High-Order Particle Integration for Particle-In-Cell Schemes using Boris with Spectral Deferred Corrections

Kristoffer Smedt

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

EPSRC Centre for Doctoral Training in Fluid Dynamics

May 2021

Declaration

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Acknowledgements

This thesis would not have been possible without the advice and support of my supervisors Jitse Niesen, Daniel Ruprecht and Steve Tobias. I would like to thank them for all they have taught me and promise it will not go to waste. I would also like to thank my family and friends without whom I would not be where I am today. Finally, I would like to acknowledge the wealth of support and ideas provided by my fellow research students and all the staff of the Leeds University CDT for Fluid Dynamics.

Abstract

The study of plasmas plays an important role in both science and technology. Plasma dynamics can be found wherever charged particles or materials interact with and generate electromagnetic fields, covering more orders of magnitude in scale and density than any other type of matter. Plasma phenomena dominate the dynamics of the sun, stars and space between them and are important to a variety of technologies, from fusion reactors to spacecraft propulsion. As plasma behaviour and associated mathematical relations are naturally complex, numerical methods and computer simulation play a crucial role in furthering the field. Particle-in-Cell (PIC) is a class of numerical scheme currently used in the simulation of hot diffuse plasmas, or denser plasmas at small scales. One crucial part of such schemes is the particle integrator, which solves the particle equations of motion, typically via time discretisation of the Newton-Lorentz force. For nearly forty years, the dominant algorithm for charged particle tracking has been leapfrog integration using Boris' algorithm. The combined scheme is often referred to simply as the classic Boris integrator and provides a directly computable centre-difference discretisation for the implicit system. As the Boris algorithm is intrinsically second order accurate, a tunable order algorithm based on Boris and spectral deferred corrections (Boris-SDC) was recently proposed and demonstrated to exhibit high order time convergence for a single-particle Penning trap with exactly known electromagnetic fields. The faster reduction in error as time-step size is decreased allowed Boris-SDC to be more computationally efficient than classic Boris, but whether the advantageous characteristics of Boris-SDC would extend to PIC and approximated fields was not investigated. This thesis contributes the implementation and performance testing of Boris-SDC within PIC schemes and generalises Boris-SDC to the relativistic regime. This relativistic extension to Boris-SDC is shown to retain higher order time convergence and improved computational performance when compared to classic Boris even in highly relativistic regimes ($> 99\%$ speed of light). The relativistic

Boris-SDC integrator is shown to produce less unphysical drift than classic Boris in the force-free scenario where electric and magnetic forces cancel. The algorithmic modifications required to implement Boris-SDC within PIC are highlighted first and the impact of spatial electric field approximation on particle integrator performance is demonstrated for the electrostatic case (ESPIC). The relativistic Boris-SDC integrator is then derived and implemented within the open-source PIC code Runko, demonstrating capability of Boris-SDC to work with existing codes. Finally, performance tests are conducted in the form of work-precision comparisons to classic Boris for two electrostatic benchmarks, the two-stream instability and Landau damping. The spatial field approximation inherent to ESPIC imposes an error saturation on the time convergence of the particle integrator, inversely proportional to the spatial resolution, which limits the achievable global error. The limited accuracy is found to erode the computational performance of Boris-SDC, as the low level of error required to offset the added computational cost of SDC cannot be reached. Above the spatial saturation point however, Boris-SDC is found to retain high order time convergence and higher accuracy for a fixed time-step size than classic Boris. As a final note, suggestions are given for further work, including use of higher order spatial methods, investigation on the significance of momentum error vs. spatial error as well as PIC applications wherein Boris-SDC might be useful despite the lack of a clear performance gain.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The following chapter outlines the context, aims and structure of this thesis, starting with the most fundamental background to understand the problems solved and techniques described later in the document. The first section starts with a description of the main physical topic of interest, plasma, the main features thereof and then provides a brief introduction to the modelling tools under investigation in this study, pointing out the novel contributions where relevant.

This fundamental introduction is then followed by the specific aims and objectives that the main body of work is meant to achieve, with the aim describing the overall direction of the project and objectives outlining the specific activities undertaken to follow through.

Lastly, an overview of the thesis layout is provided along with a description of each chapter, this covers the main contents along with a short comment on how these contents are pertinent to the project and thesis as a whole.

1.1 Background and Motivation

Plasma is known as the fourth phase of matter and it has been estimated that plasma makes up 99% of the matter in the universe [16]. Stars and the interstellar medium account for most of this, due to the incredible mass of the former and extreme volume of the latter. The typical plasma arises when gas reaches extremes of temperature and/or pressure where it ionises. Ionisation is when the atomic structure of the electrically neutral gas molecules breaks down and becomes a collection of individually charged particles. These conditions make terrestrial plasmas rare and the relevance to life on Earth can seem small. Lightning and aurora are perhaps the only two striking examples of plasma on Earth, but the central role of plasma dynamics in stellar and interstellar processes make the topic crucially important in studying the universe at large. Technological advancement has also given rise to human made plasmas, notably in thermonuclear reactors and spacecraft exhaust plumes, providing another incentive to understand the physics that distinguish this form of matter. The crucial feature of plasma that separates it physically from gas is that it responds to and generates electric and magnetic fields. In a plasma with equal numbers of positive and negative charges, the plasma as a whole is electrically neutral, ensuring that it does not immediately disintegrate from an electric charge imbalance. The individual movement of charges within the plasma generates currents and electromagnetic fields, affecting the whole plasma. This collective behaviour of the particles making up the plasma distinguishes the dynamics from that of neutral gasses, in which the dynamics are dominated by individual collisions between molecules. The plasma is "quasi-neutral" which refers to the idea of global electric neutrality and local electromagnetic interactions. Three factors are crucial in defining and mathematically modelling plasmas:

1. The so-called plasma approximation is valid when the number of opposite charges within a sphere surrounding a given charged particle is large enough to cancel out the electric field of the particle outside the sphere. The sphere is

referred to as the Debye sphere and the radius of the sphere is known as the Debye length, calculated via

$$\lambda_D = \left(\frac{K_B T_e}{n e^2} \right)^{\frac{1}{2}}, \tag{1.1}$$

in SI units, where K_B is the Boltzmann constant, T_e is the electron temperature, n is the number density of the plasma and e is the elementary charge. The number of particles in the Debye sphere is then $N_D = n\lambda_D^3$, and it is required that $N_D \gg 1$ for the plasma approximation to be valid.

2. The second criteria considers the collective behaviour mentioned above. The size of the plasma must be much larger than the Debye sphere

$$L \gg \lambda_D, \tag{1.2}$$

where L is the characteristic length-scale of the plasma under consideration. If the internal bulk of the plasma isn't sufficiently large compared to the Debye sphere, the internal electric interactions are too significant to ignore and the plasma isn't quasi-neutral.

3. The third criteria involves natural oscillations in the plasma. Consider a plasma in which the ions and electrons are displaced by some distance collectively. The lighter electrons will respond quickly to their electric attraction to the ions and accelerate towards them. Due to inertia, the mass of electrons will overshoot the ions, decelerate and return, creating an oscillating system with the frequency ω_p, the plasma frequency. When the plasma frequency is larger than the frequency of physical collisions between particles, electrostatic interactions with neighbouring particles dominate over gas kinetics and the gas can be considered a plasma.

The Debye length is important in two of the three plasma criteria, and so is a key parameter in the definition of plasmas. It essentially represents the length over

which the electric potential of a given particle acts for a specific plasma. Due to the electric attraction between opposite charges, any positive charge concentration that arises will quickly attract a sheath of electrons around it. This shields out the potential with respect to the rest of the plasma and is called "Debye shielding". Both the Debye length and plasma frequency are defined using electrons since they are typically the lightest and most mobile particles in a plasma and thus does most of the electric shielding and oscillation. Due to the thermal kinetic energy of the electrons in a real plasma, the Debye sheath has a finite thickness as the thermal motion works against the electric potential.

Quasi-neutrality then refers to the idea that from the macroscopic perspective, all electric potentials are eliminated via Debye shielding, but on the microscopic level charges still exert electric force on neighbouring particles over the Debye length. The result is an ionised gas that is neutral on the whole, but still exhibits electromagnetic dynamics.

The three plasma criteria listed above allows for a massive variety of possible plasmas with vastly different densities, temperatures and sizes. Immediately recognisable examples of two possible extremes are the Sun and Space itself. The core of the sun is a highly pressurised plasma with a density of particles per volume of $10^{32}\,\mathrm{m}^{-3}$ and a Debye length of $10^{-11}\,\mathrm{m}$. On the opposite end of the spectrum, space is not empty, but filled with diffuse plasma that ranges in density from $10^{6}\,\mathrm{m}^{-3}$ in the solar wind down to $1\,\mathrm{m}^{-3}$ in the intergalactic medium. The different Debye lengths for these two cases are then 10 metres and 100 kilometres respectively [10]. Note how density and Debye length are generally inversely proportional depending on the temperature, as per (1.1).

The mathematical models used to model plasma are not exclusively useful to cases where the three criteria are met however. Plasma-like behaviour can be found in a variety of other problems that involve charged particles, and inclusive use of the term plasma can be used to cover almost any collection of charged particles,

regardless of Debye length, temperature and density [83].

Plasmas when defined in this wider context are studied in a wide range of fields. In the field of stellar dynamics, the wave dynamics of the solar and stellar coronae was studied by Nakariakov [55] using plasma fluid modelling. For the physics of the interplanetary media, Mann et al. [51] studied the interactions of dust and plasma in the Earth's ionosphere, using experimental data from spacecraft, radar and optical observations. Arber et al. [4] reviewed the use of particle simulation codes in the modelling of laser-plasma interactions, including recent applications to problems such as short-pulse laser-solid interactions, fast-electron transport and quantum electrodynamical effects. In spacecraft propulsion, ion thruster design requires ready knowledge of plasma behaviour and the work by Andrenucci et al. [3] show the application of plasma modelling to understanding the exhaust dynamics of these thrusters. In semi-conductor fabrication, plasma is used to etch features which are too small for traditional tools to produce and the dynamics of different particle species in these plasmas were studied Bera et al. [7] using particle focused modelling. For controlled fusion, both dense and diffuse plasmas are studied via two competing methods for achieving fusion [16]. In inertial confinement fusion, high density, low Debye length plasma is created by rapid heating and pressurisation of a fuel pellet using lasers; pertinent dynamics include plasma shockwaves such as those studied by Taitano et al. [74]. In magnetic confinement fusion, a high temperature, low density plasma is confined using strong magnetic fields. The inherent instabilities of this setup have been studied for decades [41] and important instabilities continue to be studied, such as Edge Localised Modes (ELMs) which are highly disruptive to reactor operations [43].

Computer simulation of plasmas and the required numerical methods can largely be categorised by the ratio of problem length-scale L to the Debye length [8]. For dense plasmas or very large scale diffuse plasmas, resolving the particle dynamics is

not possible and fluid methods are used, where the degrees-of-freedom are reduced to macroscopic quantities like bulk density and velocity. The assumptions and approximations required to reduce the plasma modelling to macroscopic quantities often break down for plasmas where kinetic effects are important. Additionally, while reducing the degrees of freedom is fine where these are not of interest, such reduction is not desirable where the detailed microscopic information *is* of interest. For thin plasmas or small-scale problems that can be modelled at, or close to, the particle level, more fundamental "kinetic methods" can be applied, at the cost of increased computational expense. Kinetic methods involve solution of the fundamental Vlasov-Maxwell system, which is described in detail in Chapter 2. Particle methods, which resolve the actual particle dynamics to some degree, are a subtype of kinetic methods. The choice of mathematical model for plasma problems is thus dependent on scale, density and the dynamics of interest.

