k^2 + c_+\right)\Big]
\end{aligned} \tag{C.33}$$

The Galerkin basis $\{\psi_n\}_{n=0}^N$ is composed of the shifted Chebyshev polynomials $\{T_n^*\}_{n=0}^{N+2}$. The operator projecting from the shifted Chebyshev polynomials onto the Galerkin basis is denoted by the $(N+1)\times(N+3)$ matrix

$$S_x = \begin{pmatrix} 1 & a_0 & b_0 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & a_N & b_N \end{pmatrix}. \tag{C.34}$$

Because the spectral Galerkin method usually leads to dense matrices we highly recommend Jie Shens series on efficient spectral-Galerkin methods [172, 173, 174, 175], where he obtains sparse or full and banded matrices, which are mostly solved in $\mathcal{O}(N)$. Of particular interest for us is the Helmholtz equation in cylindrical geometry [174] and in spherical geometry [175]. This shows that spectral methods based on Fourier-Chebyshev polynomials are very efficient even for complex geometries. Unfortunately the torus was not treated, but since it resides somewhere between cylinder and sphere it should be possible to obtain similar results.
Boyd [54][p.389] has strong objections to using shifted Chebyshev polynomials $T_k^*(r) = T_k(2r-1)$ as a basis in radial direction since $T_k^*(r^2) = T_k(2r^2-1)$ yields a much better approximation of the Bessel function J_0. We note that this is due to the change of volume in the polar plane, which in general can be calculated by the Jacobi determinant of the coordinate transformation.

$$T(r,\theta) = (r\cos(\theta), r\sin(\theta)) \Rightarrow \det(\nabla T(r,\theta)) = r \tag{C.35}$$

Thus, an additional normalization of the basis functions by an additional change of coordinates might yield much faster convergence and should therefore be considered.
In many cases intervals are given as $[0,L]$, which can be mapped to the standard interval $[-1,1]$ by

$$\varphi : [0,L] \to [-1,1], \quad x \mapsto \frac{2x}{L} - 1, \quad \varphi' = \frac{2}{L}. \tag{C.36}$$

For the L^2-projection and derivatives, coefficients are multiplied by φ'.

Fast Chebyshev transform

A major advantage of the Chebyshev polynomials is that given an appropriate grid the transform from function values on the grid to the Chebyshev coefficients using the $\mathcal{O}(N\log(N))$ discrete cosine transform can benefit from the fast Fourier transform. The fast type-I discrete cosine transform $DCT-I$ is implemented in $FFTW$, see [256], for a length N array $(X_k)_{k=0,\ldots,N-1}$ in eqn. (C.37).

$$Y_k = X_0 + (-1)^k X_{N-1} + 2\sum_{j=1}^{N-2} X_j \cos\left(\frac{\pi jk}{N-1}\right), \quad k = 0,\ldots,N-1 \tag{C.37}$$

There is the option of choosing the Chebyshev-Gauss quadrature nodes in $(-1, 1)$, but since we want to handle boundary conditions on a collocation basis we chose Chebyshev Gauss–Lobatto points given in eqn. (C.38) as the spatial grid.

$$x_j = \cos\left(\frac{\pi j}{N-1}\right), \quad j = 0, \dots, N-1 \tag{C.38}$$

Values of the Chebyshev polynomials at these nodes are then directly expressed in eqn. (C.39).

$$T_k(x_j) = T_k\left(\cos\left(\frac{\pi j}{N-1}\right)\right) = \cos\left(\frac{\pi j \cdot k}{N-1}\right) \tag{C.39}$$

The entire complex of forward and back transforming with the discrete Chebyshev transform for a function $u : [-1, 1] \to \mathbb{R}$ is prepared in eqn. (C.40). When using the type-I discrete cosine transform in eqn. (C.37) the forward transform is normalized by $N - 1$ and the backward transform by 2.

$$\begin{aligned}
u(x) &\approx \sum_{k=0}^{N-1} \tilde{u}_j T_k(x) \\
u_j &:= u(x_j) \\
u_j &= \sum_{k=0}^{N-1} \tilde{u}_j \cos\left(\frac{\pi j \cdot k}{N-1}\right) = \tilde{u}_j + 2\sum_{k=0}^{N-1} \tilde{u}_j \cos\left(\frac{\pi j \cdot k}{N-1}\right) \\
\tilde{u}_k &= \frac{1}{(N-1)c_k} \sum_{j=0}^{N-1} \frac{2}{c_j} u_j \cos\left(\frac{\pi j \cdot k}{N-1}\right) = \frac{1}{(N-1)c_k}\left[u_0 + (-1)^k u_{N-1} + 2\sum_{j=1}^{N-2} u_j \cos\left(\frac{\pi j \cdot k}{N-1}\right)\right] \\
c_j &= \begin{cases} 2 & \text{for } j = 0, N-1 \\ 1 & \text{for } j = 1, \dots, N-2 \end{cases}
\end{aligned} \tag{C.40}$$

Poisson equation on bounded domain

Before we proceed directly with curvilinear coordinates and mappings we want to solve the Poisson equation with Robin boundary conditions on the domain $[0, 1]$ using Chebyshev polynomials.

$$\begin{aligned}
-\Delta\Phi &= \rho \\
a_-\Phi(0) + b_-\Phi'(0) &= c_- \\
a_+\Phi(1) + b_+\Phi'(1) &= c_+
\end{aligned} \tag{C.41}$$

For the variational formulation we use the $\mathcal{L}^2$ scalar product with a weight function ω and test functions φ satisfying the boundary conditions.

$$\int_0^1 \Phi'(x)\left(\varphi(x)\omega(x)\right)' \,\mathrm{d}x = \int_0^1 \rho(x)\varphi(x)\omega(x)\,\mathrm{d}x, \qquad \forall\varphi \tag{C.42}$$

$$\int_0^1 \Phi'(x)\left(\varphi(x)'\omega(x) + \omega(x)'\varphi(x)\right) \,\mathrm{d}x = \int_0^1 \rho(x)\varphi(x)\omega(x)\,\mathrm{d}x, \qquad \forall\varphi \tag{C.43}$$

$$\begin{aligned}
\omega(x) &= \sqrt{1-x^2} \\
\omega'(x) &= \frac{-x}{\sqrt{1-x^2}} \\
\omega(\cos(\theta)) &= \sin(\theta) \\
\omega'(\cos(\theta)) &= -\frac{\cos(\theta)}{\sin(\theta)}
\end{aligned} \tag{C.44}$$

We substitute with $x = \cos(\theta)$, and $\mathrm{d}x = \mathrm{d}\theta(-\sin(\theta))$ yielding

$$\begin{aligned}\int_{-1}^{1} T_i'(x)T_j(x)\omega'(x)\mathrm{d}x &= -\int_{\pi}^{0} \frac{\sin(i\theta)i}{\sin(\theta)}\cos(j\theta)\ \frac{-\cos(\theta)}{\sin(\theta)}\ \ \sin(\theta)\mathrm{d}\theta \\ &= -i\int_{0}^{\pi} \frac{\sin(i\theta)\cos(j\theta)\cos(\theta)}{\sin(\theta)}\mathrm{d}\theta \\ &= -i\begin{cases} 0 & \text{if } i = 0, \\ \frac{\pi}{2} & \text{if } i = j, \\ \pi & \text{if } i < j, i \text{ even}, j \text{ uneven}, \\ 0 & \text{else.}\end{cases}\end{aligned} \tag{C.45}$$

This results in a banded Toeplitz matrix, which can be solved very efficiently. For a suitable weight function ω Chebyshev polynomials are orthogonal, thus we denote some useful relations from [54].

$$\begin{aligned}\int_{-1}^{1} T_i(x)T_j(x)\frac{1}{\sqrt{1-x^2}}\ \mathrm{d}x &= \delta_{i,j}\begin{cases}\pi \text{ for } i = 0 \\ \frac{\pi}{2} \text{ else.}\end{cases} \\ \int_{-1}^{1} U_i(x)U_j(x)\sqrt{1-x^2}\ \mathrm{d}x &= \delta_{i,j}\frac{\pi}{2} \\ \int_{-1}^{1} T_i'(x)T_j'(x)\sqrt{1-x^2}\ \mathrm{d}x &= \delta_{(i-1),(j-1)}\frac{\pi}{2}(ij)\end{aligned} \tag{C.46}$$

$$\begin{aligned}\int_{0}^{1} T_i^*(x)T_j^*(x)\frac{1}{2\sqrt{x(x-1)}}\ \mathrm{d}x &= \delta_{i,j}\begin{cases}2\pi \text{ for } i = 0 \\ \pi \text{ else}\end{cases} \\ \int_{0}^{1} T_i^*(x)T_j^*(x)2\sqrt{x(x-1)}\ \mathrm{d}x &= \delta_{(i-1),(j-1)}\pi(ij)\end{aligned} \tag{C.47}$$

If our method is extended to the stretched domain $[0, L]$, the modified potential reads

$$\tilde{\Phi} : [0, L] \to \mathbb{R},\ \tilde{\Phi}(x) = \Phi\left(\frac{x}{L}\right)\ \Rightarrow \tilde{\Phi}'(x) = \Phi'\left(\frac{x}{L}\right)\frac{1}{L}. \tag{C.48}$$

C.1.2. Hermite functions for unbounded domains

Until now we have only treated bounded or periodic domains, but spectral methods are also very efficient on unbounded $[-\infty, \infty]$ and half open $[0, \infty]$ domains. For unbounded domains Hermite polynomials and for half open domains Laguerre polynomials can be used. The Galerkin mechanism is completely analog to the Chebyshev polynomials and it is straightforward to implement. There are plenty plasma physics applications that can benefit from the modeling of an open interval. For example, in high energy beam physics boundary conditions pose a problem, since the longitudinal model follows the moving beam in a reference frame [257], [258]. It is unphysical to make this frame periodic or bounded, although much simpler from a computational viewpoint. Therefore, we present a brief recipe for an unbounded electrostatic case. We follow [165][*Chapter 4, Spectral Methods in Unbounded Domains*], which provides a quick overview and also includes the Laguerre polynomials suited for semi-infinite intervals $[0, \infty)$. The normalized Hermite functions $\hat{H}_n$ of degree n defined in eqn. (C.49) are chosen as basis functions. They consist of a Gaussian weight, the normalization and the Hermite polynomials defined in eqn. (C.50). Hence, they are very well suited for an approximation of a localized chunk of plasma in open space.

$$\hat{H}(x) = \frac{1}{\sqrt{2^n n!}}\mathrm{e}^{-\frac{x^2}{2}}H_n(x),\ \text{for } n \geq 0,\ x \in \mathbb{R} \tag{C.49}$$

The physicists' Hermite polynomials are just as the Chebyshev polynomials defined by a three term recurrence relation (C.50).

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_n(x) &= 2xH_n(x) - 2nH_{n-1}(x) \end{aligned} \tag{C.50}$$

Yet this is not the numerically stable way of evaluating the Hermite functions and, therefore, we can use a recurrence relation for $\hat{H}_n$ provided by [165][p.146], which reads

$$\begin{aligned} \hat{H}_0(x) &= \mathrm{e}^{-\frac{x^2}{2}} \\ \hat{H}_1(x) &= \sqrt{2}x\mathrm{e}^{-\frac{x^2}{2}} \\ \hat{H}_{n+1}(x) &= x\sqrt{\frac{2}{n+1}}\hat{H}_n(x) - \sqrt{\frac{n}{n+1}}\hat{H}_{n-1}(x). \end{aligned} \tag{C.51}$$

Note that the expensive e-function appearing in eqn. (C.49) is actually only needed once in the recurrence relation (C.51) and should not be applied afterwards, since it damps the diverging Hermite polynomials. Thus, eqn. (C.51) should be used in an implementation. The derivatives of a normalized Hermite function provided in eqn. (C.52) and (C.53) are adapted from [165][p.146].