The main context of this project is the particle-in-cell method (PIC), a time-tested kinetic particle model for simulating plasma phenomena. The method employs the fundamental equations of particle motion and electromagnetics which helps ensure that most of the fundamental physics, including the non-linear effects, are retained. In practice, PIC resolves the dynamics of particles using Maxwell's equations for the electromagnetic fields and the Lorentz force equation for the motion of the particles. The use of particles and particle motion integration might suggest that physical particles must be resolved in the scheme, putting a harsh restriction on the length scales that are feasible to model with contemporary computational power, but this is not the case. PIC is not restricted to the scale of physical particles and while it can be used at this scale, it is more common for each computational particle to represent millions of physical particles. These so-called "macro-particles" can be seen as discrete elements of the particle density distribution function in position/velocity phase-space, tracked in a Lagrangian frame. The elements allow for efficient discretisation of the phase-space distribution, which will often contain large areas of zero or near-zero density. When a density distribution with such voids

is discretised using traditional methods such as a fixed grid, the result is a waste of computational resources, as large parts of the grid contribute almost nothing to the solution.

From the perspective of phase-space discretisation, a PIC scheme presents a numerical solution to the Maxwell-Vlasov system of equations, which is one of the most fundamental models for describing the dynamics of large distributions of charged particles. For a detailed treatment of this more formal mathematical justification for PIC, refer to Chapter 2 as well as the appropriate literature by Pukhov [59] as well as Hockney and Eastwood [37].

The PIC scheme as a whole consists of separate individual algorithms coupled together. Particles are defined in continuous space while the electric and magnetic fields are defined on a static grid. Coupling between particles and grid is achieved via interpolation of charge or current (from particles to grid) and fields (from grid to particles). The charge and current are used to establish and/or advance the grid-defined field in time and the particles are moved in accordance with the interpolated fields by integration of the equations of motion. The main components of PIC are thus the spatial/temporal field solver, temporal particle integrator, and two instances of spatial interpolation algorithms. The various PIC schemes can be categorised into two broad categories: Electrostatic PIC (ESPIC) refers to PIC schemes with a steady or zero magnetic field and electromagnetic PIC refers to the full PIC scheme with electromagnetic fields that are driven by the current associated with particle motion and/or external factors such as laser light input.

Common to both is the need to advance the particles in time via integration of the equations of motion, which for charged particles come from the combination of Newton's Second Law $\mathbf{F} = m\frac{d\mathbf{v}}{dt}$ with the Lorentz force

$$\mathbf{F} = q\left[\mathbf{E} + \mathbf{v} \times \mathbf{B}\right],$$

where q is the particle charge, $\mathbf{v}$ is the particle velocity and $\mathbf{E}$, $\mathbf{B}$ are the electric and magnetic fields experienced by the particle respectively.

Integration of the resulting differential equation is typically performed by the particle integrator, also called the "particle pusher". The defacto standard particle pusher for PIC schemes is a simple "leapfrog" integrator where position and velocity are updated in staggered fashion, with velocity defined half a time-step removed from position. The integrator needs the Boris algorithm [11] to resolve the apparent implicity in the Newton-Lorentz system, where the derivative of velocity depend on the cross-product of magnetic field and velocity. The combined scheme, usually referred to simply as the "Boris integrator" has been the most popular choice of particle pusher for almost 40 years. The Boris integrator works by splitting the acceleration on the particles due to electric and magnetic forces respectively, aided by the geometric observation that the magnetic field can only ever rotate the particle trajectory. It is cheap and efficient but intrinsically second order accurate, i.e. error decreases with the square of step-size [83].

In this configuration, using the Boris integrator as the particle pusher, PIC has been used successfully for a wide variety of plasma problems. Several of the plasma applications listed above made use of particle-in-cell codes for numerical studies. The review by Arber et al. [4] focuses on the core algorithms used by contemporary PIC codes to simulate laser-plasma interactions, including handling of collisions, ionisation, and quantum electrodynamic effects (QED). A few recent applications included in the review are stimulated Raman scattering used in spectroscopy, laser-solid interactions for material property studies and recent investigations into the QED effects that arise in laser-plasma problems. The study by Andrenucci et al. [3] describes the advances in PIC simulation for ion thrusters and their exhaust at a time when these were maturing and becoming useful to the design process. The authors demonstrated good agreement between contemporary 2D simulations and experimental data and development was rapidly moving towards fully 3D simula-

tions. Today, commercial codes are regularly used in the design of ion thrusters [78], where particle modelling can allow for accurate analysis of plasma channel erosion due to ion impingement, predicting product life-cycle issues during the design stage. The study by Bera et al. [7] into capacitively coupled oxygen plasmas used for semi-conductor etching was motivated by the increasing relevance of kinetic effects as etching applications transitioned to low pressure plasmas more suitable for the shrinking microelectronic features. Using a 1D PIC model combined with a fluid model for the neutral atoms, the authors demonstrated decreased electron densities for oxygen as opposed to argon plasmas.

Commercial applications of PIC and similar simulation software can be extended and improved if the underlying numerical models are improved with better accuracy and/or computational efficiency. Faster simulation at comparable accuracy opens the door for larger and more complex simulations. One way to advance these aspects is through higher order simulation methods which can produce higher accuracy for a given resolution of numerical discretisation and faster increase in accuracy as this resolution is increased. Recently, a higher order generalisation of the Boris method was introduced by Winkel et al. [92], which works by using the Boris algorithm in conjunction with spectral deferred corrections (Boris-SDC). The new high-order integrator outperformed a position-velocity synchronised variant of the Boris integrator in terms of accuracy vs. computational expense in a non-PIC test case using steady fields.

These results were promising, particularly for the use of Boris-SDC in modelling pure particle motion where electromagnetic fields are known or can be computed exactly. In PIC however, the field values provided to the integrator are approximations and it was unclear how well the favourable performance of Boris-SDC would hold up within a PIC scheme. The performance of a higher order particle integrator within traditional PIC schemes appears to remain unexplored, despite other high

order integrators and new high order PIC schemes being developed. For example, Li and Wang [47] developed an arbitrary order particle integration scheme and compared two versions to standard Boris in terms of error and CPU time. However, no comment was made on computational efficiency and both high-order integrators were implicit. As an example of complete PIC reformulation, Xiao et al. [94] developed a new explicit high order PIC scheme but did not test performance vs. traditional PIC schemes nor investigate the performance impact of high order particle integration alone. The purpose of the current study is to put Boris-SDC to the test and compare computational performance of PIC schemes when Boris is used as the particle integrator vs. the same PIC scheme using Boris-SDC.

Several commercial and open-source software packages for plasma simulation based on the PIC/Boris method exist [57], [77], [23]. The explicit and self-contained nature of Boris-SDC should allow ease of integration with existing codes. If Boris-SDC retains its advantageous characteristics in the PIC testing, the algorithm could prove beneficial for more well-developed PIC frameworks, whether commercial or open-source. Successful implementation would serve both to allow further testing for more advanced applications outside the scope of the current study and prove the commercial value of the work in the process.

1.2 Aims and Objectives

The aim of the project is to evaluate the characteristics and benefits of Boris-SDC as the particle integrator of a PIC code. This includes comparing Boris-SDC results with the classic Boris algorithm and gauging relative performance, ascertaining under what conditions the favourable convergence characteristics of Boris-SDC can be reproduced (if at all) as well as identifying any other interesting or beneficial effects of using Boris-SDC as a PIC particle integrator. The following set of objectives were defined to achieve this goal:

1. Review necessary background on mathematical modelling of plasmas, the principles and methodology of PIC schemes as well as the specifics of integrating the particle equations of motion in electromagnetic fields. These reviews will provide the necessary knowledge to construct and work directly with PIC codes as well as identify the alternative approaches to the problems PIC, Boris and high-order particle integration have been designed to solve.

2. Construct a new electrostatic PIC code, affording direct control of the Boris-SDC implementation process. This will provide deeper insight into the interaction between the two schemes and highlight any practical issues regarding the use of Boris-SDC as a direct substitute for the Boris algorithm, which was suggested as a potential benefit in the previous study. The choice of a custom ESPIC code is motivated by the need to understand in detail how PIC schemes operate and how each component interfaces with the others.

3. Using the custom ESPIC code, produce relative performance studies between Boris-SDC and Boris for two well-known electrostatic plasma instabilities in 1D, the two-stream instability and Landau damping [8]. The choice of these two problems cover both cold and thermal plasmas as well as linear and non-linear dynamics.

4. Extend Boris-SDC to the relativistic regime and implement within an existing, full-scale relativistic electromagnetic PIC code. The process might potentially raise more practical issues to highlight due to the difference in PIC grid definitions, scale and parallelism between the proof-of-concept code and larger, more sophisticated code-bases. Due to the increased complexity of EM-PIC compared to ESPIC, the implementation of a custom EMPIC code was deemed too time-consuming for the project.

1.3 Thesis Outline

As per the table of contents, the thesis is structured into chapters on existing literature, the collected implementation insight and preliminary results, followed by the final performance studies for ESPIC.

The literature review begins in Chapter 2 with the introduction of general macroscopic plasma modelling followed by derivations of fluid and particle methods from this point. The specific workings and evolution of the PIC method are then expanded upon in Chapter 3. Lastly, Chapter 4 describes the various time-integrators for the particle equations of motion in electromagnetic fields. These will provide context for a section containing the derivation and explanation of the Boris-SDC integrator that is the focal point of the project.

Chapter 5 begins with a structural overview of the custom ESPIC code "KPPS" developed for the project, followed by the main modifications to the traditional ESPIC operational flow required to implement Boris-SDC and vice-versa. Preliminary results are obtained for verification purposes, both for Boris-SDC in isolation and in combination with each component of the spatial PIC solver. The electrostatic section is followed first by a short section deriving the relativistic formulation of Boris-SDC, then by a longer section detailing the implementation considerations of relativistic Boris-SDC with the chosen EMPIC code Runko. As for the first section, preliminary results are sought to verify the implementation and give an initial indication of the performance characteristics.

Chapter 6 consists of two parts, each dedicated to one of the two chosen test cases (electrostatic two-stream instability and Landau damping respectively). Each part contains an introduction to the relevant problem, demonstrating what to expect and providing any theoretical or literature results. The chosen simulation parameters are presented and justified, followed by the performance results and discussions thereof.

Chapter 7 contains a summary of findings and conclusions as well as recommendations for further work.

The latest versions of the codes used in the current study can be found at the following online repositories:

- KPPS - https://github.com/Krissmedt/KPPS
- Runko - https://github.com/natj/runko
- Runko/Boris-SDC implementation - https://github.com/Krissmedt/runko
- Relativistic Boris-SDC - https://github.com/Krissmedt/rsdc

Chapter 2

Mathematical and Numerical Modelling of Plasma

To understand the dynamics of plasmas and the choices made in the computational study of thereof, be they necessary or merely convenient, the mathematical modelling of plasmas must also be understood. The following chapter will outline the most fundamental mathematical descriptions of plasmas, starting from the smallest scale: A collection of charged particles. The kinetic description, which is the main interest of this study, will be established on this foundation. From the kinetic description, the fluid models of plasma are then derived briefly, to provide perspective on the where the problems and computational methods of the current study fit in the grand scheme of plasma and fluid dynamics. The derivations in these sections follows closely that of Goossens [26]. Finally, following these foundational derivations, an introduction to the particle-in-cell method is given with an emphasis on how it relates to the fundamental mathematical model for the plasmas it is meant to describe.

2.1 The Vlasov-Maxwell Equations

To establish a mathematical model describing the dynamics of a plasma, one may begin by first considering the problem at the atomic level. All charged particles generate magnetic and electric fields, the sum of which yield the macroscopic fields seen in the plasma, which in turn influence the movement of the particles. However, the plasma consists of enormous numbers of these moving, charged particles; even some of the thinnest plasmas, like the solar wind, can contain thousands of particles per cubic metre. Establishing the position, velocity and field contributions of every particle thus becomes a massive, coupled problem. Even if the problem could be solved, the solution would involve an unnecessary level of detail: The position and velocity of every individual particle is not required to model the macroscopic flow of the plasma and thus particle-level resolution is rarely of interest. The natural next step in the mathematical modelling is therefore to attempt to aggregate the effect of the individual particles into macroscopic quantities more tractable to analysis.

2.1.1 The Distribution Function

Rather than considering discrete particles, consider a particle density distribution function $f(\mathbf{r}, \mathbf{w}, t)$ in the phase-space defined by position $\mathbf{r} = (x,\ y,\ z)$ and velocity $\mathbf{w} = (w_x,\ w_y,\ w_z)$. The density function will then give the number density of particles with the velocity $\mathbf{w}$ located at position $\mathbf{r}$ at the time t. Note that here $\mathbf{w}$ is velocity as an independent variable, i.e. a velocity *coordinate* representing a position in possible velocity space and a given particle can have any coordinate in the phase-space $(\mathbf{r}, \mathbf{w})$.

The number of particles of a specific species α in an infinitesimal volume of the

plasma can now be expressed as

$$dN_\alpha(\mathbf{r}, \mathbf{w}, t) = f_\alpha(\mathbf{r}, \mathbf{w}, t)\, d\mathbf{V}. \tag{2.1}$$

In other words, (2.1) gives the number of particles with a physical position infinitesimally close to $\mathbf{r}$ with a velocity infinitesimally close to $\mathbf{w}$ at time t. The distribution is assumed to be a continuous function, allowing no rapid variation between elements of phase-space.

Using f transitions the mathematical description from an N-body problem of attempting to model the interactions of N particles to dealing with one smooth function in 6D phase-space. Provided an equation or system of equations can be determined for the evolution of f, the complexity of describing the plasma particles will have been eliminated at the cost of knowledge about every individual particle.

Switching to a continuous density function rather than discrete particles is reasonable so long as each particle interacts with a multitude of other particles, not just a few close neighbours. This is the case for a high number of particles within the Debye sphere, which is the sphere with radius equal to the Debye length and represents the sphere of electrostatic influence of a particle in the plasma. The distribution function f then describes the density of particles in the infinitesimal phase-space volume element $d\mathbf{V} = dx\ dy\ dz\ dw_x\ dw_y\ dw_z$. The volume element must be large enough to contain a high number of particles for the continuity assumption to hold, but sufficiently small that the width of the element is insignificant compared to all spatial variation in physical quantities like temperature. The explicit assumptions made in the use of f are thus, that there are many particles in both the Debye sphere and the volume element $d\mathbf{V}$. These assumptions put limits on the applicability of the theory; if the length scale of any given problem under study is too small for instance, the assumptions that an infinitesimal volume contains many particles no longer holds. Likewise if the plasma studied is not dense enough for a given temper-

ature, the Debye sphere becomes sparsely populated and the interaction assumption breaks down.

In speaking about the distribution function, the terms *non-uniform* and *uniform* refer to whether or not the distribution function is independent of position, i.e. if f is uniform the particle density is independent of position and the plasma density is uniform in space and vice versa. For velocity, dependence or independence is referred to as *anisotropic* or *isotropic* respectively and refers to the direction of velocity rather than magnitude, e.g. an isotropic distribution function implies the same probability of a particle moving in any given coordinate direction.

2.1.2 Macroscopic Quantities

If a distribution function can be established, all information on the system can be extracted from it. The macroscopic parameters of the plasma as a fluid are then simply computed as follows. The number density of particles of species α particles in an infinitesimal physical volume at a given position, encompassing all velocities, is given by

$$n_\alpha(\mathbf{r}, t) = \int f_\alpha(\mathbf{r}, \mathbf{w}, t)\, d\mathbf{w}. \tag{2.2}$$

With the associated mass density given by

$$\rho_{m,\,\alpha}(\mathbf{r}, t) = n_\alpha(\mathbf{r}, t) m_\alpha, \tag{2.3}$$

where m_α is the mass of a single particle of species α.

The macroscopic 'fluid' velocity with respect to the resting frame of a whole species is then given by the weighted mean of the possible velocities $\mathbf{w}$ with the

density distribution f as a weighting function

$$\mathbf{v}_\alpha(\mathbf{r},t) = \langle \mathbf{w} \rangle = \frac{1}{n_\alpha(\mathbf{r},t)} \int \mathbf{w} f_\alpha(\mathbf{r},\mathbf{w},t)\, d\mathbf{w}. \tag{2.4}$$

Fluid velocity of the multi-species plasma is then the weighted mean of the species velocities

$$\mathbf{v}(\mathbf{r},t) = \frac{1}{\rho_m} \sum_\alpha \rho_{m,\alpha} \mathbf{v}_\alpha(\mathbf{r},t). \tag{2.5}$$

where ρ_m is the density of the bulk plasma accounting for each species

$$\rho_m = \sum_\alpha \rho_{m,\alpha}. \tag{2.6}$$

The velocity fluctuations u_α of the particles of a species with respect to the fluid as a whole then becomes

$$\mathbf{u}_\alpha(\mathbf{r},t) = \mathbf{w} - \mathbf{v}_\alpha(\mathbf{r},t), \tag{2.7}$$

where random in this context refers to the velocity deviation of a given particle with given velocity $\mathbf{w}$ from the macroscopic fluid velocity of the bulk plasma.

The thermal velocity and temperature of the particles is then defined using the mean kinetic energy $\frac{m_\alpha}{2}\langle |\mathbf{u}_\alpha(\mathbf{r},t)|^2 \rangle$, yielding

$$v_{t,\alpha}^2 = \frac{1}{3}\langle |\mathbf{u}_\alpha(\mathbf{r},t)|^2 \rangle = \frac{1}{n_\alpha} \int |\mathbf{u}_\alpha(\mathbf{r},t)|^2 f_\alpha(\mathbf{r},\mathbf{w},t)\, d\mathbf{w} = \frac{k_B T_\alpha}{m_\alpha}, \tag{2.8}$$

Where the *kinetic* temperature T of the whole plasma, defined by

$$3k_B nT = \sum_\alpha n_\alpha m_\alpha \langle |\mathbf{w} - \mathbf{v}(\mathbf{r},t)|^2 \rangle = \sum_\alpha m_\alpha \int |\mathbf{w} - \mathbf{v}(\mathbf{r},t)|^2 f_\alpha(\mathbf{r},\mathbf{w},t)\, d\mathbf{w}, \tag{2.9}$$

is then a measure of the kinetic energy of the deviating velocities with respect to the bulk plasma, where $k_B = 1.3807 \cdot 10^{-23}$ J K^{-1} is the Boltzmann constant.

Note that this temperature quantity is not the same as temperature under thermal equilibrium, wherein the velocity distribution of particles follow the symmetrical Maxwellian distribution. Rather, T in this case is simply a measure of the spread of the velocity distribution.

2.1.3 The Boltzmann Equation

To derive an equation for the evolution of the distribution function in time, start by considering the movement of the particles contained in the volume $d\mathbf{r}\, d\mathbf{w} = dx\ dy\ dz\ dw_x\ dw_y\ dw_z$ at position $(\mathbf{r}, \mathbf{w})$ at time t. For some given force function on the species $\mathbf{F}_\alpha$, at time $t' = t + \Delta t$ the particles will have moved to inhabit the volume $d\mathbf{r}'d\mathbf{w}'$ at position $(\mathbf{r}', \mathbf{w}')$.
With a given force F_α and acceleration $\mathbf{a}$, the change in phase-space position for an infinitesimal time-step Δt must be given by

$$\mathbf{r}' = \mathbf{r} + \mathbf{w}\Delta t, \quad \mathbf{w}' = \mathbf{w} + \mathbf{a}\Delta t, \quad \mathbf{a} = \frac{\mathbf{F}_\alpha}{m_\alpha}. \tag{2.10}$$

The number density of particles in the new volume can now be written

$$dN_\alpha(\mathbf{r}', \mathbf{w}', t') = f_\alpha(\mathbf{r}', \mathbf{w}', t')\, d\mathbf{r}\, d\mathbf{w}. \tag{2.11}$$

Writing the number density for the new volume in terms of the old variables yields

$$dN_\alpha(\mathbf{r}', \mathbf{w}', t') = f_\alpha(\mathbf{r} + \mathbf{w}\Delta t, \mathbf{w} + \mathbf{a}\Delta t, t + \Delta t) J\, d\mathbf{r}\, d\mathbf{w}. \tag{2.12}$$

The new volume element is given by the transformation $d\mathbf{r}'d\mathbf{w}' = Jd\mathbf{r}d\mathbf{w}$, where $J = det(M)$ is the Jacobian of the transformation from $(\mathbf{r}, \mathbf{w})$ to $(\mathbf{r}', \mathbf{w}')$ with $M = D(\mathbf{r}', \mathbf{w}')/D(\mathbf{r}, \mathbf{w})$. The diagonal terms of the six-by-six matrix M are equal to 1 and the rest vanish in the absence of particle collisions, giving $J = 1$. This indicates the element can change in shape but retains the volume at the original position.

The distribution function at $(\mathbf{r}', \mathbf{w}', t')$ can be written in terms of the original values by applying a Taylor series expansion up to the first order terms

$$f_\alpha(\mathbf{r}', \mathbf{w}', t') = f_\alpha(\mathbf{r}, \mathbf{w}, t) + \mathbf{w}\nabla_r f_\alpha(\mathbf{r}, \mathbf{w}, t) + \mathbf{a}\nabla_w f_\alpha(\mathbf{r}, \mathbf{w}, t) + \frac{\partial f_\alpha}{\partial t}\Delta t + O(\Delta t^2), \tag{2.13}$$

as f is assumed to be continuous and differentiable. With the number density of both old and new volumes both given in terms of $(\mathbf{r}, \mathbf{w}, t)$, subtracting dN_α and dN'_α yields

$$\begin{aligned} &dN_\alpha(\mathbf{r}', \mathbf{w}', t') - dN_\alpha(\mathbf{r}, \mathbf{w}, t) \\ &= \left[\left(\frac{\partial f_\alpha}{\partial t} + \mathbf{w}\cdot\nabla_r f_\alpha(\mathbf{r}, \mathbf{w}, t) + \mathbf{a}\cdot\nabla_w f_\alpha(\mathbf{r}, \mathbf{w}, t)\right)\Delta t + O(\Delta t^2)\right] d\mathbf{r}\, d\mathbf{w} \\ &= \left(\frac{\partial f_\alpha}{\partial t}\right)_{coll} d\mathbf{r}\, d\mathbf{w}\ \ \Delta t, \end{aligned} \tag{2.14}$$

by noticing where factors of Δt, $O(\Delta t^2)$ and $d\mathbf{r}d\mathbf{w}$ can be taken out.