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x}\hat{H}_0(x) &= -\frac{1}{\sqrt{2}}\hat{H}_1(x) \\ \frac{\mathrm{d}}{\mathrm{d}x}\hat{H}_n(x) &= \hat{H}_n'(x) = \sqrt{\frac{n}{2}}\hat{H}_{n-1}(x) - \sqrt{\frac{n+1}{2}}\hat{H}_{n+1}(x) \end{aligned} \tag{C.52}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\hat{H}_n(x) = \frac{\sqrt{n(n-1)}}{2}\hat{H}_{n-2}(x) - \frac{n+1}{2}\hat{H}_n(x) + \frac{\sqrt{(n+1)(n+2))}}{2}\hat{H}_{n+2}(x) \tag{C.53}$$

Here a Hermite function series and its derivative are denoted in eqn. (C.54).

$$u(x) = \sum_n u_n\hat{H}_n(x), \quad u'(x) = \sum_n u_n'\hat{H}_n(x). \tag{C.54}$$

The derivative of a Hermite function series itself is again a Hermite function series and the coefficients u_n' are obtained by

$$u_n' = -\sqrt{\frac{n}{2}}\ u_{n-1} - \sqrt{\frac{n+1}{2}}\ u_{n+1}. \tag{C.55}$$

In order to reconstruct the charge density or solve the Poisson equation, we can use the sparse mass (C.56) and stiffness matrices (C.57) provided in [165].

$$\int_{\mathbb{R}} \hat{H}_n(x)\hat{H}_m(x)\ \mathrm{d}x = \sqrt{\pi}\delta_{m,n} \tag{C.56}$$

$$\int_{\mathbb{R}} \hat{H}_n'(x)\hat{H}_m'(x) = \begin{cases} -\frac{\sqrt{n(n-1)\pi}}{2} & \text{for } m = n-2 \\ \sqrt{\pi}\left(n+\frac{1}{2}\right) & \text{for } m = n \\ -\frac{\sqrt{(n+2)(n+1)\pi}}{2} & \text{for } m = n+2 \\ 0 & \text{else} \end{cases} \tag{C.57}$$

For assembling a constant Galerkin right hand side, eqn. (C.58) resulting in eqn. (C.59) is helpful.

$$\int_{-\infty}^{\infty} \mathrm{e}^{-\frac{x^2}{2}} \hat{H}_n(x)\,\mathrm{d}x = \begin{cases} \frac{n!}{(n/2)!}\sqrt{2\pi} & \text{for } n \text{ even} \\ 0 & \text{for } n \text{ odd} \end{cases} \tag{C.58}$$

$$\int_{-\infty}^{\infty} \hat{H}_n(x)\,\mathrm{d}x = \begin{cases} \frac{\sqrt{n!}}{(n/2)!\sqrt{2}^n}\sqrt{2\pi} & \text{for } n \text{ even} \\ 0 & \text{for } n \text{ odd} \end{cases} \tag{C.59}$$

Therefore, the integral over a Hermite series requires only every second coefficient, see eqn. (C.60).

$$\int_{-\infty}^{\infty} \sum_n u_n \hat{H}_n(x)\,\mathrm{d}x = \sum_n u_{2n} \frac{\sqrt{(2n)!}}{n!2^n}\sqrt{2\pi}. \tag{C.60}$$

Efficient evaluation of a Hermite function series is achieved by the Clenshaw algorithm in eqn. (C.61).

$$\begin{aligned} b_{N+1} &:= 0 \text{ and } b_{N+2} := 0 \\ b_n &:= u_n + x\sqrt{\frac{2}{n+1}}b_{n+1} - \sqrt{\frac{n+1}{n+2}}b_{n+2}, \quad n = N, \ldots, 0 \\ u(x) &= \sum_{n=0}^{N} u_n \hat{H}_n(x) = e^{\frac{-x^2}{2}} b_0 \end{aligned} \tag{C.61}$$

C.1.3. Legendre polynomials

Using the Clenshaw algorithm [254] on (3.74) yields the efficient and numerically stable evaluation of a Legendre series in eqn. (C.62).

$$\begin{aligned} b_{N+1} &:= 0 \text{ and } b_{N+2} := 0 \\ b_n &:= u_n + x\frac{2n+1}{n+1}b_{n+1} - \frac{n+1}{n+2}b_{n+2}, \quad n = N, \ldots, 0 \\ u(x) &= \sum_{n=0}^{N} u_n P_n(x) = b_0 \end{aligned} \tag{C.62}$$

Here, we work with a Legendre series where the derivative or the anti-derivative can be expressed again as a Legendre series, see eqns. (C.63) and (C.64).

$$P_n(x) = \frac{1}{2n+1}\left[P'_{n+1}(x) - P'_{n-1}(x)\right] \tag{C.63}$$

$$P'_{n+1}(x) = \sum_{0 \le k \le \frac{n}{2}} (2(n-2k)+1)P_{n-2k}(x) \tag{C.64}$$

For the indefinite integral we set the integration constant $U_0 = 0$ to zero, define $u_n = 0$, $\forall\, n > N$ and use (C.63) to obtain eqn. (C.65). The coefficients in (C.65) and (C.66) are also found

in [54][pp.500-501].

$$\begin{aligned} U(x) &= \int u(x)\ \mathrm{d}x = \sum_{n=0}^{N+1} U_n P_n(x) \\ U_n &= \frac{1}{2n-1}u_{n-1} - \frac{1}{2n+3}u_{n+1} \text{ for } n > 1 \\ U_N &= \frac{1}{2N-1}u_{N-1} \\ U_{N+1} &= \frac{1}{2N+1}u_N \\ U_0 &= 0 \end{aligned} \tag{C.65}$$

The derivative is quite involved and not as straightforward as the integral, here [54] provides us eqn. (C.66).

$$\begin{aligned} u'(x) &= \sum_{n=0}^{\infty} u'_n P_n(x) \\ u'_n &= (2n+1) \sum_{p=n+1, p+n \text{ odd}}^{\infty} u_p \end{aligned} \tag{C.66}$$

Equation (C.66) is rewritten into eqn. (C.67) by introducing a series (a_n), which is the reversed cumulative sum of the Legendre coefficients c_n of odd or even index n.

$$\begin{aligned} u'(x) &= \sum_{n=0}^{N} u'_n P_n(x) \\ a_{2k-2} &= a_{2k} + u_{2k-2} \\ a_{2k-1} &= a_{2k+1} + u_{2k-1} \\ u'_n &= (2n+1)a_{n+1} \\ a_{N+1} &= a_{N+2} = 0 \end{aligned} \tag{C.67}$$

Following [172] the weak Poisson equation with Dirichlet boundary conditions can be solved efficiently with Legendre polynomials by defining basis functions

$$\psi_n(x) = \frac{1}{\sqrt{4n+6}} \left(P_n(x) - P_{n+2}(x)\right), \quad \psi_n(x). \tag{C.68}$$

The basis (ψ_n) satisfies the homogeneous Dirichlet condition $\psi_n(\pm 1) = 0$ because $P_n(\pm 1) = (\pm 1)^n$. The derivative reduces with eqn. (C.63) to a Legendre polynomial

$$\psi'_n(x) = \frac{1}{\sqrt{4n+6}}(2(n+1)+1)P_{n+1}(x) = \sqrt{n+\frac{3}{2}}\ P_{n+1}(x), \tag{C.69}$$

which means that the derivative of a function expressed in the basis (ψ_n) can be directly evaluated as a Legendre series by (C.62) avoiding (C.66). Using the orthogonality (3.75) it becomes clear that on this basis the weak Poisson solve with homogeneous Dirichlet boundary conditions is trivial since

$$\int_{-1}^{1} \psi'_n(x)\psi'_m(x)\ \mathrm{d}x = -\int_{-1}^{1} \psi''_n(x)\psi_m(x)\ \mathrm{d}x = \delta_{n,m}. \tag{C.70}$$

It remains to conclude, that we can obtain in Euclidean space the L^2 projection (3.75), the integral (C.65) and the derivative (C.66) in $\mathcal{O}(N)$.

In polar coordinates the Jacobian r enters the L^2 projection destroying the orthogonality in eqn. (3.75). For this, general formulas like eqn. (C.71) and eqn. (C.72) are useful.

$$\begin{aligned} u(x) &= \sum_{n=0}^{N} u_n P_n(x) \\ xu(x) = v(x) &= \sum_{n=0}^{N} v_n P_n(x) \\ v_0 &= 0 \\ v_n &= \frac{n}{2n-1} u_{n-1} + \frac{n+1}{2n+1} u_{n+1}, \quad n \geq 1 \end{aligned} \tag{C.71}$$

$$\int_{-1}^{1} x P_n(x) P_m(x) = \begin{cases} \frac{2n+1}{(2n+1)(2n+3)} & \text{for } m = n+1 \\ \frac{2n}{(2n-1)(2n+1)} & \text{for } m = n-1 \end{cases} \tag{C.72}$$

C.2. Complex to real transforms for PIF

Fourier coefficients form a set of complex conjugates which means, that half of the coefficients Fourier are just a redundant replication. In order to always gain a factor of two, one of the following formulas can be used. Although this is very basic math we state the different options here explicitly because they are very helpful for the implementation. Let $f(\varphi) : [0, 2\pi] \to \mathbb{R}$ be given. We want to approximate f by $\hat{f}$ with N_φ modes. One calculates the coefficient vector as

$$\hat{F}(n) = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{e}^{-\mathrm{i}n\varphi} f(\varphi) \, \mathrm{d}\varphi, \quad \text{for all } n = -N_\varphi, \ldots, 0, \ldots N_\varphi \tag{C.73}$$

and receives the reconstructed density as

$$\hat{f}(\varphi) = \sum_{n=-N_\varphi}^{N_\varphi} \hat{F}(n) \mathrm{e}^{\mathrm{i}n\varphi} \approx f(\varphi). \tag{C.74}$$

Now we perform a lengthy calculation to show that we only need to calculate and save $\hat{F}(n)$ for $n = 0, \ldots, N_\varphi$. First we note that

$$\begin{aligned} \overline{\hat{F}(-n)} &= \frac{1}{2\pi} \int_0^{2\pi} \overline{\mathrm{e}^{-\mathrm{i}(-n)\varphi} \underbrace{f(\varphi)}_{\in \mathbb{R}}} \, \mathrm{d}\varphi = \frac{1}{2\pi} \int_0^{2\pi} \overline{\mathrm{e}^{\mathrm{i}n\varphi}} f(\varphi) \, \mathrm{d}\varphi = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{e}^{\mathrm{i}n\varphi} f(\varphi) \, \mathrm{d}\varphi \\ &= \hat{F}(n), \qquad \text{for all } n = -N_\varphi, \ldots, 0, \ldots N_\varphi. \end{aligned} \tag{C.75}$$

Then we want to use this identity to simplify

$$\begin{aligned}
\hat{f}(\varphi) &= \sum_{n=-N_\varphi}^{-1} \hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi}\hat{F}(0) + \sum_{n=1} \hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi} \\
&= \hat{F}(0) + \sum_{n=1} \hat{F}(-n)\mathrm{e}^{-\mathrm{i}n\varphi} \;+\; \hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi} \\
&= \hat{F}(0) + \sum_{n=1} \overline{\hat{F}(n)}\mathrm{e}^{-\mathrm{i}n\varphi} \;+\; \hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi} \\
&= \hat{F}(0) + \sum_{n=1} \overline{\hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi}} \;+\; \hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi} \\
&= \hat{F}(0) + \sum_{n=1} 2\mathfrak{R}\left(\hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi}\right) \;+\; 2\mathfrak{I}\left(\hat{F}(n)\mathrm{e}^{\mathrm{i}n\varphi}\right) \\
&= \hat{F}(0) + 2\sum_{n=1} \mathfrak{R}\left(\hat{F}(n)\right)\mathfrak{R}\left(\mathrm{e}^{\mathrm{i}n\varphi}\right) \;-\; \mathfrak{I}\left(\hat{F}(n)\right)\mathfrak{I}\left(\mathrm{e}^{\mathrm{i}n\varphi}\right) \\
&= \hat{F}(0) + 2\sum_{n=1} \mathfrak{R}\left(\hat{F}(n)\right)\cos(n\varphi) \;-\; \mathfrak{I}\left(\hat{F}(n)\right)\sin(n\varphi) \\
&= \hat{F}(0) + 2\sum_{n=1} \mathfrak{R}\left(\hat{F}(n)\right)\mathfrak{R}\left(\mathrm{e}^{-\mathrm{i}n\varphi}\right) \;+\; \mathfrak{I}\left(\hat{F}(n)\right)\mathfrak{I}\left(\mathrm{e}^{-\mathrm{i}n\varphi}\right)
\end{aligned} \tag{C.76}$$

For a two dimensional Fourier transform defined in eqn. (C.77) half of the modes can be neglected by the complex conjugates given in eqn. (C.78).

$$\begin{aligned}
\hat{F}(m,n) =& \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{2\pi} \mathrm{e}^{-\mathrm{i}(m\theta+n\varphi)} f(\theta,\varphi)\,\mathrm{d}\varphi \\
&\text{for all } m = -N_\theta,\ldots,0,\ldots N_\theta \text{ and } n = -N_\varphi,\ldots,0,\ldots N_\varphi
\end{aligned} \tag{C.77}$$

$$\overline{\hat{F}(-m,-n)} = \hat{F}(m,n), \quad \overline{\hat{F}(-m,n)} = \hat{F}(m,-n), \quad \overline{\hat{F}(0,-n)} = \hat{F}(0,n) \tag{C.78}$$

A typical mistake is to neglect all modes where m or n are negative, thus one should be careful at this point. Various representations of the discrete back-transform are given in eqn. (C.79) and can be combined with eqn. (C.78) in order to gain an increase in efficiency.

$$\begin{aligned}
\hat{f}(\theta,\varphi) &= \sum_{m=-N_\theta}^{N_\theta}\sum_{n=-N_\varphi}^{N_\varphi} \hat{F}(m,n)\mathrm{e}^{\mathrm{i}(m\theta+n\varphi)} \\
&= \sum_{n=-N_\varphi}^{N_\varphi} \hat{F}(0,n)\mathrm{e}^{\mathrm{i}n\varphi} \;+\; \sum_{m=1}^{N_\theta}\left[\sum_{n=-N_\varphi}^{N_\varphi} \hat{F}(m,n)\mathrm{e}^{\mathrm{i}(m\theta+n\varphi)} + \underbrace{\hat{F}(-m,-n)}_{=\overline{\hat{F}(m,n)}}\mathrm{e}^{-\mathrm{i}(m\theta+n\varphi)}\right] \\
&= \hat{F}(0,0) + \sum_{n=1}^{N_\varphi}\left[\hat{F}(0,n)\mathrm{e}^{\mathrm{i}n\varphi} + \hat{F}(0,-n)\mathrm{e}^{\mathrm{i}n\varphi}\right] \\
&\qquad + \sum_{m=1}^{N_\theta}\left[\sum_{n=-N_\varphi}^{N_\varphi} \hat{F}(m,n)\mathrm{e}^{\mathrm{i}(m\theta+n\varphi)} + \underbrace{\hat{F}(-m,-n)}_{=\overline{\hat{F}(m,n)}}\mathrm{e}^{-\mathrm{i}(m\theta+n\varphi)}\right]
\end{aligned} \tag{C.79}$$

C.3. Particle-in-Fourier for Vlasov–Maxwell (3d3v)

For the sake of completeness we also discretize the six dimensional Vlasov–Maxwell equations, such that the provided six dimensional codes can be better understood. For the initialization of the Vlasov–Maxwell solver at $t = 0$ we need to find the electric field by solving the Poisson equation

$$-\Delta\Phi(x,t) = \sum_s \rho_s(x,t) \text{ and } E(x,t) = -\nabla\Phi(x,t). \tag{C.80}$$

Transforming this into spatial Fourier space yields an equation for every mode $k \neq (0,0,0)$ and $j \in \{1,2,3\}$.

$$-(k_1^2+k_2^2+k_3^2)\tilde{\Phi}(k,t) = \tilde{\rho}(x,t) \text{ and } \tilde{E}_j(x,t) = \mathrm{i}k_x\tilde{\Phi}(k,t) \tag{C.81}$$

The Fourier modes of the electric field are then uniquely defined as

$$\tilde{E}_j(k,t) = \frac{-\mathrm{i}k_j}{k_1^2+k_2^2+k_3^2}\tilde{\rho}(k,t), \quad \forall j \in \{1,2,3\} \text{ and } k \neq (0,0,0). \tag{C.82}$$

At any time the Poisson error, the conservation of Gauss' electrostatic and magnetic law, can be checked by verifying

$$\begin{aligned} \mathrm{div}(E(x,t)) &= \nabla\cdot E(x,t) = \partial_{x_1}E_1(x,t) + \partial_{x_2}E_2(x,t) + \partial_{x_3}E_3(x,t) = \sum_s \rho_s(x,t), \\ \mathrm{div}(B(x,t)) &= \nabla\cdot B(x,t) = \partial_{x_1}B_1(x,t) + \partial_{x_2}B_2(x,t) + \partial_{x_3}B_3(x,t) = 0, \end{aligned} \tag{C.83}$$

which in Fourier space reads

$$\begin{aligned} \mathrm{i}k_1\tilde{E}_1(k,t) + \mathrm{i}k_2\tilde{E}_2(k,t) + \mathrm{i}k_3\tilde{E}_3(k,t) &= \sum \tilde{\rho}_s(k,t) \\ \mathrm{i}k_1\tilde{B}_1(k,t) + \mathrm{i}k_2\tilde{B}_2(k,t) + \mathrm{i}k_3\tilde{B}_3(k,t) &= 0. \end{aligned} \tag{C.84}$$

In the following the different equations for each split part of the Hamiltonian $\mathcal{H} = \mathcal{H}_E + \mathcal{H}_B + \mathcal{H}_p$ with their respective Particle-In-Fourier discretization are provided.

- Electric Energy

$$\begin{aligned} \mathcal{H}_E &= \frac{1}{2}\int |E(x,t)|^2 \,\mathrm{d}x \approx \hat{\mathcal{H}}_E = \frac{1}{2}\int_0^{L_3}\int_0^{L_2}\int_0^{L_1} |E(x,t)|^2 \mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3 \\ &= \frac{1}{2}\sum_k \left(|\tilde{E}_1(k,t)|^2 + |\tilde{E}_2(k,t)|^2 + |\tilde{E}_3(k,t)|^2\right) L_1L_2L_3 \end{aligned} \tag{C.85}$$

$$\begin{aligned} \partial_t f + \frac{q}{m}E(x,t)\cdot\partial_v f &= 0 \\ \partial_t B(x,t) = -\nabla\times E(x,t) &= -\begin{pmatrix} \partial_{x_2}E_3(x,t) - \partial_{x_3}E_2(x,t) \\ \partial_{x_3}E_1(x,t) - \partial_{x_1}E_3(x,t) \\ \partial_{x_1}E_2(x,t) - \partial_{x_2}E_1(x,t) \end{pmatrix} \end{aligned} \tag{C.86}$$

$$\partial_t\tilde{B}(k,t) = -\mathrm{i}\begin{pmatrix} k_2\tilde{E}_3(k,t) - k_3\tilde{E}_2(k,t) \\ k_3\tilde{E}_1(k,t) - k_1\tilde{E}_3(k,t) \\ k_1\tilde{E}_2(k,t) - k_2\tilde{E}_1(k,t) \end{pmatrix} \tag{C.87}$$

- Magnetic energy

$$\mathcal{H}_B = \frac{1}{2}\int |B(x,t)|^2 \,\mathrm{d}x \approx \hat{\mathcal{H}}_B = \frac{1}{2}\sum_k \left(|\tilde{B}_1(k,t)|^2 + |\tilde{B}_2(k,t)|^2 + |\tilde{B}_3(k,t)|^2\right) L_1L_2L_3 \tag{C.88}$$

$$\partial_t E(x,t) = c^2 \nabla \times B(x,t) = c^2 \begin{pmatrix} \partial_{x_2} B_3(x,t) - \partial_{x_3} B_2(x,t) \\ \partial_{x_3} B_1(x,t) - \partial_{x_1} B_3(x,t) \\ \partial_{x_1} B_2(x,t) - \partial_{x_2} B_1(x,t) \end{pmatrix} \tag{C.89}$$

$$\partial_t \tilde{E}(k,t) = c^2 \mathrm{i} \begin{pmatrix} k_2 \tilde{B}_3(k,t) - k_3 \tilde{B}_2(k,t) \\ k_3 \tilde{B}_1(k,t) - k_1 \tilde{B}_3(k,t) \\ k_1 \tilde{B}_2(k,t) - k_2 \tilde{B}_1(k,t) \end{pmatrix} \tag{C.90}$$

- Kinetic energy (in 3d)

$$\mathcal{H}_p = \frac{1}{2} \int |v|^2 f(x,v,t) \, \mathrm{d}x\mathrm{d}v \approx \hat{\mathcal{H}}_p = \frac{1}{2} \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \left(v_{1,n}^2 + v_{2,n}^2 + v_{3,n}^2 \right) \tag{C.91}$$

$$\begin{aligned} \partial_t f(x,v,t) + v \cdot \nabla_x f(x,v,t) + \frac{q}{m} (v \times B(x,t)) \cdot \nabla_v f(x,v,t) = 0 \\ \partial_t B(x,t) = 0 \\ \partial_t E(x,t) = - \sum_s q_s \int v f_s(x,v,t) \mathrm{d}v \end{aligned} \tag{C.92}$$

$$\begin{aligned} \dot{x}_1(t) &= v_1(t) \\ \dot{x}_2(t) &= v_2(t) \\ \dot{x}_3(t) &= v_3(t) \\ \dot{v}_1(t) &= \frac{q}{m} \left[v_2(t) B_3(x(t)) - v_3(t) B_2(x(t)) \right] \\ \dot{v}_2(t) &= \frac{q}{m} \left[v_3(t) B_1(x(t)) - v_1(t) B_3(x(t)) \right] \\ \dot{v}_3(t) &= \frac{q}{m} \left[v_1(t) B_2(x(t)) - v_2(t) B_1(x(t)) \right] \end{aligned} \tag{C.93}$$

The Hamiltonian $\mathcal{H}_p = \mathcal{H}_{p_1} + \mathcal{H}_{p_2} + \mathcal{H}_{p_3}$ can be furthermore split into three parts along the spatial components.

- Kinetic energy ($d = 1$), $\hat{H}_{p_1}$

$$\begin{aligned} \dot{v}_1(t) &= 0 \\ \dot{x}_1(t) &= v_1(t) = v_1(0) \\ \dot{v}_2(t) &= -\frac{q}{m} v_1(t) B_3(x(t) = -\frac{q}{m} v_1(0) B_3(x(t)) \\ \dot{v}_3(t) &= \frac{q}{m} v_1(t) B_2(x(t)) = \frac{q}{m} v_1(0) B_2(x(t)) \\ \partial_t E_1(t) &= -q \int v_1 f(x,v,t) \mathrm{d}v \end{aligned} \tag{C.94}$$

Since v_1 is constant we can integrate exactly over $\partial_t x_1(t)$.

$$x_1(\tau) := x_1(0) + \tau v_1(0) \tag{C.95}$$

$$
\begin{aligned}
v_2(t) &= v_2(0) + \left(-\frac{q}{m}\right) \int_0^t v_1(0) \sum_k \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}(k_1 x_1(\tau) + k_2 x_2(0) + k_3 x_3(0))} \mathrm{d}\tau \\
&= v_2(0) + \left(-\frac{q}{m}\right) \sum_k \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}(k_2 x_2(0) + k_3 x_3(0))} \int_0^t v_1(0) \mathrm{e}^{\mathrm{i}k_1 x_1(\tau)} \mathrm{d}\tau \\
&= v_2(0) + \left(-\frac{q}{m}\right) \sum_k \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}(k_2 x_2(0) + k_3 x_3(0))} \int_0^t v_1(0) \mathrm{e}^{\mathrm{i}k_1 (x_1(0) + \tau v_1(0))} \mathrm{d}\tau \\
&= v_2(0) + \left(-\frac{q}{m}\right) \sum_k \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}(k_1 x_1(0) + k_2 x_2(0) + k_3 x_3(0))} \int_0^t v_1(0) \mathrm{e}^{\mathrm{i}k_1 \tau v_1(0)} \mathrm{d}\tau \\
&= v_2(0) + \left(-\frac{q}{m}\right) \sum_{k,\ k_1 \neq 0} \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} \frac{1}{\mathrm{i}k_1} \left[\mathrm{e}^{\mathrm{i}k_1 t v_1(0)} - 1\right] \\
&\qquad + \left(-\frac{q}{m}\right) \sum_{\substack{k \\ k_1 = 0}} \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_1(0) \\
&= v_2(0) + \left(-\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_1 \neq 0}} \tilde{B}_3(k,0) \frac{1}{\mathrm{i}k_1} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)}\right] + \sum_{\substack{k \\ k_1 = 0}} \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_1(0) \right\}
\end{aligned}
\tag{C.96}
$$

The same procedure is applied to the electric field. Here we already see that we only need to calculate the additional $\mathrm{e}^{\mathrm{i}k_1 v_1(0)}$, which can result in a speed-up since there are multiple duplicates in k_1.

$$
\begin{aligned}
x_{1,n}(t) &= x_{1,n}(0) + t v_{1,n}(0) \\
v_{2,n}(t) &= v_{2,n}(0) \\
&\qquad + \left(-\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_1 \neq 0}} \tilde{B}_3(k,0) \frac{1}{\mathrm{i}k_1} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)}\right] + \sum_{\substack{k \\ k_1 = 0}} \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_1(0) \right\} \\
v_{3,n}(t) &= v_{3,n}(0) \\
&\qquad + \left(\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_1 \neq 0}} \tilde{B}_2(k,0) \frac{1}{\mathrm{i}k_1} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)}\right] + \sum_{\substack{k \\ k_1 = 0}} \tilde{B}_2(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_1(0) \right\} \\
\tilde{E}_1(k,t) &= \tilde{E}_1(k,0) + q \frac{1}{L} \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \begin{cases} \frac{1}{\mathrm{i}k_1} \left[\mathrm{e}^{-\mathrm{i}k\cdot x_n(t)} - \mathrm{e}^{-\mathrm{i}k\cdot x_n(0)}\right] & \text{for } k_1 \neq 0 \\ -t\ v_{1,n}(0) & \text{for } k_1 = 0 \end{cases}
\end{aligned}
\tag{C.97}
$$

- Kinetic energy ($d = 2$), $\hat{H}_{p_2}$

$$
\begin{aligned}
\dot{x}_2(t) &= v_2(0) \\
\dot{v}_1(t) &= \frac{q}{m} v_2(0) B_3(x(t)) \\
\dot{v}_3(t) &= -\frac{q}{m} v_2(0) B_1(x(t)) \\
\partial_t E_2(t) &= -q \int v_2 f(x,v,t) \mathrm{d}v
\end{aligned}
\tag{C.98}
$$

$$
\begin{aligned}
x_{2,n}(t) &= x_{2,n}(0) + t v_{2,n}(0) \\
v_{1,n}(t) &= v_{1,n}(0) \\
&\quad + \left(\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_2 \neq 0}} \tilde{B}_3(k,0) \frac{1}{\mathrm{i}k_2} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)} \right] + \sum_{\substack{k \\ k_2 = 0}} \tilde{B}_3(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_2(0) \right\} \\
v_{3,n}(t) &= v_{3,n}(0) \\
&\quad + \left(-\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_2 \neq 0}} \tilde{B}_1(k,0) \frac{1}{\mathrm{i}k_2} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)} \right] + \sum_{\substack{k \\ k_2 = 0}} \tilde{B}_1(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_2(0) \right\} \\
\tilde{E}_2(k,t) &= \tilde{E}_2(k,0) + q \frac{1}{L} \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \begin{cases} \frac{1}{\mathrm{i}k_2} \left[\mathrm{e}^{-\mathrm{i}k.x_n(t)} - \mathrm{e}^{-\mathrm{i}k\cdot x_n(0)} \right] & \text{for } k_2 \neq 0 \\ -t \; v_{2,n}(0) & \text{for } k_2 = 0 \end{cases}
\end{aligned}
\tag{C.99}
$$