Any difference in particle number density between the two volumes must be due to physical pair-wise particle collisions, wherein two particles physically touch and instantly exchange some momentum. These collisions, henceforth referred to as *binary* collisions, have yet to be considered. The difference is accounted for by equalling the change in number density to the so-called *collision operator* $(\partial f_\alpha/\partial t)_{coll}$ over the volume $d\mathbf{r}d\mathbf{w}$ and time Δt.

Dividing by the volume element $d\mathbf{r}\, d\mathbf{w}$ and Δt followed by taking the limit $\Delta t \to 0$, the classic *Boltzmann equation* is found to be

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{w}\cdot\nabla_r f_\alpha(\mathbf{r}, \mathbf{w}, t) + \mathbf{a}\cdot\nabla_w f_\alpha(\mathbf{r}, \mathbf{w}, t) = \left(\frac{\partial f_\alpha}{\partial t}\right)_{coll}. \tag{2.15}$$

If binary collisions between individual particles can be neglected, the equation becomes the *collisionless Boltzmann equation*, also called the *Vlasov equation*

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{w}\cdot\nabla_x f_\alpha(\mathbf{x}, \mathbf{w}, t) + \mathbf{a}\cdot\nabla_w f_\alpha(\mathbf{x}, \mathbf{w}, t) = 0. \tag{2.16}$$

Depending on the particle collision dynamics of the fluid under study, either the Boltzmann or Vlasov equation can now be used to study the evolution of the distribution function, given the relevant forcing term $\mathbf{F}_\alpha$. For a plasma, this forcing term depends on the electromagnetic fields, which in turn require the Maxwell equations to describe.

2.1.4 Maxwell's Equations and Plasma Electromagnetism

Maxwell's equations which relate the magnetic induction and electric field to the charge and current densities of the material, consists of Faraday's law, the divergence theorem for $\mathbf{B}$, Ampere's law (corrected) and Gauss' law. These are listed here in order using SI units and vector calculus formulation:

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon_0}\left(\frac{1}{\mu_0}\nabla \times \mathbf{B} - \mathbf{j}\right), \tag{2.17}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \tag{2.18}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{2.19}$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}, \tag{2.20}$$

where μ and ϵ are the vacuum magnetic permeability and electric permittivity respectively. For reference, these values are $\mu_0 = \pi \cdot 10^{-7}$ H m^{-1} and $\epsilon_0 = 8.8542 \cdot 10^{-12}$ F m^{-1} in SI units. The equations are coupled to the dynamics of the charged particles via the charge and current densities ρ and $\mathbf{j}$, which depend in turn on the macroscopic plasma quantities described above. Given an individual particle charge for a species q_α, the continuous charge and current densities associated with the

plasma distribution function f must be

$$\rho_\alpha(\mathbf{r}, t) = n_\alpha(\mathbf{r}, t) q_\alpha \tag{2.21}$$

and

$$\mathbf{j}_\alpha(\mathbf{r}, t) = \rho_\alpha(\mathbf{r}, t)\mathbf{v}_\alpha(\mathbf{r}, t), \tag{2.22}$$

respectively, which shows how the evolution of the distribution function affects the evolution of the electromagnetic fields.
The fields affect the particles in return, via an electromagnetic force acting on all charges within the fields. The Newton-Lorentz equations define the motion of charged particles in electromagnetic fields, and are needed to determine how the charge and current densities evolve along with the fields in a self-consistent manner.

To determine the electric field, Coulomb's law can be used to calculate the electric force between two particles as a function of their charge and relative position

$$\mathbf{F}_{E,1} = k_e \frac{q_1 q_2}{|\mathbf{r}_{21}|^2} \hat{\mathbf{r}}_{21}, \tag{2.23}$$

where $\mathbf{F}_{E,1}$ is the force on particle 1, $k_e = 9 \cdot 10^9\,\mathrm{Nm^2C^{-2}}$ is Coulomb's constant, q_i is the charge for particle i and $\mathbf{r}_{21}$ denotes the position vector from particle 2 to 1. Note how a positive product of the particle charges result in particle 1 being repelled from particle 2. For a large number of particles interactions, the force equation for a given particle with charge q_1 is simply the sum of interactions, which can be rewritten by taking out the charge as a common factor. The equation can then be rewritten, defining the electric field E acting on the particle:

$$\mathbf{F}_E = q_1 \mathbf{E} = q_1 \; k_e \sum_{j=1}^{N} \frac{q_j}{|\mathbf{r}_{j1}|^2} \hat{\mathbf{r}}_{j1}, \tag{2.24}$$

where N is the number of charges. Coulomb's law can be combined with imposed electric and magnetic fields to easily but expensively model the fields for an electro-

static plasma. The electrostatic case is defined by a temporally constant magnetic field, an assumption which hinges on the electric field being irrotational $\nabla \times \mathbf{E} = 0$ so that Faraday's law (2.18) gives $\frac{\partial \mathbf{B}}{\partial t} = 0$.

In the presence of both electric and magnetic fields, the force on any single charge is then given by the Newton-Lorentz force

$$\mathbf{F} = q\big[\mathbf{E}(\mathbf{r}, t) + \mathbf{w} \times \mathbf{B}(\mathbf{r}, t)\big], \tag{2.25}$$

The form of this force is worthy of note, as only the electric field contribution $q\mathbf{E}$ serves to push the particle in straight lines along the lines of the field as one might intuitively expect from a force field. The magnetic field contribution $\mathbf{w} \times \mathbf{B}$, called the Lorentz force, instead serves to rotate particles around the lines of the magnetic field. Plasma particle trajectories thus take the form of loops and helices, rotating in the plane normal to the local magnetic field. The cross-product in the equation and dependence on velocity has consequences for the numerical modelling of particle motion and is of crucial importance to the present study, which will be elaborated in Chapter 4.

Combined with Newton's second law $\mathbf{F} = m\mathbf{a}$, the Newton-Lorentz force determines the motion of particles purely under electromagnetic forces.

For the charged particles in a plasma, the only forces acting upon them is gravity and the Newton-Lorentz force (2.25). The forcing term $\mathbf{F}_\alpha$ in the Boltzmann/Vlasov equation for a given particle species in a plasma then becomes

$$\mathbf{F}_\alpha = m_\alpha \mathbf{g} + q_\alpha(\mathbf{E} + \mathbf{w} \times \mathbf{B}), \tag{2.26}$$

The Boltzmann/Vlasov equation and Maxwell's laws are now mutually coupled and form a closed set of equations for the time evolution of the electromagnetic fields $\mathbf{E}$, $\mathbf{B}$ as well as the charged particles within them (by way of their distribution function

f).

Deciding on either the Boltzmann or Vlasov equation to model a given problem depends on the collision mechanics of the plasma. Coulomb collisions, when two or more particles interact in collision-like fashion due to the electric Coulomb force between them, dominate over neutral collisions in high-temperature plasmas. In fact, they dominate in plasmas that are ionised by even a few percent, with neutral collisions requiring ionisation levels below 10^{-3} % to dominate. When Coulomb collisions dominate, Vlasov [90] recognised that the plasma can be considered collisionless in the sense of binary physical collisions. The argument was predicated upon two types of experimental results, namely that of natural vibrations and anomalous scattering of electrons in plasma, neither of which can be explained by binary collisions. More fundamentally, binary collisions do not apply to Coulomb interacting particles, as the Coulomb force diverges as the distance between charges goes to zero. On the basis of these observations, the Vlasov equation becomes the best description of the dynamics of the plasma.

2.1.5 Summary

The governing equations for kinetic modelling of a plasma under the influence of gravity are then, in order, Boltzmann's equation or the Vlasov equation, using the Lorentz force (with gravity), coupled with Maxwell's equations:

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{w} \cdot \nabla_r f_\alpha(\mathbf{r}, \mathbf{w}, t) + \frac{\mathbf{F}_\alpha}{m_\alpha} \cdot \nabla_w f_\alpha(\mathbf{r}, \mathbf{w}, t) = \left(\frac{\partial f_\alpha}{\partial t} \right)_{coll}, \tag{2.27a}$$

or

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{w} \cdot \nabla_x f_\alpha(\mathbf{x}, \mathbf{w}, t) + \frac{\mathbf{F}_\alpha}{m_\alpha} \cdot \nabla_w f_\alpha(\mathbf{x}, \mathbf{w}, t) = 0, \tag{2.27b}$$

with

$$\mathbf{F}_\alpha = m_\alpha \mathbf{g} + q_\alpha(\mathbf{E} + \mathbf{w} \times \mathbf{B}), \tag{2.27c}$$

and

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon_0}\left(\frac{1}{\mu_0}\nabla \times \mathbf{B} - \mathbf{j}\right), \tag{2.27d}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \tag{2.27e}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{2.27f}$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}, \tag{2.27g}$$

where

$$\rho_\alpha(\mathbf{r}, t) = n_\alpha(\mathbf{r}, t) q_\alpha, \tag{2.27h}$$

$$\mathbf{j}_\alpha(\mathbf{r}, t) = \rho_\alpha(\mathbf{r}, t)\mathbf{v}_\alpha(\mathbf{r}, t), \tag{2.27i}$$

recalling that $n_\alpha(\mathbf{r}, t) = \int f_\alpha(\mathbf{r}, \mathbf{w}, t)\, d\mathbf{w}$ and $\mathbf{v}_\alpha(\mathbf{r}, t) = n_\alpha(\mathbf{r}, t)^{-1} \int \mathbf{w} f_\alpha(\mathbf{r}, \mathbf{w}, t)\, d\mathbf{w}$,

In this project, a particle method for hot diffuse plasmas is studied and given the previous notes on collisions, the collision-less Vlasov equation is the most relevant to the work. In fact, the PIC method can be seen as a way of solving the Maxwell-Vlasov system of equations above, as discussed in section (2.3). The specific research conducted for this project centres on the numerical integration of the Newton-Lorentz force (2.25), the time discretisation and advancement of the equation and whether improvements on thereof can be successfully implemented within the greater PIC method.

2.2 Fluid Description of Plasma

The Maxwell-Boltzmann equations described above accurately model the full range of plasmas from the microscopic regime to astrophysical length and time-scales. Even at scales so small that the assumption of a smooth density distribution function breaks down, the individual particle orbits and generated fields are then simply described by the Newton-Lorentz force and Maxwell's equations, which are also included almost explicitly with the model.

However, even with the reduction from $6N$ degrees of freedom present in the Coulomb force equation to merely 6 plus time in the Vlasov-Maxwell equations, the system of equations remains too detailed and complicated to solve for anything but the simplest cases. Generally, for non-microscopic plasmas the quantities of interest are the macroscopic ones known from fluids; most prominently temperature, density, pressure and large scale flow velocities. The situation is much the same as with the initial picture containing collections of charged particles: Unnecessary detail provided at small scales at the cost of complexity in modelling at the larger, desired scales. Describing the plasma as a fluid would alleviate these issues and contains the added advantage that fluid dynamics is an established field with a large volume of work and ideas to apply to modelling complications.

To model plasma as a fluid, a series of simplifications and closures of the arising systems of equations must be made. Fundamentally, the description revolves around moments of the Boltzmann equation taken over velocity space, summing the contributions of particles at all velocities. The relation between particle number density and macroscopic quantities can then be applied to yield expressions of a larger scale that do not involve particle values.

2.2.1 Multi-Fluid Theory

Taking the zeroth moment of the Boltzmann Equation "BE" (2.27a) over velocity space ($\int \text{BE}\ d\mathbf{w}$) and using the expressions for macroscopic quantities previously derived leads to the conservation equations for species α. For elastic binary collisions, the number of particles is conserved and thus particle number density is also conserved. Taking the zeroth moment over the collision operator simply yields zero and taking the moment of BE yields the conservation equation

$$\frac{\partial n_\alpha}{\partial t} + \nabla(n_\alpha \mathbf{v}_\alpha) = 0 \tag{2.28}$$

for the species α. As particle density is conserved, mass density is conserved as well yielding an identical equation for $\rho_{m,\alpha}$. Note here that taking the moment over velocity space eliminates velocity as an independent variable $\mathbf{w}$ and expressions are now functions of position $\mathbf{r}$ and time t only.

More important to note is the assumption of elastic collisions only. Inelastic collisions between particles result in energy exchanges in other forms than momentum exchange, e.g. through ionisation or recombination, which would add or remove particles respectively.

The first moment ($m_\alpha \int BE\ \mathbf{w}_\alpha d\mathbf{w}$) leads to conservation of momentum for species α

$$\rho_\alpha \frac{d_\alpha \mathbf{v}_\alpha}{dt} + \nabla \cdot \mathbf{P} - \rho_\alpha \mathbf{g} - q_\alpha \mathbf{E} - \mathbf{j}_\alpha \times \mathbf{B} = \boldsymbol{\mu}_\alpha, \tag{2.29}$$

where $\mathbf{P}$ is the *pressure tensor* and $\boldsymbol{\mu}_\alpha$ is the average transfer of momentum to species α from other species due to collisions.

This is the multi-fluid plasma equivalent of Navier-Stokes equation, coupled to Maxwell's equations by the electromagnetic force term $q_\alpha \mathbf{E} - \mathbf{j}_\alpha \times \mathbf{B}$. In the derivation, terms involving the velocity fluctuations are combined with the definitions of pressure and temperature of an ideal gas to establish the pressure tensor $\mathbf{P}$, com-

monly split into isotropic and anisotropic parts $\mathbf{P} = p_\alpha \mathbf{I} + \mathbf{\Pi}_\alpha$. The isotropic part $p_\alpha \mathbf{I}$ represents pressure in the common sense, a scalar value p_α for any point in the plasma so long as the velocity distribution is close to the Maxwellian distribution. The anisotropic stress tensor $\mathbf{\Pi}_\alpha$ then contains all deviations from the Maxwellian distribution, in which case temperature and pressure for a point depends on direction. An example would be a plasma with different temperatures along and across the magnetic field lines, respectively. In the case where the Maxwellian velocity distribution holds in all parts of the plasma, $\mathbf{\Pi}_\alpha$ is zero.

Finally, conservation of energy is established by taking the second moment

$$m_\alpha \int BE \; \frac{\mathbf{w}_\alpha^2}{2} \, d\mathbf{w}, \tag{2.30}$$

and recognising the third order term involving velocity fluctuations that appears $\rho_\alpha < u_\alpha^2 \mathbf{u}_\alpha >_\alpha$ as the flux of kinetic energy $\mathbf{\Phi}_\alpha$ due to the velocity fluctuations, i.e. the heat flux due to random motions. Including a term for heat loss/gain due to collisions with other particles H_α, a rather long expression for the conservation of energy of species α is achieved. By using the equation for change in internal energy, the internal energy term in the total energy equation can be rewritten, and energy conservation for species α becomes

$$\frac{d_\alpha p_\alpha}{dt} - \gamma_\alpha \frac{p_\alpha}{\rho_\alpha} \frac{d_\alpha \rho_\alpha}{dt} = (\gamma_\alpha - 1) \Bigg[- \nabla \cdot \mathbf{\Phi}_\alpha - \sum_{l=1}^{3} \sum_{k=1}^{3} \Pi_{\alpha,kl} \frac{\partial v_l}{\partial r_k} + H_\alpha \Bigg], \tag{2.31}$$

where γ_α is a factor depending on the degrees of freedom available to the particles, with $\gamma_\alpha = \frac{5}{3}$ for the standard 3 translational degrees of freedom.

In summary, three main equations of interest arise from the first round of fluid simplifications, at the loss of the information from the distribution function f in velocity space. As there are more unknown variables than equations, the set of equations is not yet closed. To reduce the model further, more physical assumptions are needed.

The major assumption made in the above derivation, which limits the applicability of the multi-fluid model, was that of purely elastic collisions. In practice, this excludes the model from describing plasmas with any significant increase or decrease in ionisation without additional terms to model such phenomena. Laboratory plasmas are often created by gradually ionising a neutral gas through heating, which limits the applicability of the model in this regard.

2.2.2 Two-Fluid Theory

Modelling the plasma as just two interpenetrating fluids is straightforward using the above equations. Two-fluid specialisation of the multi-fluid model is of special interest to plasmas since, while the specific species may change from problem to problem, two types of species are always present in the form of negative and positive charges. The two intersecting fluids are likely to be, for instance, an electron fluid and an ion fluid.

Two sets of equations are obtained by replacing the subscript and quantities relating to species α with those of electrons e and ions i. The two sets of mass, momentum and energy conservation equations are then coupled by the electromagnetic force term and Maxwell's equations, but several quantities (e.g. the collisional effects $\boldsymbol{\mu}_\alpha$, H_α) remain unknown. The set of equations can be closed with some further assumptions, namely $\boldsymbol{\mu} = H = \boldsymbol{\Phi} = \boldsymbol{\Pi} = 0$, which in effect stipulates a collisionless plasma, unmagnetised and free from velocity fluctuations.

While such an extensive simplification would generally be inapplicable, this non-collisional model has been useful for studying plasma oscillations, leading to the equation for *Langmuir waves*. These are foundational in plasma dynamics, occurring in every plasma containing both positive and negative charges. The oscillations can be seen as an instability inherent to the free electron fluid: Displace a part of the electron fluid from the intersecting ion fluid and the electrons will be pulled

back by the electrostatic potential, overshoot due to inertia and repeat the motion, creating a sustained oscillation.

2.2.3 Single-Fluid Theory

To establish a model of plasma using only one conducting fluid representing several species rather than a component fluid for each particle species, the conservation equations of species α are summed as contributions to the bulk plasma conservation equations. The added terms of each equation must then be considered individually, to determine if the equivalent bulk plasma quantity is simply the sum of the species contributions. Number, charge and mass density for example, are all simply summed so that charge density of the bulk plasma is $Q = \sum_\alpha Q_\alpha$. On the other hand, the pressure, pressure tensor, internal energy and heat flux all require summing over an integral, e.g. $P_{ij} = \sum_\alpha \int f_\alpha u_i u_j \mathbf{w}$. Here it is important to note that the fluctuating velocity $\mathbf{u}$ is now defined with respect to the average velocity $\mathbf{v}$ of the bulk plasma $\mathbf{u} = \mathbf{w} - \mathbf{v}$. With definitions for the various macroscopic quantities in a single-fluid plasma, the conservation of mass, charge, momentum and internal energy can be written

$$\frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{v} = 0, \tag{2.32}$$

$$\frac{\partial Q}{\partial t} + \nabla \cdot \mathbf{j} = 0, \tag{2.33}$$

$$\rho \frac{d\mathbf{v}}{dt} + \nabla \cdot \mathbf{P} - \rho \mathbf{g} - Q\mathbf{E} - \mathbf{j} \times \mathbf{B} = 0, \tag{2.34}$$

$$\frac{dp}{dt} - \frac{\gamma p}{\rho} \frac{d\rho}{dt} = (\gamma - 1) \Bigg[- \nabla \cdot \mathbf{\Phi} + \mathbf{E}^* \cdot \mathbf{j}^* - \sum_{l=1}^{3} \sum_{k=1}^{3} \Pi_{kl} \frac{\partial v_l}{\partial r_k} \Bigg], \tag{2.35}$$

respectively, where $\mathbf{E}^* = \mathbf{E} + \mathbf{v} \times \mathbf{B}$ and $\mathbf{j}^* = \mathbf{j} - Q\mathbf{v}$ is the electric field and current from the reference frame of the moving plasma. No new assumptions need to be made in the derivation of these equations, but as all species contributions have been summed up, the electric field and current terms above are also aggregate terms,

which makes things more complicated.
This set of equations is still not closed, with no expressions for the anisotropic stress tensor and heat flux. Additionally, an equation is required to evolve current density $\mathbf{j}$ in time, linking it to the plasma variables ρ, $\mathbf{P}$, $\mathbf{v}$.

2.2.4 Magnetohydrodynamics (MHD)

An expression linking current and the electromagnetic field to the plasma variables is obtained by taking the first charge moment of the Boltzmann equation, resulting in the *generalised Ohm's law*. The conservation equations and Ohm's law together can then be simplified and rewritten to describe the plasma behaviour only in terms of the average velocity, pressure and magnetic field. Several important physical assumptions are made during these derivations, limiting the use of the resulting equations.

Firstly, the equations are specialised to deal with electrons and ions where the mass of the former is considered negligible compared to the latter. Quasi-neutrality is also assumed, which means the number density of the two species must be equal. In classic MHD, the anisotropic tensor is removed by assuming that the ion and electron temperatures are equal and isotropic, i.e. the velocity distribution of each is identical and follows the Maxwellian. Letting the anisotropic stress tensor go to zero also means that the classic MHD equations are inviscid however, which is why no viscosity term appears below.

In what is known as resistive MHD, the internal energy equation is simplified by assuming identical ion and electron temperatures and neglecting heat loss/gain. In this manner, the single-fluid system of equations is closed, resulting in the conser-

vation laws for mass, momentum and internal energy

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}), \tag{2.36}$$

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla p + \rho \mathbf{g} + \frac{1}{\mu_0}(\nabla \times \mathbf{B}) \times \mathbf{B}, \tag{2.37}$$

$$\frac{dp}{dt} = -\gamma p \nabla \cdot \mathbf{v} + (\gamma - 1)\frac{|\nabla \times \mathbf{B}|^2}{\mu_0^2 \sigma}, \tag{2.38}$$

where σ is the conductivity and μ_0 is the vacuum permeability. Also arising is the *induction equation*, which takes the form

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B}, \tag{2.39}$$

for resistive MHD. Here $\nabla \cdot \mathbf{B} = 0$ and $\eta = \frac{1}{\mu\sigma}$ is the plasma diffusivity. The ideal MHD equations are obtained by letting $\sigma \to \infty$, i.e. assuming an infinitely conducting plasma.

In practice, the assumptions made stops the resistive MHD model from describing plasmas with a different composition than the standard electron/ion mix and plasmas with directionally dependent temperatures and pressures. An example where the first assumption may not hold are pure ion or electron plasmas. While these do not meet the quasi-neutrality criteria of plasmas, such collections still behave as plasmas when trapped in strong electromagnetic fields. The assumption of isotropy is the more limiting, barring use for any plasmas not in thermal equilibrium, which includes all cold plasmas.

Cold plasmas, where the velocity distributions of one or more species are not isotropic and have little to no spread in velocity space, are of particular interest in the current study. The case studied in Chapter 6, Section 1 is a premiere example of a cold plasma which gradually thermalises but does not become universally isotropic.