- Kinetic energy ($d = 3$), $\hat{H}_{p_3}$

$$
\begin{aligned}
\dot{x}_3(t) &= v_3(t) \\
\dot{v}_1(t) &= -\frac{q}{m} v_3(t) B_2(x(t)) \\
\dot{v}_2(t) &= \frac{q}{m} v_3(0) B_1(x(t)) \\
\partial_t E_3(t) &= -q \int v_3 f(x,v,t) \mathrm{d}v
\end{aligned}
\tag{C.100}
$$

$$
\begin{aligned}
x_{3,n}(t) &= x_{3,n}(0) + t v_{3,n}(0) \\
v_{1,n}(t) &= v_{1,n}(0) \\
&\quad + \left(-\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_3 \neq 0}} \tilde{B}_2(k,0) \frac{1}{\mathrm{i}k_3} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)} \right] + \sum_{\substack{k \\ k_3 = 0}} \tilde{B}_2(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_3(0) \right\} \\
v_{2,n}(t) &= v_{2,n}(0) \\
&\quad + \left(\frac{q}{m}\right) \left\{ \sum_{\substack{k \\ k_3 \neq 0}} \tilde{B}_1(k,0) \frac{1}{\mathrm{i}k_3} \left[\mathrm{e}^{\mathrm{i}k\cdot x(t)} - \mathrm{e}^{\mathrm{i}k\cdot x(0)} \right] + \sum_{\substack{k \\ k_3 = 0}} \tilde{B}_1(k,0) \mathrm{e}^{\mathrm{i}k\cdot x(0)} t v_3(0) \right\} \\
\tilde{E}_3(k,t) &= \tilde{E}_3(k,0) + q \frac{1}{L} \frac{1}{N_p} \sum_{n=1}^{N_p} w_n \begin{cases} \frac{1}{\mathrm{i}k_3} \left[\mathrm{e}^{-\mathrm{i}k.x_n(t)} - \mathrm{e}^{-\mathrm{i}k\cdot x_n(0)} \right] & \text{for } k_3 \neq 0 \\ -t \; v_{3,n}(0) & \text{for } k_3 = 0 \end{cases}
\end{aligned}
\tag{C.101}
$$

The total momentum is conserved over time, such that the discrete total momentum has to be considered which reads

$$\begin{aligned}\mathcal{P} &= m \iint v f(x,v,t)\ \mathrm{d}x\mathrm{d}v + \int E(x,t) \times B(x,t)\ \mathrm{d}x \\ &= m \iint v f(x,v,t)\ \mathrm{d}x\mathrm{d}v + \begin{pmatrix} \int E_2(x,t)B_3(x,t) - E_3(x,t)B_2(x,t)\ \mathrm{d}x \\ \int E_3(x,t)B_1(x,t) - E_1(x,t)B_3(x,t)\ \mathrm{d}x \\ \int E_1(x,t)B_2(x,t) - E_2(x,t)B_1(x,t)\ \mathrm{d}x \end{pmatrix} \\ \hat{\mathcal{P}} &= \frac{m}{N_p} \sum_{n=1}^{N_p} w_n v_n + L \sum_k \begin{pmatrix} \tilde{E}_2(k,t)\tilde{B}_3(k,t) - \tilde{E}_3(k,t)\tilde{B}_2(k,t) \\ \tilde{E}_3(k,t)\tilde{B}_1(k,t) - \tilde{E}_1(k,t)\tilde{B}_3(k,t) \\ \tilde{E}_1(k,t)\tilde{B}_2(k,t) - \tilde{E}_2(k,t)\tilde{B}_1(k,t) \end{pmatrix}.\end{aligned} \tag{C.102}$$

Appendix D.

PIF and Semi-Lagrange Vlasov–Poisson in 6d

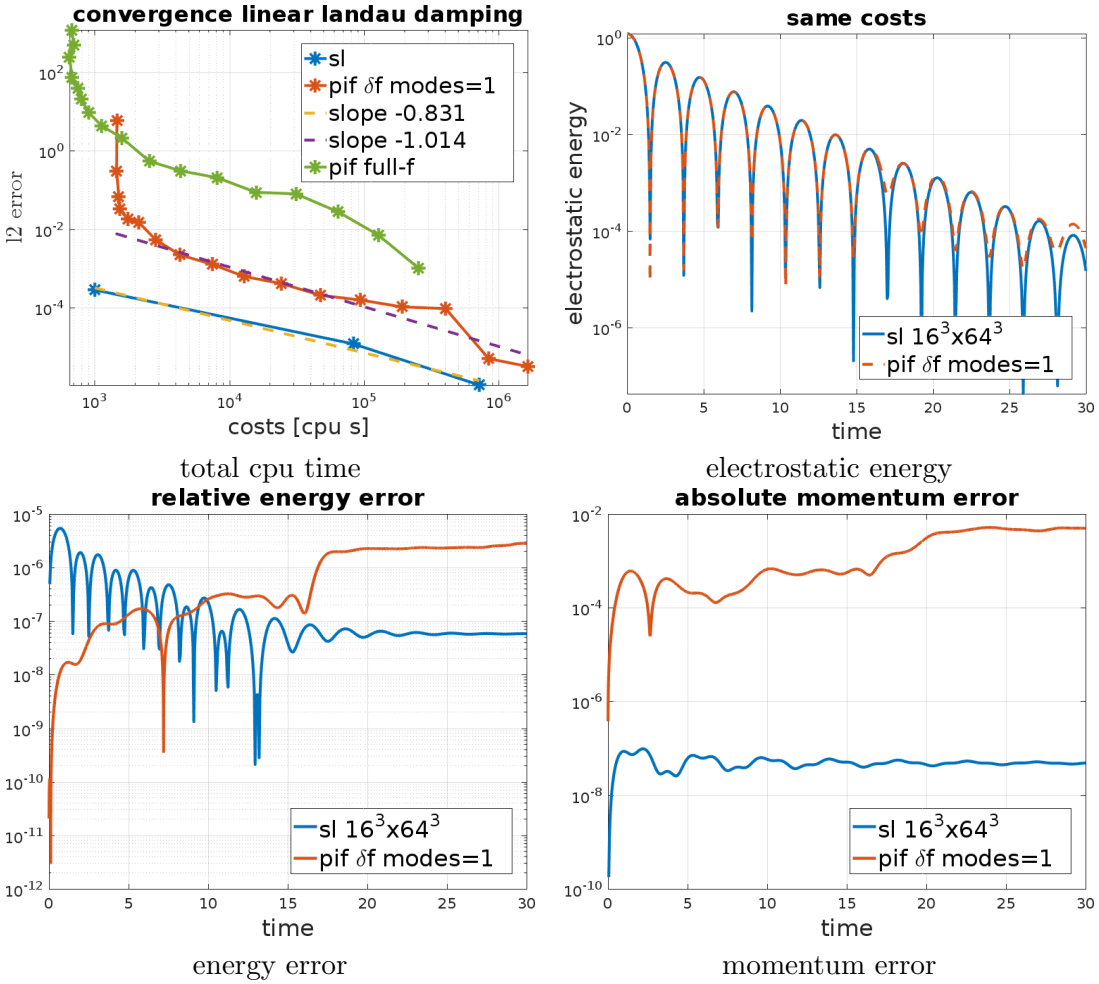
Access to Helios[1] in the scope of the Selavlas project, enabled us to benchmark a single species Vlasov–Poisson PIF scheme on a larger scale. For this a standard PIF for arbitrary integer dimensions was implemented in SeLaLib. Time integration is done by symplectic Runge-Kutta schemes up to fourth order. In presence of a constant external magnetic field the phase space conserving Boris scheme is implemented. The MPI parallelization is done by domain cloning, which means that every node holds all Fourier modes. Since the Poisson solve is trivial and only few modes are used this approach is computationally feasible. Solving the Poisson equation in Fourier space is simple, yet the charge projection onto the spectral grid is expensive, since every particle contributes to every Fourier mode. Each Fourier mode is calculated by evaluation of a complex exponential such that there is no roundoff. In order to achieve a $\mathcal{O}(N)$ convergence, the quasi monte carlo Sobol sequence is used for the random samples. We present simulations of Landau damping and a Bump-on-Tail instability and compare the results as well as the computational performance to a grid based Semi-Lagrangian Vlasov–Poisson solver. This results are a joint work with K. Kormann and were presented at the PASC16 conference [204]. The Semi-Lagrangian solver also developed in SeLaLib was using the full grid in order to compare the 6D performance [259, 260]. Other implemented variants using e.g. the tensor train format [261, 262] take advantage of our simply constructed test-cases such that a comparison is pointless.
We consider in d dimensions the wave vector $k, k^0 \in \mathbb{R}^n$. The length of the d-dimensional periodic box $[0, L_1] \times \cdots \times [0, L_d]$ is given as $L_n = \frac{2\pi}{k_n^0}, \quad \forall n = 1, \ldots, d$. The initial conditions for Landau damping (eqn. (D.1)) and the bump-on-tail instability(eqn. (D.2)) are extended from one to d dimensions by tensor product.

$$f(x, v, t = 0) = \left(1 + \epsilon \sum_{i=1}^{d} \cos(k_i x_i)\right) \frac{1}{(\sqrt{2\pi})^d} e^{-\frac{1}{2}\sum_{i=1}^{d} v_i^2} \tag{D.1}$$

$$f(x, v, t = 0) = \left(1 + \epsilon \sum_{i=1}^{d} \cos\left(k_i x_i\right)\right) \frac{1}{\sqrt{2\pi}} \left((1 - n_b)\mathrm{e}^{-\frac{|v|^2}{2}} + \frac{n_b}{\sigma}\mathrm{e}^{-\frac{|v-v_0|^2}{2\sigma^2}}\right) \tag{D.2}$$

By this Ansatz reference solutions can be synthesized from a one dimensional spectral solver. In the following the L^2 error on the electrostatic energy is compared. Since for a damped mode one would only compare the initial condition therefore, the first half of the simulation is neglected. The QMC convergence is achieved for Landau damping, see figs.D.1,D.2 and the PIF scales perfectly due to the small amount of modes. But the Semi-Lagrangian solver cannot be beaten with this naive Fortran implementation, especially as the costs increase rapidly with more Fourier modes. The same problem appears for the Bump-on-tail instability, where the PIF performs a little bit better, see fig.D.3. Nevertheless, we have to note that

[1]This work was carried out using the HELIOS supercomputer system at Computational Simulation Centre of International Fusion Energy Research Centre (IFERC-CSC), Aomori, Japan, under the Broader Approach collaboration between Euratom and Japan, implemented by Fusion for Energy and QST.

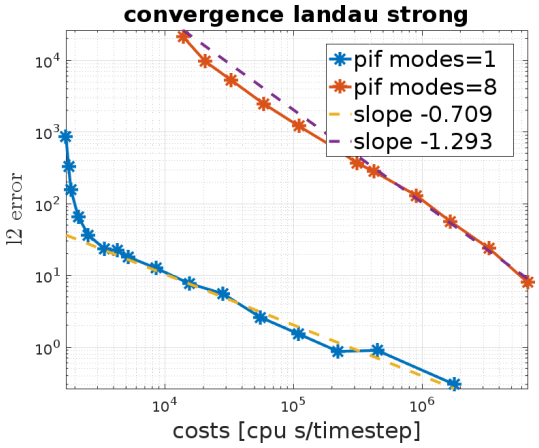


total cpu time | electrostatic energy

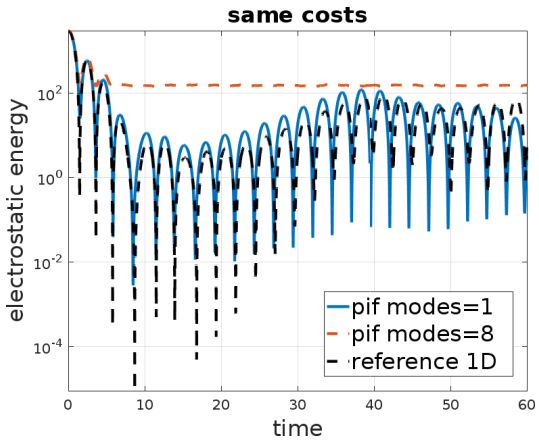
energy error | momentum error

Figure D.1.: Linear Landau damping $\epsilon = 0.01,\ \Delta t = 0.1,\ k = [0.5, 0.5, 0.5]$.

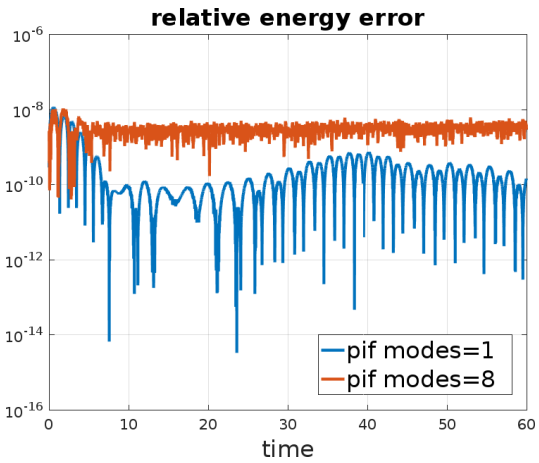
such comparison can be influenced already by minor optimizations such that this is a constant race for performance.



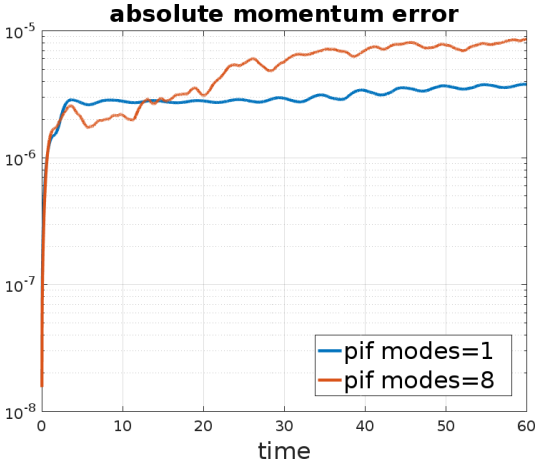
total cpu time



electrostatic energy



energy error



momentum error

Figure D.2.: Strong Landau damping $\epsilon = 0.5,\ \Delta t = 0.01,\ k = [0.5, 0.5, 0.5]$.

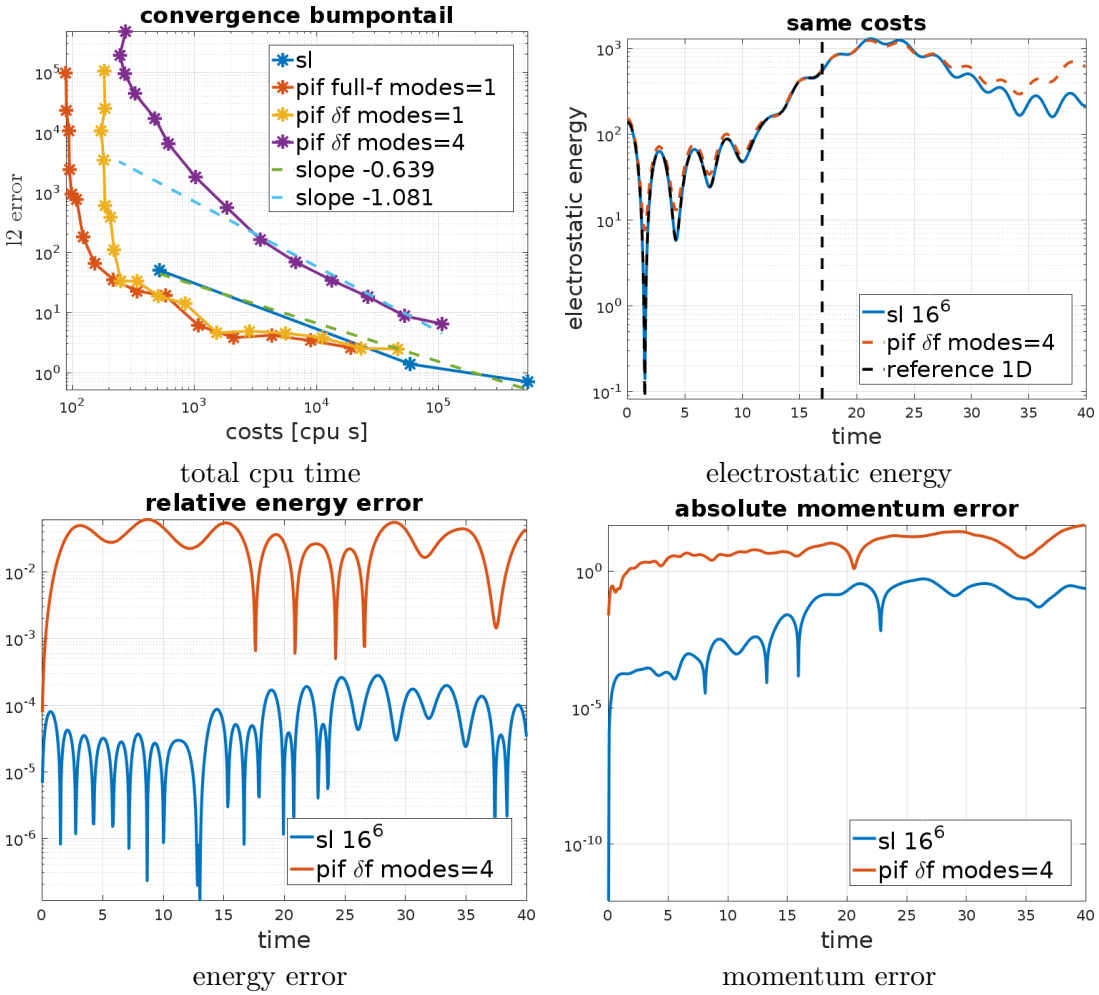


Figure D.3.: Bump-on-tail instability $\epsilon = 0.5,\ \Delta t = 0.01,\ k = [0.3, 0.3, 0.3],\ n_b = 0.1,\ v_0 = 4.5$

Acknowledgments

I would like to thank you, Eric Sonnendrücker, for this enormous opportunity, the trust and the freedom you offered me during this time. I highly appreciated your steadily open door resulting in joyful and encouraging discussions about mathematics and physics. I learned a lot. Thank You very much!

I would like to thank Katharina Kormann, Michael Kraus and Andreas Denner for the extensive discussions, the fruitful cooperation and the valuable insights.

At last I would like to thank Cecilia for steadily sparking my enthusiasm for physics, the support and all the love.

Bibliography

[1] Clément Mouhot and Cédric Villani. "On landau damping". In: *Acta mathematica* 207.1 (2011), pp. 29–201.

[2] Cédric Villani. "Particle systems and nonlinear Landau damping a". In: *Physics of Plasmas* 21.3 (2014), p. 030901.

[3] Magdi Shoucri and Georg Knorr. "Numerical integration of the Vlasov equation". In: *Journal of Computational Physics* 14.1 (1974), pp. 84–92.

[4] Francis Filbet and Eric Sonnendrücker. "Comparison of eulerian vlasov solvers". In: *Computer Physics Communications* 150.3 (2003), pp. 247–266.

[5] Charles K Birdsall and A Bruce Langdon. *Plasma physics via computer simulation*. CRC Press, 2004.

[6] Roger W Hockney and James W Eastwood. *Computer simulation using particles*. crc Press, 1988.

[7] Eric Sonnendrücker et al. "The semi-Lagrangian method for the numerical resolution of the Vlasov equation". In: *Journal of computational physics* 149.2 (1999), pp. 201–220.

[8] H.Ralph Lewis. "Energy-conserving numerical approximations for Vlasov plasmas". In: *Journal of Computational Physics* 6.1 (1970), pp. 136 –141. ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/0021-9991(70)90012-4. URL: http://www.sciencedirect.com/science/article/pii/0021999170900124.

[9] Yu. N. Grigoryev, V.A. Vshivkov, and M.P. Fedoruk. *Numerical "Particle-In-Cell" Methods: Theory and Applications*. Walter de Gruyter, 2002.

[10] Evstati G Evstatiev and Bradley A Shadwick. "Variational formulation of particle algorithms for kinetic plasma simulations". In: *Journal of Computational Physics* 245 (2013), pp. 376–398.

[11] A Bottino and E Sonnendrücker. "Monte Carlo particle-in-cell methods for the simulation of the Vlasov–Maxwell gyrokinetic equations". In: *Journal of Plasma Physics* 81.05 (2015), p. 435810501.

[12] Eric Sonnendrücker and K Kormann. "Monte Carlo Methods with applications to plasma physics". In: *Vorlesung (SS 2014)*. 2014.

[13] Ahmet Y Aydemir. "A unified Monte Carlo interpretation of particle simulations and applications to non-neutral plasmas". In: *Physics of Plasmas (1994-present)* 1.4 (1994), pp. 822–831.

[14] Michael Kraus. "Variational integrators in plasma physics". In: *arXiv preprint arXiv:1307.5665* (2013).

[15] Eric Sonnendrücker et al. "A split control variate scheme for {PIC} simulations with collisions". In: *Journal of Computational Physics* 295 (2015), pp. 402 –419. ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/j.jcp.2015.04.004. URL: http://www.sciencedirect.com/science/article/pii/S0021999115002442.

[16] Michael B Giles. "Multilevel Monte Carlo methods". In: *Acta Numerica* 24 (2015), p. 259.

[17] LF Ricketson. "A multilevel Monte Carlo method for a class of McKean-Vlasov processes". In: *arXiv preprint arXiv:1508.02299* (2015).

[18] Lee F Ricketson and Antoine J Cerfon. "Sparse grid techniques for particle-in-cell schemes". In: *Plasma Physics and Controlled Fusion* 59.2 (2016), p. 024002.

[19] Michael Kraus et al. "GEMPIC: Geometric electromagnetic particle-in-cell methods". In: *Journal of Plasma Physics* 83.4 (2017).

[20] Glenn Joyce, Georg Knorr, and Homer K Meier. "Numerical integration methods of the Vlasov equation". In: *Journal of Computational Physics* 8.1 (1971), pp. 53–63.

[21] T-H Watanabe and Hideo Sugama. "Vlasov and drift kinetic simulation methods based on the symplectic integrator". In: *Transport Theory and Statistical Physics* 34.3-5 (2005), pp. 287–309.

[22] Nicolas Crouseilles, Lukas Einkemmer, and Erwan Faou. "Hamiltonian splitting for the Vlasov–Maxwell equations". In: *Journal of Computational Physics* 283 (2015), pp. 224–240.

[23] Francis Filbet and Luis Rodrigues. "Asymptotically preserving particle-in-cell methods for inhomogenous strongly magnetized plasmas". In: *arXiv preprint arXiv:1701.06868* (2017).

[24] Francis Filbet and Luis Miguel Rodrigues. "Asymptotically Stable Particle-In-Cell Methods for the Vlasov–Poisson System with a Strong External Magnetic Field". In: *SIAM Journal on Numerical Analysis* 54.2 (2016), pp. 1120–1146.

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

[26] Natalia Tronko, Alberto Bottino, and Eric Sonnendrücker. "Second order gyrokinetic theory for particle-in-cell codes". In: *Physics of Plasmas* 23.8 (2016), p. 082505.

[27] Thomas Müller-Gronbach, Erich Novak, and Klaus Ritter. *Monte Carlo-Algorithmen*. Springer-Verlag, 2012.

[28] Christiane Lemieux. *Monte carlo and quasi-monte carlo sampling*. New York: Springer, 2009. ISBN: 978-0-387-78165-5.

[29] Ernst Hairer. *Geometric Numerical Integration. Lecture notes.* 2010.

[30] Robert I McLachlan and G Reinout W Quispel. "Geometric integrators for ODEs". In: *Journal of Physics A: Mathematical and General* 39.19 (2006), p. 5251.

[31] Hong Qin et al. "Why is Boris algorithm so good?" In: *Physics of Plasmas* 20.8, 084503 (2013), pp. –. DOI: 10.1063/1.4818428. URL: http://scitation.aip.org/content/aip/journal/pop/20/8/10.1063/1.4818428.

[32] Etienne Forest and Ronald D Ruth. "Fourth order symplectic integration". In: *Physica* 43.LBL-27662 (1989), pp. 105–117.

[33] Bernt Øksendal. "Stochastic differential equations". In: *Stochastic differential equations*. Springer, 2003, pp. 65–84.

[34] J. C. Butcher. *Numerical methods for ordinary differential equations*. Chichester, West Sussex, England Hoboken: J. Wiley, 2003. ISBN: 0-471-96758-0.

[35] E Hairer. *Solving ordinary differential equations*. Berlin New York: Springer-Verlag, 1993. ISBN: 9783540566700.

[36] Etienne Forest and Ronald D. Ruth. "Fourth-order symplectic integration". In: *Physica D: Nonlinear Phenomena* 43.1 (May May 1990), pp. 105–117.

[37] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Vol. 31. Springer Science & Business Media, 2006.

[38] Jan Beirlant et al. "Nonparametric entropy estimation: An overview". In: *International Journal of Mathematical and Statistical Sciences* 6.1 (1997), pp. 17–39.

[39] Nader Ebrahimi, Kurt Pflughoeft, and Ehsan S Soofi. "Two measures of sample entropy". In: *Statistics & Probability Letters* 20.3 (1994), pp. 225–234.

[40] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. "Estimating mutual information". In: *Physical review E* 69.6 (2004), p. 066138.

[41] Harshinder Singh et al. "Nearest neighbor estimates of entropy". In: *American journal of mathematical and management sciences* 23.3-4 (2003), pp. 301–321.

[42] WW Lee and WM Tang. "Gyrokinetic particle simulation of ion temperature gradient drift instabilities". In: *Physics of Fluids (1958-1988)* 31.3 (1988), pp. 612–624.

[43] John A Krommes and Genze Hu. "The role of dissipation in the theory and simulations of homogeneous plasma turbulence, and resolution of the entropy paradox". In: *Physics of Plasmas (1994-present)* 1.10 (1994), pp. 3211–3238.

[44] Alberto Bottino. "Entropy evolution and dissipation in collisionless particle-in-cell gyrokinetic simulations". In: *Numerical Methods for the Kinetic Equations of Plasma Physics (NumKin 2013)*. 2013.

[45] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[46] SE Parker and WW Lee. "A fully nonlinear characteristic method for gyrokinetic simulation". In: *Physics of Fluids B: Plasma Physics (1989-1993)* 5.1 (1993), pp. 77–86.

[47] Srinath Vadlamani. "An algorithmic unification of particle-in-cell and continuum methods and a wave-particle description for the electron temperature gradient (etg) instability saturation". In: (2005).

[48] Aurore Back et al. "An axisymmetric PIC code based on isogeometric analysis". In: *Esaim: proceedings*. Vol. 32. EDP Sciences. 2011, pp. 118–133.

[49] C. de Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York, 2001. ISBN: 9780387953663. URL: `http://books.google.de/books?id=m0QDJvBI\_ecC`.

[50] Klaus Höllig. *Finite Element Methods with B-Splines*. Society for Industrial and Applied Mathematics, 2003. DOI: `10.1137/1.9780898717532`. eprint: `http://epubs.siam.org/doi/pdf/10.1137/1.9780898717532`. URL: `http://epubs.siam.org/doi/abs/10.1137/1.9780898717532`.

[51] Bernard. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1986.

[52] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[53] Berwin A Turlach et al. *Bandwidth selection in kernel density estimation: A review*. Université catholique de Louvain Louvain-la-Neuve, 1993.

[54] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[55] C-K Huang et al. "Finite grid instability and spectral fidelity of the electrostatic Particle-In-Cell algorithm". In: *arXiv preprint arXiv:1508.03360* (2015).

[56] Rong Liu and Lijian Yang. "Kernel estimation of multivariate cumulative distribution function". In: *Journal of Nonparametric Statistics* 20.8 (2008), pp. 661–677.

[57] Bradley Efron. *The jackknife, the bootstrap and other resampling plans*. SIAM, 1982.