2.3 Particle Modelling of Plasma

Returning to the kinetic description of plasmas from section 2.1, if one wishes to avoid the MHD description of plasma with the simplifications and loss of detail it brings, the Vlasov-Maxwell system must be solved as is. A few test problems with analytical solutions exist [26], but generally, numerical methods are required to use the equations in practice. Having to discretise a phase-space as opposed to the regular configuration-space is in no way an insurmountable challenge, and groups such as Godfrey and Vay [25] have pursued the straightforward approach of solving (2.15) with Maxwell's equations on a finite difference Eulerian grid. The advantage of these "Vlasov codes" are an inherent smoothness in the solution, which is represented by the distribution function - a real number denoting the probability of finding a particle at given point in the simulated phase-space. Unfortunately, the distribution is likely to be non-zero only in localised areas of phase-space, i.e. much of the simulated domain will be empty or almost empty of particles and add nothing to the simulation but added computational cost. The problem is demonstrated clearly on the left plot of Figure 2.1, which shows a plasma distribution in $1d1v$ phase-space, that is, 1 dimension each in position and velocity space. Vlasov codes are already limited by the computational cost, with the cost scaling N^6 for the cell number N in each dimension as opposed to N^3 for models that only consider configuration-space. The efficiency of such schemes thus becomes low for full $3d3v$ simulations, where much of the considered phase-space, all resolved by a costly N^6 grid, becomes effectively empty. [59]

To overcome these difficulties in solving the Vlasov-Maxwell system, one might use a particle-in-cell "PIC" scheme. The original development and use of particle schemes came not from distribution functions and Vlasov, but from the more fundamental level of actual particle dynamics. In the late 1950s and early 1960s, pioneering work by Buneman [14] and Dawson [18] showed that, when performed appropriately, smaller systems of particles can serve as an accurate model of real

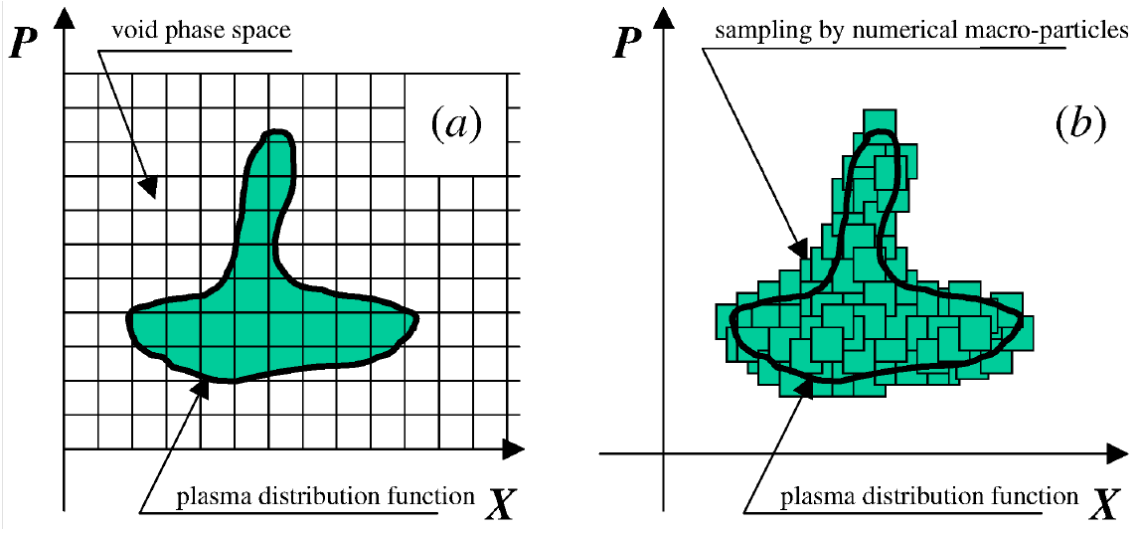

Figure 2.1: Vlasov (a) vs. PIC (b) method Pukhov [59]. Finite difference discretisation of all phase-space required for (a), but only the plasma distribution of interest is marked by Finite Phase Fluid Elements (FPFE) for (b).

plasmas containing a much larger number of particles. In this manner, plasmas containing too many particles for direct modelling of each particle via the Maxwell and Lorentz equations can still be simulated accurately by modelling the dynamics of 'macro-particles' or 'super-particles', each representing thousands of physical particles moving with the same velocity. To begin to justify such an a simplification, one can consider the criteria related to plasma behaviour, and attempt to fulfil said criteria while changing the assumed composition of the simulated plasma under study.

For instance, recall the number of particles in the Debye sphere $N_D \approx n\lambda_D^3$ for a plasma under study, in the case of the Earth's ionosphere $N_D \approx 10^4$. If the plasma of interest has $N_D \gg 1$ as is the case here and the length scale of interest $L \gg \lambda_D$, simulating all physical particles is indeed prohibitively expensive. However, the fundamental dynamics of a plasma does not necessarily require an enormous number of representative particles, and a given system might be representable using fewer particles. Consider the energy balance between electrostatic attraction and thermal motion, which is given by the ratio of thermal to local potential energy for the

particles

$$\frac{\text{Thermal kinetic energy KE}}{\text{Local potential energy PE}} \gg 1. \tag{2.40}$$

This ratio is equal to N_D for a real plasma, but the fundamental dynamics only require that $N_D \gg 1$ which could be satisfied at values as low as $N_D = 10$, an order of magnitude difference. It is not unreasonable to think that the macroscopic plasma dynamics of a given configuration can be captured accurately by fewer, bigger particles, so long as the fundamental dynamical criteria are met. Indeed, the original mid-century studies proved this to be the case.

The drawbacks of a macro-particle approximation such as this include higher noise levels, higher collision rates and of course, a loss of detail from fully resolved particles. The increased noise stems from the fact that 1 macro-particle represents the charge and mass of thousands to millions of real particles but inhabit only a single point in phase-space. In the real picture, a cloud of particles is unlikely to display the same velocity with position naturally distributed through the physical space inhabited by the cloud. Aggregating particle positions and velocities of a cloud, which represent a smooth element of the distribution function in phase-space, into a single number leads to concentration of charge appearing as noise. Despite the initially less formal mathematical grounding of particle methods compared to MHD, they became and have remained popular since conception [8].

While the origins of particle methods of plasma simulation appear to be wholly separate from the kinetic theory outlined in section (2.1), the approach can be linked back to the fundamental Vlasov equation and its discretisation in phase-space. As previously mentioned, each macro-particle represents a finite cloud of ions or electrons, where the position and velocity of the macro-particle then represent the centre-of-mass and mean velocity of the clouds respectively. In this manner, the macro-particles can be seen as volumes of incompressible 'phase-fluid'. Using this idea, the problems of the Vlasov codes outlined above can be circumvented by using a type of finite element method instead of finite differences. As seen on the right

plot of Figure 2.1, so-called Finite Phase-Fluid Elements (FPFE) can be used to sample the distribution function:

$$f(\mathbf{x},\mathbf{v}) = \sum_n W_n S(\mathbf{x}-\mathbf{x}_n, \mathbf{v}-\mathbf{v}_n), \tag{2.41}$$

where W_n is the weighting and S is the shape function of the element in phase-space, and the centre of the element is given by position and momentum $\mathbf{x}_n, \mathbf{v}_n$ respectively. A simple shape function to use is the 6D hypercube:

$$S(\mathbf{x},\mathbf{v}) = \begin{cases} 1, & \text{if} \quad |x_j| < \frac{\Delta x_j}{2} \quad \text{and} \quad |v_j| < \frac{\Delta v_j}{2} \quad \text{for all} \quad j = x,y,z \\ 0, & \text{otherwise} \end{cases} \tag{2.42}$$

where Δx_j, Δv_j are the element sizes along configuration and momentum space axis j respectively.

To show how the Vlasov equation is formally discretised using these phase-fluid elements, consider an infinitesimal volume of phase-space $d\mathbf{V} = d\mathbf{x}\, d\mathbf{v}$. In this view, $f(\mathbf{x},\mathbf{v},t)\, d\mathbf{V}$ gives the probability that a charged particle inhabits the volume at time t. The particle will then move to a new volume $d\mathbf{V}' = d\mathbf{x}'\, d\mathbf{v}'$ at time t' following the equations of motion

$$\mathbf{x}' = \mathbf{x} + \int_t^{t'} \mathbf{v}\, dt, \qquad \mathbf{v}' = \mathbf{v} + \int_t^{t'} \frac{\mathbf{F}}{m}\, dt. \tag{2.43}$$

From this reasoning, the distribution in general is

$$f(\mathbf{x}',\mathbf{v}',t') = f(\mathbf{x},\mathbf{v},t). \tag{2.44}$$

These two equations together are a re-statement of (2.16), and the equation of

motion (2.43) is the characteristic equation thereof. The physical interpretation of this is that the density distribution f is conserved along particle trajectories.

To avoid mapping f for every infinitesimal volume, take a number N_s of sample points $\{\mathbf{x}_i, \mathbf{v}_i\}$ with $i = 1, ..., N_s$. Each point represents one phase-fluid element i containing $N = \int_i f d\mathbf{x}\, d\mathbf{v}$ plasma particles. The trajectories of the sample points/phase-fluid elements are then

$$\frac{d\mathbf{x_i}}{dt} = \mathbf{v}_i, \qquad \frac{d\mathbf{v}_i}{dt} = \frac{1}{m_n}(\mathbf{F} + \mathbf{F}_{st}), \tag{2.45}$$

where $\mathbf{F}$ is the force and $\mathbf{F}_{st}$ is the effective force from collisions and $m_s = Nm$ is the mass of the element, representing the combined mass of the constituent plasma particles. The appeal of interpreting the elements as macro-particles becomes clear from (2.45). Using this model, keeping track of f becomes unnecessary as all pertinent plasma quantities can be retrieved approximately from the macro-particle description alone. Consider for example taking the velocity moments of f needed to establish the bulk plasma velocity field in configuration-space. These can be approximated by

$$\int \mathbf{v}_i f\, d\mathbf{v} \simeq \frac{N}{\lambda} \int_{\mathbf{x}-\frac{\lambda}{2}}^{\mathbf{x}+\frac{\lambda}{2}} d\mathbf{x}' \int \mathbf{v}_i \tilde{f}\, d\mathbf{v}, \tag{2.46}$$

$$= \frac{N}{\lambda} \sum_i \mathbf{v}_i, \tag{2.47}$$

where $\tilde{f} = \sum_{i=1}^{N_s} \delta(\mathbf{x}-\mathbf{x}_i)\delta(\mathbf{v}-\mathbf{v}_i)$ is the distribution of phase-fluid elements/macro-particles and $\sum_i$ is the sum over all macro-particles in the volume λ around the point $\mathbf{x}$ where the bulk velocity is being evaluated.

These are simply the particle equations of motion, i.e. the phase-fluid elements behave as particles and the finite element method described is equivalent to PIC.

In this manner, maintaining a 6D Eulerian mesh is avoided and the phase-fluid elements, which shall be referred to as macro-particles or simply particles henceforth, sample the distribution function efficiently with no waste. The electromagnetic fields still need advancing via Ampere (2.17) and Faraday's (2.18) laws however, done via finite difference discretisation in configuration-space. Maintaining a static mesh has thus not been completely avoided, but it remains only three dimensional rather than six. For the electrostatic case, it may be tempting to forego a mesh entirely and model the electric field via the Coulomb interactions (2.24). PIC still maintains the advantage in this case however, as the Coulomb interactions represent an N^2 problem, where the electric field acting on each of the N particles depends on the position of every other particle.

To avoid the N^2 interactions and to couple the particle motion with the static mesh fields, interpolation is applied to establish charge/current on the mesh points from that of the surrounding particles within the cell volume and inverse interpolation to determine the electric and magnetic fields on the particle from the nearest mesh values. Between the interpolations, Maxwell's equations can be solved to determine grid electromagnetic field values from the interpolated charge and current. From this separation comes the name *particle-in-cell*, with coupling and concurrent solution of the mean-fields on a grid and the particles/distribution function in continuous space. The history and details of PIC will be more fully covered in Chapter 3.

PIC schemes could be considered Lagrangian Vlasov schemes, but are in reality closer to the fundamental physics than these: As the scale of the plasma is reduced, one macro-particle will eventually represent one physical particle or close thereto.

In the simulation of the above-mentioned cold plasmas, where the particle velocity distribution of one or more species does not conform to the Maxwellian, the efficiency of PIC as compared to Vlasov codes in representing the phase-space is particularly evident. The particles of a cold plasma exhibit only limited, potentially infinitesimal, regions of velocity space. The plasma in these cases thus inhabits only a 3D

hypersurface out of the 6D phase-space, which may be said to be degenerate. Said surface can be sampled effectively using only a small number of macro-particles and remains degenerate even as the system evolves, so long as no heating is present. This last point is a major reason why it is crucial that PIC codes be energy conserving [59].

Chapter 3

The Particle-In-Cell Method

The previous chapter detailed the fundamental mathematical models used in the study of plasma dynamics. The Particle-in-Cell (PIC) numerical method, which is popularly used to simulate the phase-space evolution of a plasma, was briefly introduced. In this chapter PIC is explored in-depth, starting with the history and fundamentals common to all variants. Two distinct types of PIC, ESPIC and EMPIC, are then expanded upon separately. In electrostatic PIC (ESPIC) the magnetic field is constant or zero, while in electromagnetic PIC (EMPIC) the full Maxwell-Vlasov system is solved which includes time-varying magnetic fields.

Finally, while the current study applies PIC in its most basic forms, more recent additions and improvements on the basic PIC scheme have been given a brief treatment in Section 3.4 with a focus on higher order methods in subsection 3.5. These sections should provide perspective on some modern capabilities of the method and what capabilities the current research is meant to contribute. The following historical overview and demonstration of the scheme fundamentals follow those of Verboncoeur [83] and Langdon [45].

3.1 Fundamentals

3.1.1 History

PIC began with basic physics models by Buneman [14] and Dawson [18] in the late 1950s, which demonstrated the power of computational plasma physics. For instance, Dawson [19] demonstrated Landau damping of electrostatic waves in plasma using PIC, before it had been observed experimentally. Originally, kinetic simulations of plasma were conducted in 1D using particles that were essentially 'sheets', with electric force computed via Coulomb's Law [45]. In these models, the electric force between particles was calculated using direct application of Coulomb's law [28], which requires N^2 calculations for the interactions between N particles. The community sought to improve on this over the following decade, leading to the first particle-mesh schemes, in which Poisson's equation is solved on a grid and the calculated electric forces interpolated to the particles, which were tracked in continuous space. These schemes were called Cloud-in-Cell (CIC) or Particle-in-Cell (PIC) [83].

The difference between CIC and PIC is mainly one of charge deposition. In CIC every particle is treated as a rectangular charge with the dimensions of a cell in the field grid. The rectangular area can then be split by proximity to the nearby nodes and each element assigned to the nearest-grid-point. For PIC, the charges are considered point particles and the charge is scattered to the nearest-grid-points based on simple area weighting [45].

The first 2D PIC codes were electrostatic and arose when it became possible to solve Poisson's equation (3.3) on a grid quickly and accurately (ca. 1964). The 2D PIC codes were eventually extended to cover the full Ampere-Faraday-Maxwell electromagnetic equations including relativistic regimes. Much of the theory and proper codification of the PIC scheme was worked out during the 1970s [45], resulting in the seminal texts by Birdsall and Langdon [8] as well as Hockney and Eastwood

[37].

Boundary interactions became a focus during the 1980s. Second-order accurate models of both plasma and electrical circuits connected to the plasma were achieved by the end of the decade [85] and combined plasma/circuit modelling was eventually extended to 2D [80]. In the same time-frame, simple cross-section approaches for charged-neutral particle collisions were produced by Boswell and Morey [12]. In the 1990s, collisional models were improved with differential cross-sections [81], which allow for the inclusion of detailed probabilities of collision between particles depending on their species and current state.

More recently, the code-bases for PIC have been updated to reflect the prevalence of object oriented programming [84], and models taking advantage of modern massively parallel processor computers (MPP) have been developed by Blahovec et al. [9], Liewer and Decyk [48], Di Martino et al. [20]. By 2005, on the order $10^6 - 10^8$ particles could be used for 2D and 3D simulations on serial computers, with $10^8 - 10^{10}$ possible on MPP platforms [83].

3.1.2 Algorithm Overview

Recall that PIC works by simulating the movement of charged particles via forces arising from the electromagnetic (EM) fields. The fields are defined on a spatial grid and advanced in time separately from the particles by gathering the charge and current at the grid-points. Doing so allows Ampere and Faraday's laws to be solved on the grid, which establishes the global electric and magnetic fields arising from the distribution and movements of the charged particles. These grid defined values can then be interpolated back to the particle positions to solve the equations of motion and ´push' the particles in time. Doing so sidesteps the N^2 particle interactions required to determine the electric field via Coulomb's law, requiring only the interactions between each particle and nearby field nodes. The number of

these interactions is only proportional to N [45]. The split also provides a simpler framework to couple the charge and current densities associated with the particle movements to Maxwell's laws and advancement of the fields, allowing the full set of Maxwell equations to be solved rather than just Coulomb's law.

The interaction of particle pairs in close proximity compared to the grid spacing is not captured by PIC without additional corrective schemes, as the charge and current is dispersed from particles in close proximity to the surrounding grid. The success of PIC methods in simulating weakly coupled (hot, diffuse) plasmas despite this shortcoming is possible due to the dominance of long distance over short distance electromagnetic forces. As explained in Chapter 2, this observation leads to the assumption that collisional effects can be neglected for hot, diffuse plasmas. This in turn yields the Vlasov equation, wherein the short-range forces are negligible and the movement of the charged particles is wholly determined by the configuration of the global fields. In PIC, the global fields are captured accurately on the grid, making the scheme appropriate for modelling plasmas governed by the Vlasov equation.

Discarding the short-range forces actually becomes an advantage in PIC, as these forces would lead to strong interactions between the macro-particles. Recall that higher noise is expected compared to continuous function Vlasov solvers since each macro-particle represents many real particles by one value of position and velocity. This noise is directly counteracted by the dispersal of particle quantities to the surrounding grid which works to make the particles less point-like in practice [37].

In spite of these serendipities, the scheme does have both numerical and computational weaknesses. PIC must resolve scales closer to the physical particle scale than fluid models, meaning large ranges of scale in time and space can require immense computational resources. As for the statistical model employed, for N particles, the numerical fluctuations converge as $N^{-1/2}$, i.e. to achieve meaningful statistics a large number of particles must still be used.

Visualising one iteration in time of some general PIC algorithm (Figure 3.1) aptly demonstrates the flow of the method. In the classic PIC schemes as described by Birdsall and Langdon [8], note that the iteration begins with the particle velocity update. This requires the chosen particle integration scheme to also start with the velocity update, which some do not, see Chapter 4 and 5.

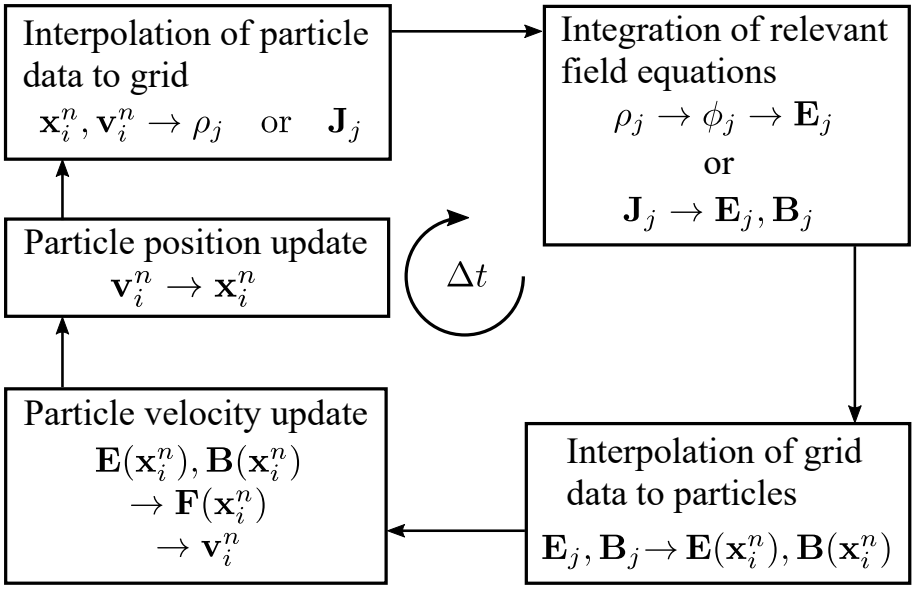

Figure 3.1: General PIC algorithm iteration, adapted from Verboncoeur [83].

In short, the method progresses by determining the electric $\mathbf{E}$ and magnetic $\mathbf{B}$ fields at discrete positions in space via the charge density ρ and current $\mathbf{J}$. These are in turn established by advancing the positions and velocities of the macro-particles. The resulting distribution of charges gives rise to the charge density and the local movement of charges through cells determines the currents.

Recall that the focus of this project lies in testing higher order numerical solutions of the particle equations of motion via Boris-SDC, improving the accuracy of advancing the particle positions and velocities in time. The improved particle integration should in turn improve the field solution, by providing more accurate time evolution of the charge and current distributions. The effect of this improvement on the PIC scheme as a whole can then be determined by testing and comparing the performance of PIC using both classic Boris and Boris-SDC. The limitations of

the lower order interpolation and field solutions on the combined PIC/Boris-SDC scheme are of particular interest.

3.2 Electrostatic Particle-in-Cell (ESPIC)

For electrostatic problems, where the magnetic field is constant in time or zero, using the full set of Maxwell's laws becomes unnecessary. The formal definition of electrostatic is that the electric field is irrotational $\nabla \times \mathbf{E} = 0$, which by Faraday's law means the magnetic field must be constant if it exists $\frac{\partial \mathbf{B}}{\partial t} = 0$. An irrotational electric field can be represented by a scalar electric potential, defined by

$$\mathbf{E}(\mathbf{x}, t) = \nabla \phi(\mathbf{x}, t). \tag{3.1}$$

Assuming a constant magnetic field means the magnetic field contributions produced by the charged particle currents are considered negligible. The system can now be simplified to consider only the electric field as defined by the instantaneous particle positions. The relationship between particle positions and electric field comes from Gauss' law

$$\nabla \cdot \mathbf{E}(\mathbf{x}, t) = \frac{\rho(\mathbf{x}, t)}{\epsilon}, \tag{3.2}$$

which relates the latter to charge density. The fundamental equation in ESPIC is the Poisson equation, which comes from combining (3.2) and (3.1):

$$\nabla \cdot \nabla \phi(\mathbf{x}, t) = \nabla^2 \phi(\mathbf{x}, t) = \frac{\rho(\mathbf{x}, t)}{\epsilon}, \tag{3.3}$$

where the electric field can be retrieved by differentiating the electric potential using (3.1).