[58] A.C. Rencher. *Multivariate statistical inference and applications*. Wiley series in probability and statistics: Texts and references section. Wiley, 1998. ISBN: 9780471571513. URL: http://books.google.de/books?id=C7fvAAAAMAAJ.

[59] A.M. Mathai and Serge B. Provost. *Quadratic Forms in Random Variables*. Vol. 126. Statistics Series. MARCEL DEKKER, INC., 1992. ISBN: 0-8247-8691-2.

[60] Alain-Sol Sznitman. "Topics in propagation of chaos". In: *Ecole d'Eté de Probabilités de Saint-Flour XIX — 1989*. Ed. by Paul-Louis Hennequin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 165–251. ISBN: 978-3-540-46319-1. DOI: 10.1007/BFb0085169. URL: http://dx.doi.org/10.1007/BFb0085169.

[61] Sylvie Méléard. "Asymptotic behaviour of some interacting particle systems; McKean-Vlasov and Boltzmann models". In: *Probabilistic models for nonlinear partial differential equations*. Springer, 1996, pp. 42–95.

[62] Francois Golse. "The mean-field limit for a regularized Vlasov-Maxwell dynamics". In: *Communications in Mathematical Physics* 310.3 (2012), pp. 789–816.

[63] Takashi Nakamura and Takashi Yabe. "Cubic interpolated propagation scheme for solving the hyper-dimensional vlasov—poisson equation in phase space". In: *Computer Physics Communications* 120.2 (1999), pp. 122–154.

[64] Roman Hatzky et al. "Energy conservation in a nonlinear gyrokinetic particle-in-cell code for ion-temperature-gradient-driven modes in θ-pinch geometry". In: *Physics of Plasmas (1994-present)* 9.3 (2002), pp. 898–912. DOI: 10.1063/1.1449889. URL: http://scitation.aip.org/content/aip/journal/pop/9/3/10.1063/1.1449889.

[65] Russel E. Caflisch. "Monte Carlo and quasi-Monte Carlo methods". In: *Acta Numerica* 7 (Jan. 1998), pp. 1–49. ISSN: 1474-0508. DOI: 10.1017/S0962492900002804. URL: http://journals.cambridge.org/article_S0962492900002804.

[66] Art B Owen and Seth D Tribble. "A quasi-monte carlo metropolis algorithm". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.25 (2005), pp. 8844–8849.

[67] J LV Lewandowski. "Numerical loading of a Maxwellian probability distribution function". In: *Canadian journal of physics* 81.8 (2003), pp. 989–996.

[68] R. Kleiber et al. "An improved control-variate scheme for particle-in-cell simulations with collisions". In: *Computer Physics Communications* 182.4 (2011), pp. 1005 –1012. ISSN: 0010-4655. DOI: http://dx.doi.org/10.1016/j.cpc.2010.12.045. URL: http://www.sciencedirect.com/science/article/pii/S0010465510005382.

[69] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013. URL: http://statweb.stanford.edu/~owen/mc/.

[70] Fabio Riva, Carrie F Beadle, and Paolo Ricci. "A methodology for the rigorous verification of Particle-in-Cell simulations". In: *Physics of Plasmas* 24.5 (2017), p. 055703.

[71] R. Hatzky, A. Könies, and A. Mishchenko. "Electromagnetic gyrokinetic {PIC} simulation with an adjustable control variates method". In: *Journal of Computational Physics* 225.1 (2007), pp. 568 –590. ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/j.jcp.2006.12.019. URL: http://www.sciencedirect.com/science/article/pii/S0021999106006085.

[72] Fred J. Hickernell, Christiane Lemieux, and Art B. Owen. "Control Variates for Quasi-Monte Carlo". In: *Statist. Sci.* 20.1 (Feb. 2005), pp. 1–31. DOI: 10.1214/088342304000000468. URL: http://dx.doi.org/10.1214/088342304000000468.

[73] Nicol N Schraudolph, Jin Yu, Simon Günter, et al. "A Stochastic Quasi-Newton Method for Online Convex Optimization." In: *AISTATS*. Vol. 7. 2007, pp. 436–443.

[74] Richard H Byrd et al. "A stochastic quasi-Newton method for large-scale optimization". In: *SIAM Journal on Optimization* 26.2 (2016), pp. 1008–1031.

[75] Juris Vencels et al. "SpectralPlasmaSolver: a spectral code for multiscale simulations of collisionless, magnetized plasmas". In: *Journal of Physics: Conference Series*. Vol. 719. 1. IOP Publishing. 2016, p. 012022.

[76] Stuart C Schwartz. "Estimation of probability density by an orthogonal series". In: *The Annals of Mathematical Statistics* (1967), pp. 1261–1265.

[77] Gilbert G Walter. "Properties of Hermite series estimation of probability density". In: *The Annals of Statistics* (1977), pp. 1258–1264.

[78] René F. Swarttouw (auth.) Roelof Koekoek Peter A. Lesky. *Hypergeometric Orthogonal Polynomials and Their q-Analogues*. 1st ed. Springer Monographs in Mathematics. Springer-Verlag Berlin Heidelberg, 2010. ISBN: 3642050131,9783642050138.

[79] Shane G Henderson and Burt Simon. "Adaptive simulation using perfect control variates". In: *Journal of applied probability* 41.03 (2004), pp. 859–876.

[80] Nomesh Bolia and Sandeep Juneja. "Function-approximation-based perfect control variates for pricing American options". In: *Simulation Conference, 2005 Proceedings of the Winter*. IEEE. 2005, pp. 1876–1883.

[81] Art B Owen. "Monte Carlo variance of scrambled net quadrature". In: *SIAM Journal on Numerical Analysis* 34.5 (1997), pp. 1884–1910.

[82] Pierre L'Ecuyer and Christiane Lemieux. "Recent advances in randomized quasi-Monte Carlo methods". In: *Modeling uncertainty*. Springer, 2005, pp. 419–474.

[83] Roman Hatzky. "Domain cloning for a particle-in-cell (PIC) code on a cluster of symmetric-multiprocessor (SMP) computers". In: *Parallel Computing* 32.4 (2006), pp. 325–330.

[84] Josef Dick. "Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands". In: *The Annals of Statistics* (2011), pp. 1372–1398.

[85] Josef Dick, Frances Y Kuo, and Ian H Sloan. "High-dimensional integration: the quasi-Monte Carlo way". In: *Acta Numerica* 22 (2013), pp. 133–288.

[86] S Ethier, WM Tang, and Z Lin. "Gyrokinetic particle-in-cell simulations of plasma microturbulence on advanced computing platforms". In: *Journal of Physics: Conference Series*. Vol. 16. 1. IOP Publishing. 2005, p. 1.

[87] Nicolas Crouseilles, Michel Mehrenberger, and Hocine Sellama. "Numerical solution of the gyroaverage operator for the finite gyroradius guiding-center model". In: *Communications in Computational Physics* 8.3 (2010), p. 484.

[88] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[89] David Blackwell. "Conditional Expectation and Unbiased Sequential Estimation". In: *Ann. Math. Statist.* 18.1 (Mar. 1947), pp. 105–110. DOI: 10.1214/aoms/1177730497. URL: http://dx.doi.org/10.1214/aoms/1177730497.

[90] Christian Robert and George Casella. *Introducing Monte Carlo Methods with R*. Springer Science & Business Media, 2009.

[91] Roberto Szechtman. "A hilbert space approach to variance reduction". In: *Handbooks in Operations Research and Management Science* 13 (2006), pp. 259–289.

[92] Christophe Steiner et al. "Gyroaverage operator for a polar mesh". In: *The European Physical Journal D* 69.1 (2015), pp. 1–16.

[93] J. A. C. Weideman. "Numerical Integration of Periodic Functions: A Few Examples". In: *The American Mathematical Monthly* 109.1 (2002), pp. 21–36. ISSN: 00029890, 19300972. URL: http://www.jstor.org/stable/2695765.

[94] Alfio Quarteroni. *Numerical Mathematics*. New York, NY: Springer, 2007. ISBN: 978-3-540-49809-4.

[95] Robert P Christian and George Casella. *Monte Carlo statistical methods*. 2007.

[96] Hua Li and Hua Zou. "A random integral quadrature method for numerical analysis of the second kind of Volterra integral equations". In: *Journal of Computational and Applied Mathematics* 237.1 (2013), pp. 35–42.

[97] Qiang Du, Vance Faber, and Max Gunzburger. "Centroidal Voronoi tessellations: Applications and algorithms". In: *SIAM review* 41.4 (1999), pp. 637–676.

[98] Yang Liu et al. "On centroidal Voronoi tessellation - energy smoothness and fast computation". In: *ACM Transactions on Graphics (ToG)* 28.4 (2009), p. 101.

[99] M Vorechovskỳ, V Sadılek, and J Eliáš. "Application of Voronoi Weights in Monte Carlo Integration with a Given Sampling Plan". In: ().

[100] Jingyan Zhang, Maria Emelianenko, and Qiang Du. "Periodic centroidal Voronoi tessellations". In: *International Journal of Numerical Analysis and Modeling* 9.4 (2012), pp. 950–969.

[101] Sylvain Corlay et al. "Functional quantization-based stratified sampling methods". In: *Monte Carlo Methods and Applications* 21.1 (2015), pp. 1–32.

[102] Nikolaos A Gatsonis and Anton Spirkin. "A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay–Voronoi grids". In: *Journal of Computational Physics* 228.10 (2009), pp. 3742–3761.

[103] Phuc T Luu, T Tückmantel, and A Pukhov. "Voronoi particle merging algorithm for PIC codes". In: *Computer Physics Communications* 202 (2016), pp. 165–174.

[104] Peter W Glynn and Roberto Szechtman. "Some new perspectives on the method of control variates". In: *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer, 2002, pp. 27–49.

[105] Nawaf Bou-Rabee and Houman Owhadi. "Long-run accuracy of variational integrators in the stochastic context". In: *SIAM Journal on Numerical Analysis* 48.1 (2010), pp. 278–297.

[106] Nawaf Bou-Rabee and Houman Owhadi. "Stochastic variational integrators". In: *arXiv preprint arXiv:0708.2187* (2007).

[107] Ernst Hairer. "Symmetric projection methods for differential equations on manifolds". In: *BIT Numerical Mathematics* 40.4 (2000), pp. 726–734.

[108] Pierre Degond. "Spectral theory of the linearized Vlasov-Poisson equation". In: *Transactions of the American Mathematical Society* 294.2 (1986), pp. 435–453.

[109] Thomas H Stix. *Waves in plasmas*. Springer Science & Business Media, 1992.

[110] Eric Sonnendrücker and K Kormann. "Numerical methods for Vlasov equations". In: *Lecture notes* (2013).

[111] Pedro Gonnet, Ricardo Pachón, and Lloyd N Trefethen. "Robust rational interpolation and least-squares". In: *Electronic Transactions on Numerical Analysis* 38 (2011), pp. 146–167.

[112] Anthony P Austin, Peter Kravanja, and Lloyd N Trefethen. "Numerical algorithms based on analytic function values at roots of unity". In: *SIAM Journal on Numerical Analysis* 52.4 (2014), pp. 1795–1821.

[113] Bengt Eliasson. "Numerical Simulations of the Fourier-Transformed Vlasov-Maxwell System in Higher Dimensions—Theory and Applications". In: *Transport Theory and Statistical Physics* 39.5-7 (2010), pp. 387–465.

[114] Takashi Minoshima, Yosuke Matsumoto, and Takanobu Amano. "Multi-moment advection scheme in three dimension for Vlasov simulations of magnetized plasma". In: *Journal of Computational Physics* 236 (2013), pp. 81–95.

[115] Yann Kempf et al. "Wave dispersion in the hybrid-Vlasov model: Verification of Vlasiator". In: *Physics of Plasmas* 20.11 (2013), p. 112114.

[116] J W S Cook, R O Dendy, and S C Chapman. "Particle-in-cell simulations of the magnetoacoustic cyclotron instability of fusion-born alpha-particles in tokamak plasmas". In: *Plasma Physics and Controlled Fusion* 55.6 (2013), p. 065003. URL: `http://stacks.iop.org/0741-3335/55/i=6/a=065003`.

[117] Wei Li et al. "Enhanced dispersion analysis of borehole array sonic measurements with amplitude and phase estimation method". In: *SEG Technical Program Expanded Abstracts 2012*. Society of Exploration Geophysicists, 2012, pp. 1–5.

[118] Tomasz Zieliński and Krzysztof Duda. "Frequency and damping estimation methods-an overview". In: *Metrology and Measurement Systems* 18.4 (2011), pp. 505–528.

[119] El-Hadi Djermoune, Magalie Thomassin, and Marc Tomczak. "First-order analysis of the mode and amplitude estimates of a damped sinusoid using matrix pencil". In: *Signal Processing Conference, 2009 17th European*. IEEE. 2009, pp. 1017–1021.

[120] G Jost et al. "Global linear gyrokinetic simulations in quasi-symmetric configurations". In: *Physics of Plasmas* 8.7 (2001), pp. 3321–3333.

[121] AG Peeters et al. "The nonlinear gyro-kinetic flux tube code GKW". In: *Computer Physics Communications* 180.12 (2009), pp. 2650–2672.

[122] Ryusuke Numata et al. "AstroGK: Astrophysical gyrokinetics code". In: *Journal of Computational Physics* 229.24 (2010), pp. 9347–9372.

[123] V Grandgirard et al. "Global full-f gyrokinetic simulations of plasma turbulence". In: *Plasma Physics and Controlled Fusion* 49.12B (2007), B173.

[124] Hannes Risken. *The Fokker-Planck Equation Methods of Solution and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. ISBN: 978-3-642-61544-3.

[125] Thibaut Vernay. "Collisions in global gyrokinetic simulations of tokamak plasmas using the delta-f particle-in-cell approach: Neoclassical physics and turbulent transport". PhD thesis. Ecole Polytechnique Fédérale de Lausanne, 2013.

[126] Arnaud Doucet. *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer New York, 2001. ISBN: 978-1-4757-3437-9.

[127] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. "The quickhull algorithm for convex hulls". In: *ACM Transactions on Mathematical Software (TOMS)* 22.4 (1996), pp. 469–483.

[128] Martin Campos Pinto et al. "Noiseless Vlasov–Poisson simulations with linearly transformed particles". In: *Journal of Computational Physics* 275 (2014), pp. 236–256.

[129] Danial Faghihi et al. "A Particle Down-Sampling Method with Application to Multifidelity Plasma Particle-in-Cell Simulations". In: *arXiv preprint arXiv:1702.05198* (2017).

[130] Jonathon Shlens. "A tutorial on principal component analysis". In: *arXiv preprint arXiv:1404.1100* (2014).

[131] Bedros Afeyan et al. "Kinetic electrostatic electron nonlinear (KEEN) waves and their interactions driven by the ponderomotive force of crossing laser beams". In: *arXiv preprint arXiv:1210.8105* (2012).

[132] Bedros Afeyan et al. "Simulations of kinetic electrostatic electron nonlinear (KEEN) waves with variable velocity resolution grids and high-order time-splitting". In: *The European Physical Journal D* 68.10 (2014), pp. 1–21.

[133] RH Levy. "Diocotron instability in a cylindrical geometry". In: *Physics of Fluids (1958-1988)* 8.7 (1965), pp. 1288–1295.

[134] Michel Mehrenberger et al. "Solving the guiding-center model on a regular hexagonal mesh". In: *ESAIM: Proceedings and Surveys* 53 (2016), pp. 149–176.

[135] Yasutaro Nishimura et al. "A finite element Poisson solver for gyrokinetic particle simulations in a global field aligned mesh". In: *Journal of Computational Physics* 214.2 (2006), pp. 657–671.

[136] Rainald Löhner and John Ambrosiano. "A vectorized particle tracer for unstructured grids". In: *Journal of Computational Physics* 91.1 (1990), pp. 22–31.

[137] Eric Sonnendrücker, John J Ambrosiano, and Scott T Brandon. "A finite element formulation of the Darwin PIC model for use on unstructured grids". In: *Journal of Computational Physics* 121.2 (1995), pp. 281–297.

[138] A. Haselbacher, F.M. Najjar, and J.P. Ferry. "An efficient and robust particle-localization algorithm for unstructured grids". In: *Journal of Computational Physics* 225.2 (2007), pp. 2198 –2213. ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/j.jcp.2007.03.018. URL: //www.sciencedirect.com/science/article/pii/S002199910700126X.

[139] SB Kuang, AB Yu, and ZS Zou. "A new point-locating algorithm under three-dimensional hybrid meshes". In: *International Journal of Multiphase Flow* 34.11 (2008), pp. 1023–1030.

[140] AJ Peurrung and J Fajans. "Experimental dynamics of an annulus of vorticity in a pure electron plasma". In: *Physics of Fluids A: Fluid Dynamics* 5.2 (1993), pp. 493–499.

[141] Nicolas Crouseilles et al. "A new fully two-dimensional conservative semi-Lagrangian method: applications on polar grids, from diocotron instability to ITG turbulence". In: *The European Physical Journal D* 68.9 (2014), pp. 1–10.

[142] Jérôme Pétri. "Non-linear evolution of the diocotron instability in a pulsar electrosphere: two-dimensional particle-in-cell simulations". In: *Astronomy & Astrophysics* 503.1 (2009), pp. 1–12.

[143] Zhihong Lin and Liu Chen. "A fluid–kinetic hybrid electron model for electromagnetic simulations". In: *Physics of Plasmas* 8.5 (2001), pp. 1447–1450.

[144] Pierre Degond, Giacomo Dimarco, and Lorenzo Pareschi. "The moment-guided Monte Carlo method". In: *International Journal for Numerical Methods in Fluids* 67.2 (2011), pp. 189–213.

[145] Adam Speight. "A multilevel approach to control variates". In: (2009).

[146] Viktor K Decyk. "Description of Spectral Particle-in-Cell Codes from the UPIC Framework". In: *Presentation at ISSS-10* (2011). URL: `https://picksc.idre.ucla.edu/wp-content/uploads/2015/05/UPICModels.pdf`.

[147] Remi Lehe et al. "A spectral, quasi-cylindrical and dispersion-free Particle-In-Cell algorithm". In: *Computer Physics Communications* 203 (2016), pp. 66–82.

[148] Igor A Andriyash, Remi Lehe, and Agustin Lifschitz. "Laser-plasma interactions with a Fourier-Bessel particle-in-cell method". In: *Physics of Plasmas* 23.3 (2016), p. 033110.

[149] Sergio Briguglio et al. "Parallelization of plasma simulation codes: gridless finite size particle versus particle in cell approach". In: *Future Generation Computer Systems* 16.5 (2000), pp. 541–552.

[150] G Vlad et al. "Gridless finite-size-particle plasma simulation". In: *Computer physics communications* 134.1 (2001), pp. 58–77.

[151] AF Lifschitz et al. "Particle-in-Cell modelling of laser–plasma interaction using Fourier decomposition". In: *Journal of Computational Physics* 228.5 (2009), pp. 1803–1814.

[152] Adam Davidson et al. "Implementation of a hybrid particle code with a PIC description in r–z and a gridless description in ϕ into OSIRIS". In: *Journal of Computational Physics* 281 (2015), pp. 1063–1077.

[153] N Ohana et al. "Towards the optimization of a gyrokinetic Particle-In-Cell (PIC) code on large-scale hybrid architectures". In: *Journal of Physics: Conference Series*. Vol. 775. 1. IOP Publishing. 2016, p. 012010.

[154] Stephen D Webb. "A spectral canonical electrostatic algorithm". In: *Plasma Physics and Controlled Fusion* 58.3 (2016), p. 034007.

[155] AJ Klimas and WM Farrell. "A splitting algorithm for Vlasov simulation with filamentation filtration". In: *Journal of computational physics* 110.1 (1994), pp. 150–163.

[156] Lukas Einkemmer and Alexander Ostermann. "A strategy to suppress recurrence in grid-based Vlasov solvers". In: *The European Physical Journal D* 68.7 (2014), pp. 1–7.

[157] Enrico Camporeale et al. "On the velocity space discretization for the Vlasov-Poisson system: comparison between Hermite spectral and Particle-in-Cell methods. Part 1: semi-implicit scheme". In: *arXiv preprint arXiv:1311.2098* (2013).

[158] Enrico Camporeale et al. "On the velocity space discretization for the Vlasov-Poisson system: Comparison between implicit Hermite spectral and Particle-in-Cell methods". In: *Computer Physics Communications* 198 (2016), pp. 47–58.

[159] Michael Pippig. "PFFT: An extension of FFTW to massively parallel architectures". In: *SIAM Journal on Scientific Computing* 35.3 (2013), pp. C213–C236.

[160] BF McMillan and L Villard. "Accuracy of momentum and gyrodensity transport in global gyrokinetic particle-in-cell simulations". In: *Physics of Plasmas (1994-present)* 21.5 (2014), p. 052501.

[161] R.M. Gray. *Toeplitz and Circulant Matrices: A Review*. Foundations and Trends in Technology. Now Publishers, 2006. ISBN: 9781933019239. URL: `https://books.google.de/books?id=PrOi92L5dAUC`.

[162] Roger Peyret. *Spectral methods for incompressible viscous flow*. Vol. 148. Springer Science & Business Media, 2013.

[163] M. Unser. "Sampling—50 Years After Shannon". In: *Proceedings of the IEEE* 88.4 (2000), pp. 569–587. URL: `{http://bigwww.epfl.ch/publications/unser0001.html}`.

[164] WM Nevins et al. "Discrete particle noise in particle-in-cell simulations of plasma microturbulence". In: *Physics of Plasmas (1994-present)* 12.12 (2005), p. 122305.

[165] Jie Shen Tao Tang. "Spectral and High-Order Methods with Applications". In: (2006).

[166] Bradley K Alpert and Vladimir Rokhlin. "A fast algorithm for the evaluation of Legendre expansions". In: *SIAM Journal on Scientific and Statistical Computing* 12.1 (1991), pp. 158–179.

[167] Nicholas Hale and Alex Townsend. "A fast, simple, and stable Chebyshev–Legendre transform using an asymptotic formula". In: *SIAM Journal on Scientific Computing* 36.1 (2014), A148–A167.

[168] Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. "deal. II—a general-purpose object-oriented finite element library". In: *ACM Transactions on Mathematical Software (TOMS)* 33.4 (2007), p. 24.

[169] Lento Manickathan and Artur Palha. "Hybrid Eulerian-Lagrangian vortex particle method". PhD thesis. Ph. D. thesis, Delft University of Technology, 2015.

[170] T. A Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publications, 2014. URL: `http://www.chebfun.org/docs/guide/`.

[171] Sheehan Olver and Alex Townsend. "A practical framework for infinite-dimensional linear algebra". In: *High Performance Technical Computing in Dynamic Languages (HPTCDL), 2014 First Workshop for*. IEEE. 2014, pp. 57–62.

[172] Jie Shen. "Efficient spectral-Galerkin method I. Direct solvers of second-and fourth-order equations using Legendre polynomials". In: *SIAM Journal on Scientific Computing* 15.6 (1994), pp. 1489–1505.

[173] Jie Shen. "Efficient spectral-Galerkin method II. Direct solvers of second-and fourth-order equations using Chebyshev polynomials". In: *SIAM Journal on Scientific Computing* 16.1 (1995), pp. 74–87.

[174] Jie Shen. "Efficient spectral-Galerkin methods III: Polar and cylindrical geometries". In: *SIAM Journal on Scientific Computing* 18.6 (1997), pp. 1583–1604.

[175] Jie Shen. "Efficient spectral-Galerkin methods IV. Spherical geometries". In: *SIAM Journal on Scientific Computing* 20.4 (1999), pp. 1438–1455.

[176] Vladimir Zakharov. *Math 456 Lecture Notes: Bessel Functions and their Applications to Solutions of Partial Differential Equations*. University Lecture. 2009. URL: `{http://math.arizona.edu/~zakharov/BesselFunctionFinal.pdf}`.

[177] Milton Abramowitz, Irene A Stegun, et al. "Handbook of mathematical functions". In: *Applied mathematics series* 55 (1966), p. 62.

[178] Charles Cerjan. "Zernike-Bessel representation and its application to Hankel transforms". In: *JOSA A* 24.6 (2007), pp. 1609–1616.

[179] Aluizio Prata and WVT Rusch. "Algorithm for computation of Zernike polynomials expansion coefficients". In: *Applied Optics* 28.4 (1989), pp. 749–754.

[180] Rafael Navarro et al. "Generalization of Zernike polynomials for regular portions of circles and ellipses". In: *Optics express* 22.18 (2014), pp. 21263–21279.

[181] Sam Efromovich. "Orthogonal series density estimation". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.4 (2010), pp. 467–476.

[182] Gian Luca Delzanno. "Multi-dimensional, fully-implicit, spectral method for the Vlasov–Maxwell equations with exact conservation laws in discrete form". In: *Journal of Computational Physics* 301 (2015), pp. 338–356.

[183] Marlis Hochbruck and Alexander Ostermann. "Exponential integrators". In: *Acta Numerica* 19 (2010), pp. 209–286.

[184] Marlis Hochbruck. "A short course on exponential integrators". In: *Matrix functions and matrix equations* 19 (2015), p. 28.

[185] Emmanuel Frenod et al. "Long time behaviour of an exponential integrator for a Vlasov-Poisson system with strong magnetic field". In: *Communications in Computational Physics* 18.02 (2015), pp. 263–296.

[186] Emmanuel Frenod, Sever A Hirstoaga, and Mathieu Lutz. "Long-time simulation of a highly oscillatory Vlasov equation with an exponential integrator". In: *Comptes Rendus Mécanique* 342.10 (2014), pp. 595–609.

[187] Elena Celledoni, David Cohen, and Brynjulf Owren. "Symmetric exponential integrators with an application to the cubic Schrödinger equation". In: *Foundations of Computational Mathematics* 8.3 (2008), pp. 303–317.

[188] Ben-yu Guo and Zhong-qing Wang. "Legendre–Gauss collocation methods for ordinary differential equations". In: *Advances in Computational Mathematics* 30.3 (2009), pp. 249–280.

[189] Steven A Orszag. "Spectral methods for problems in complex geometries". In: *Journal of Computational Physics* 37.1 (1980), pp. 70–92.

[190] V. Grandgirard et al. "A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation". In: *Journal of Computational Physics* 217.2 (2006), pp. 395–423. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2006.01.023. URL: http://www.sciencedirect.com/science/article/pii/S0021999106000155.

[191] Parry Moon and Domina E Spencer. *Field theory handbook: including coordinate systems, differential equations and their solutions*. Springer, 2012.

[192] Virginie Grandgirard et al. "Computing ITG turbulence with a full-f semi-Lagrangian code". In: *Communications in Nonlinear Science and Numerical Simulation* 13.1 (2008), pp. 81–87.

[193] Francis Filbet and Chang Yang. *Mixed semi-Lagrangian/finite difference methods for plasma simulations*. 2014. eprint: arXiv:1409.8519.

[194] Jack E Volder. "The birth of CORDIC". In: *Journal of VLSI signal processing systems for signal, image and video technology* 25.2 (2000), pp. 101–105.

[195] John C Mason and David C Handscomb. *Chebyshev polynomials*. CRC Press, 2002.

[196] Stef Graillat and Valérie Ménissier-Morain. "Accurate summation, dot product and polynomial evaluation in complex floating point arithmetic". In: *Information and Computation* 216 (2012), pp. 57–71.

[197] Viktor K. Decyk and Tajendra V. Singh. "Particle-in-Cell algorithms for emerging computer architectures". In: *Computer Physics Communications* 185.3 (2014), pp. 708–719. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2013.10.013. URL: http://www.sciencedirect.com/science/article/pii/S001046551300341X.

[198] Viktor K Decyk. "Skeleton PIC codes for parallel computers". In: *Computer Physics Communications* 87.1-2 (1995), pp. 87–94.

[199] Jeff Bezanson et al. "Julia: A fresh approach to numerical computing". In: *SIAM Review* 59.1 (2017), pp. 65–98.

[200] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. "The NumPy array: a structure for efficient numerical computation". In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[201] Andreas Klöckner et al. "PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation". In: *Parallel Computing* 38.3 (2012), pp. 157–174. ISSN: 0167-8191. DOI: 10.1016/j.parco.2011.09.001.

[202] Marat Dukhan and Richard Vuduc. "Methods for high-throughput computation of elementary functions". In: *International Conference on Parallel Processing and Applied Mathematics*. Springer. 2013, pp. 86–95.

[203] Ira B Bernstein, John M Greene, and Martin D Kruskal. "Exact nonlinear plasma oscillations". In: *Physical Review* 108.3 (1957), p. 546.

[204] J Ameres, K Kormann, and E Sonnendrücker. "Particle in Fourier Discretization of Kinetic Equations". In: *PASC: Proceedings of the Platform for Advanced Scientific Computing Conference*. Lausanne, Switzerland: ACM, 2016. ISBN: 978-1-4503-4126-4.

[205] B Izrar et al. "Integration of vlasov equation by a fast fourier eulerian code". In: *Computer physics communications* 52.3 (1989), pp. 375–382.

[206] Bengt Eliasson. "Outflow boundary conditions for the Fourier transformed one-dimensional Vlasov–Poisson system". In: *Journal of scientific computing* 16.1 (2001), pp. 1–28.

[207] Bengt Eliasson. "Numerical modelling of the two-dimensional Fourier transformed Vlasov–Maxwell system". In: *Journal of Computational Physics* 190.2 (2003), pp. 501–522.

[208] Francis Filbet, Eric Sonnendrücker, and Pierre Bertrand. "Conservative numerical schemes for the Vlasov equation". In: *Journal of Computational Physics* 172.1 (2001), pp. 166–187.

[209] TD Arber and RGL Vann. "A critical comparison of Eulerian-grid-based Vlasov solvers". In: *Journal of computational physics* 180.1 (2002), pp. 339–357.

[210] Andreas Denner, Oliver Junge, and Daniel Matthes. "Computing coherent sets using the Fokker-Planck equation". In: *arXiv preprint arXiv:1512.03761* (2015).

[211] L Pareschi, G Russo, and G Toscani. "Fast spectral methods for the Fokker–Planck–Landau collision operator". In: *Journal of Computational Physics* 165.1 (2000), pp. 216–236.

[212] Hong Qin et al. "Comment on "Hamiltonian splitting for the Vlasov–Maxwell equations"". In: *Journal of Computational Physics* 297 (2015), pp. 721–723.

[213] Yang He et al. "Hamiltonian time integrators for Vlasov-Maxwell equations". In: *Physics of Plasmas* 22.12 (2015), p. 124503.

[214] Awad H Al-Mohy and Nicholas J Higham. "Computing the action of the matrix exponential, with an application to exponential integrators". In: *SIAM journal on scientific computing* 33.2 (2011), pp. 488–511.

[215] Cleve Moler and Charles Van Loan. "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later". In: *SIAM review* 45.1 (2003), pp. 3–49.

[216] Robert I McLachlan. "On the numerical integration of ordinary differential equations by symmetric composition methods". In: *SIAM Journal on Scientific Computing* 16.1 (1995), pp. 151–168.

[217] Yingda Cheng et al. "Discontinuous Galerkin Methods for the Vlasov–Maxwell Equations". In: *SIAM Journal on Numerical Analysis* 52.2 (2014), pp. 1017–1049.

[218] Evangelos Siminos, Didier Bénisti, and Laurent Gremillet. "Stability of nonlinear Vlasov-Poisson equilibria through spectral deformation and Fourier-Hermite expansion". In: *Physical Review E* 83.5 (2011), p. 056402.

[219] Roberto Szechtman. "Control variate techniques for monte carlo simulation: control variates techniques for monte carlo simulation". In: *Proceedings of the 35th conference on Winter simulation: driving innovation*. Winter Simulation Conference. 2003, pp. 144–149.

[220] Anaïs Crestetto, Nicolas Crouseilles, and Mohammed Lemou. "Kinetic/fluid micro-macro numerical schemes for Vlasov-Poisson-BGK equation using particles". In: *Kinetic and related models* 5.4 (2012), pp. 787–816.

[221] Francis Filbet and Chang Yang. "Mixed semi-Lagrangian/finite difference methods for plasma simulations". In: *arXiv preprint arXiv:1409.8519* (2014).

[222] Marc Buffat and Lionel Le Penven. "A spectral fictitious domain method with internal forcing for solving elliptic PDEs". In: *Journal of Computational Physics* 230.7 (2011), pp. 2433–2450.

[223] Nicolas Crouseilles, Lukas Einkemmer, and Martina Prugger. "An exponential integrator for the drift-kinetic model". In: *arXiv preprint arXiv:1705.09923* (2017).

[224] Sergey Mikhailovich Ermakov and VG Zolotukhin. "Polynomial approximations and the Monte-Carlo method". In: *Theory of Probability & Its Applications* 5.4 (1960), pp. 428–431.

[225] J. Hammersley. *Monte Carlo Methods*. Monographs on Statistics and Applied Probability. Springer Netherlands, 2013. ISBN: 9789400958197. URL: `https://books.google.de/books?id=3rDvCAAAQBAJ`.

[226] Phelim Boyle, Mark Broadie, and Paul Glasserman. "Monte Carlo methods for security pricing". In: *Journal of economic dynamics and control* 21.8 (1997), pp. 1267–1321.

[227] Roberto Szechtman and Peter W Glynn. "Constrained Monte Carlo and the method of control variates". In: *Proceedings of the 33nd conference on Winter simulation*. IEEE Computer Society. 2001, pp. 394–400.

[228] José A Bittencourt. *Fundamentals of plasma physics*. Springer Science & Business Media, 2013.

[229] Donald Gary Swanson. *Plasma kinetic theory*. CRC Press, 2008.

[230] Guangye Chen, Luis Chacón, and Daniel C Barnes. "An energy-and charge-conserving, implicit, electrostatic particle-in-cell algorithm". In: *Journal of Computational Physics* 230.18 (2011), pp. 7018–7036.

[231] G Manzini et al. "A Legendre–Fourier spectral method with exact conservation laws for the Vlasov–Poisson system". In: *Journal of Computational Physics* 317 (2016), pp. 82–107.

[232] Ulrich Stroth et al. "Experimental turbulence studies for gyro-kinetic code validation using advanced microwave diagnostics". In: *Nuclear fusion* 55.8 (2015), p. 083027.

[233] Mirko Salewski et al. "Tomography of fast-ion velocity-space distributions from synthetic CTS and FIDA measurements". In: *Nuclear Fusion* 52.10 (2012), p. 103008.

[234] Hong Qin et al. "Canonical symplectic particle-in-cell method for long-term large-scale simulations of the Vlasov–Maxwell equations". In: *Nuclear Fusion* 56.1 (2015), p. 014001.

[235] KL Cartwright, JP Verboncoeur, and CK Birdsall. "Nonlinear hybrid Boltzmann–particle-in-cell acceleration algorithm". In: *Physics of Plasmas* 7.8 (2000), pp. 3252–3264.

[236] Ahmed Ratnani. "Isogeometric analysis in plasma physics and electromagnetism". PhD thesis. Université de Strasbourg, 2011.

[237] Mark Andrews. "Alternative separation of Laplace's equation in toroidal coordinates and its application to electrostatics". In: *Journal of electrostatics* 64.10 (2006), pp. 664–672.

[238] B Ph Van Milligen and A Lopez Fraguas. "Expansion of vacuum magnetic fields in toroidal harmonics". In: *Computer physics communications* 81.1-2 (1994), pp. 74–90.

[239] Blaise Faugeras et al. "2D interpolation and extrapolation of discrete magnetic measurements with toroidal harmonics for equilibrium reconstruction in a Tokamak". In: *Plasma Physics and Controlled Fusion* 56.11 (2014), p. 114010.

[240] Blaise Faugeras. "Tokamak plasma boundary reconstruction using toroidal harmonics and an optimal control method". In: *Fusion Science and Technology* 69.2 (2016), pp. 495–504.

[241] V.D. Shafranov. "Equilibrium of a plasma toroid in a magnetic field". In: *SOVIET PHYSICS JETP-USSR* 10.4 (1960), pp. 775–779.

[242] Amparo Gil and Javier Segura. "{DTORH3} 2.0: A new version of a computer program for the evaluation of toroidal harmonics". In: *Computer Physics Communications* 139.2 (2001), pp. 186 –191. ISSN: 0010-4655. DOI: http://dx.doi.org/10.1016/S0010-4655(01)00188-6. URL: http://www.sciencedirect.com/science/article/pii/S0010465501001886.

[243] Amparo Gil, Javier Segura, and Nico M Temme. "Computing toroidal functions for wide ranges of the parameters". In: *Journal of Computational Physics* 161.1 (2000), pp. 204–217.

[244] J Segura and A Gil. "Evaluation of toroidal harmonics". In: *Computer physics communications* 124.1 (2000), pp. 104–122.

[245] Serdar Kuyucak, Matthew Hoyles, and Shin-Ho Chung. "Analytical Solutions of Poisson's Equation for Realistic Geometrical Shapes of Membrane Ion Channels". In: *Biophysical Journal* 74.1 (1998), pp. 22 –36. ISSN: 0006-3495. DOI: http://dx.doi.org/10.1016/S0006-3495(98)77763-X. URL: http://www.sciencedirect.com/science/article/pii/S000634959877763X.

[246] Matthew Hoyles, Serdar Kuyucak, and Shin-Ho Chung. "Solutions of Poisson's equation in channel-like geometries". In: *Computer Physics Communications* 115.1 (1998), pp. 45–68.

[247] RL Miller et al. "Noncircular, finite aspect ratio, local equilibrium model". In: *Physics of Plasmas* 5.4 (1998), pp. 973–978.

[248] Weston M Stacey. "Applications of the Miller equilibrium to extend tokamak computational models". In: *Physics of Plasmas* 15.12 (2008), p. 122505.

[249] Antoine J Cerfon and Jeffrey P Freidberg. ""One size fits all" analytic solutions to the Grad–Shafranov equation". In: *Physics of Plasmas* 17.3 (2010), p. 032502.

[250] SP Hirshman, P Merkel, et al. "Three-dimensional free boundary calculations using a spectral Green's function method". In: *Computer Physics Communications* 43.1 (1986), pp. 143–155.

[251] HP Callaghan, PJ McCarthy, and J Geiger. "Fast equilibrium interpretation on the W7-AS stellarator using function parameterization". In: *Nuclear Fusion* 39.4 (1999), p. 509.

[252] J Nührenberg and R Zille. "Quasi-helically symmetric toroidal stellarators". In: *Physics Letters A* 129.2 (1988), pp. 113–117.

[253] Nicolas Crouseilles et al. "Semi-Lagrangian simulations on polar grids: from diocotron instability to ITG turbulence". Feb. 2014. URL: https://hal.archives-ouvertes.fr/hal-00977342.

[254] CW Clenshaw. "A note on the summation of Chebyshev series". In: *Mathematics of Computation* 9.51 (1955), pp. 118–120.

[255] Fei Liu, Xingde Ye, and Xinghua Wang. "Efficient Chebyshev spectral method for solving linear elliptic PDEs using quasi-inverse technique". In: *Numerical Mathematics: Theory, Methods and Applications* 4.02 (2011), pp. 197–215.

[256] Matteo Frigo and Steven G. Johnson. "The Design and Implementation of FFTW3". In: *Proceedings of the IEEE* 93.2 (2005). Special issue on "Program Generation, Optimization, and Platform Adaptation", pp. 216–231.

[257] Marco Venturini, Robert Warnock, and Alexander Zholents. "Vlasov solver for longitudinal dynamics in beam delivery systems for x-ray free electron lasers". In: *Physical Review Special Topics-Accelerators and Beams* 10.5 (2007), p. 054403.

[258] Ronald C Davidson and Edward A Startsev. "Self-consistent Vlasov-Maxwell description of the longitudinal dynamics of intense charged particle beams". In: *Physical Review Special Topics-Accelerators and Beams* 7.2 (2004), p. 024401.

[259] K Kormann, K Reuter, and E Sonnendrücker. "Parallelization Strategies for a Semi-Lagrangian Vlasov Code". In: *Platform for Advanced Scientific Computing Conference 2016 (PASC16)*. 2016.

[260] K Kormann et al. *Massively parallel semi-Lagrangian solution of the 6d Vlasov-Poisson problem*. NUMKIN 2016 : International Workshop on Numerical Methods for Kinetic Equations. URL: http://www-irma.u-strasbg.fr/IMG/pdf/kormann_numkin2016.pdf.

[261] Katharina Kormann. "A semi-Lagrangian Vlasov solver in tensor train format". In: *SIAM Journal on Scientific Computing* 37.4 (2015), B613–B632.

[262] Katharina Kormann. "Solving the 6D Vlasov Equation in Tensor Train Format". In: *European Numerical Mathematics and Advanced Applications (ENUMATH 2015)*. 2015.