The Poisson equation in ESPIC is commonly solved via a second-order accurate

central difference scheme

$$\frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{\Delta x^2} + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta y^2} + \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{\Delta z^2} = \frac{\rho_{i,j,k}}{\epsilon}, \tag{3.4}$$

where i, j, k subscripts indicate the mesh node index in the x, y, z directions respectively. The resulting linear system can be solved using a suitable algorithm such as LU or Cholesky decomposition, direct inversion of the square matrix or an iterative method.

The charge density depends on the particle positions which in turn is evolved using the velocity, both of which depend on the electric field value. For the system to be closed, the charge density on the grid must be coupled to the particle charges and the field values on the particles to those on the grid. The coupling is performed via interpolation between particles and the surrounding grid nodes. This takes the general form

$$q_j^n = \sum_{i=1}^{N_q} W_j(\mathbf{x}_i^n) q_i, \tag{3.5}$$

for the deposition of charge from particles to the grid. Here q_j^n is the charge on grid node j at time $t = n$, N_q is the total number of individual particles i, q_i is the particle charge and $W(\mathbf{x}_i^n)$ is a weighting function dependent the position $\mathbf{x}_i^n$ of particle i at time $t = n$. In practice, not all particle contributions are summed for each cell, as most particles will be too far away from a node (weighted zero) and not contribute to the nodal charge. To efficiently scatter charge, the particles can be looped over once, with affected nodes selected based on the particle shape (weight) for each particle.

With accumulated charge established on the grid nodes, the cell charge densities corresponding to particle positions $\mathbf{x}_i^n$ are calculated via

$$\rho_j^n = \frac{q_j^n}{\Delta V}, \tag{3.6}$$

where ΔV is the cell volume.

After solving Poisson's equation on the grid, the mesh of charge density values can be differentiated using simple central differences of the form

$$E_j = \frac{\rho_{j-1} - \rho_{j+1}}{2\Delta x} \tag{3.7}$$

to yield the electric field components, here in 1D yielding the scalar component along the x-axis for clarity. These are defined on the grid nodes collocated with the electric charge and potential. At the mesh boundaries, forward and back differences must be used instead. The nodal electric field components can then be interpolated back to the particle positions $\mathbf{x}_i^n$ using

$$\mathbf{E}_i^n = \sum_{j=1}^{N} W_j(\mathbf{x}_i^n)\mathbf{E}_j^n, \tag{3.8}$$

where $\mathbf{E}_i^n$ is the electric field at the particle positions, $\mathbf{E}_j^n$ is the electric field at the grid nodes, and N is the number of grid nodes.

As a visual example, Figure 3.2 shows interpolation of charge from particles to grid nodes for 2D ESPIC diagramatically, using linear area weighting to only the surrounding nodes. This operation of charge deposition is sometimes referred to as charge "scattering".

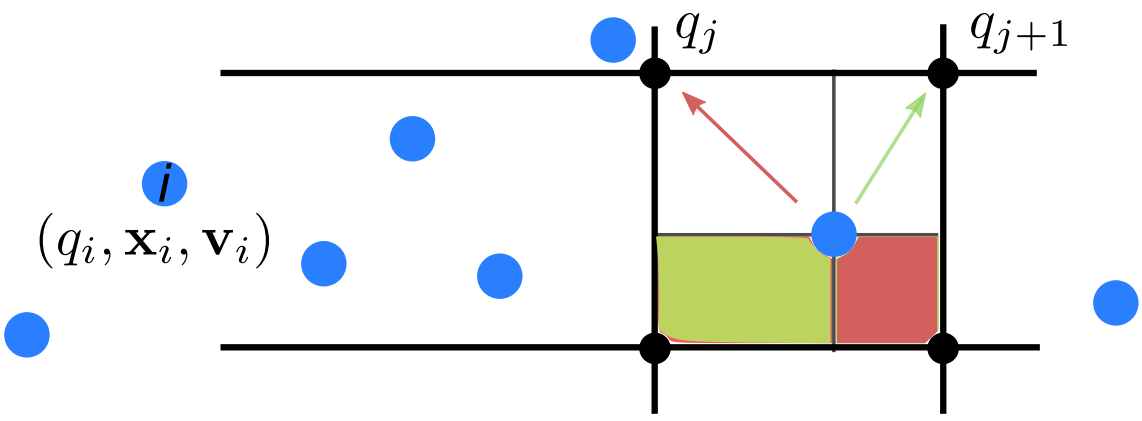

Figure 3.2: Charge scattering from particle to Eulerian mesh.

Similarly, Figure 3.3 shows the interpolation of the fields from the grid nodes back to the particles for the same example. This operation is sometimes referred to as field "gathering".

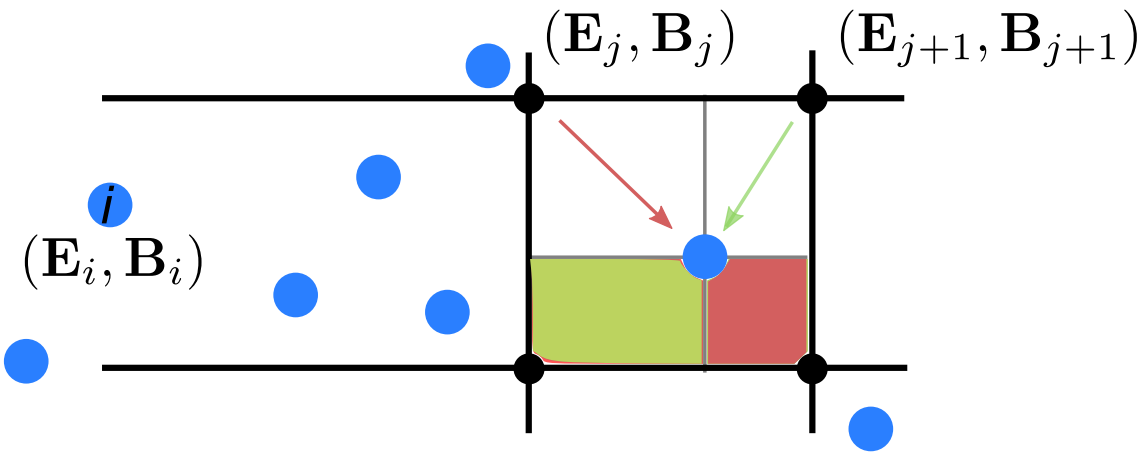

Figure 3.3: Field gathering from Eulerian mesh to particle.

The bilinear area weighting and its 3D equivalent, trilinear volume weighting, are 1st order accurate weight functions commonly used in ESPIC as well as EMPIC when performing the initial charge scatter and field gather to satisfy Gauss' law.

The linear weight function determining the impact of a particle of position $\mathbf{x}_i$ on a grid node of position $\mathbf{X}_j$ can be calculated individually for any scalar component of the two position vectors as

$$W(x) = \begin{cases} 1 - \frac{|x-X|}{\Delta X} & |x - X| < \Delta X \\ 0 & |x - X| > \Delta X \end{cases} \tag{3.9}$$

where x and X are the scalar positions of the particle and grid node along a given axis and ΔX is the cell width along the same axis. As shown on the diagrams above, this weighting scheme interpolates values between particles and the closest surrounding grid nodes. This linear interpolation scheme for scattering and gathering is thus equivalent to the particle having a finite shape, in this case the same shape as the cell: A line, square or cube of size ΔX. Other weighting schemes than

linear can be used, such as quadratic, cubic splines or further high order splines [68]. The simplest weighting would be to simply assign the charge to the nearest-grid-point, which was extensively used in the early days of PIC, but suffers from increased noise and has the highest error.

Once the fields have been interpolated from the grid to the particles, the particle positions and velocities can be advanced in time using an appropriate numerical method, as detailed in Chapter 4.

In summary, the electrostatic PIC procedure for modelling a plasma consists of coupled evolution of the electric field and macro-particles using Poisson's equation to establish the electric field for a given particle distribution and advancing in time by integrating the particle equations of motion using this electric field, coupling the particles and grid via interpolation. Figure 3.4 demonstrates the solution loop for ESPIC as used in this study.

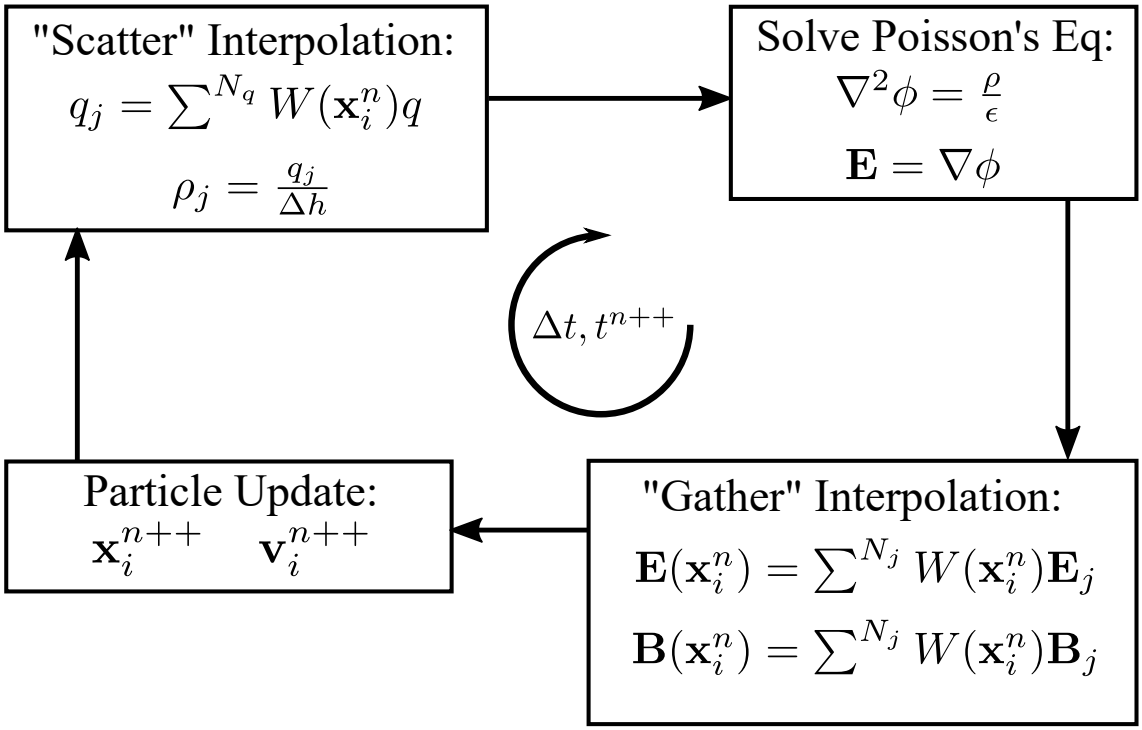

Figure 3.4: Solution loop of the ESPIC scheme.

Note that while **B** in the ESPIC cycle is static and might not require interpolation, an empricially determined field without an appropriate analytical expression would

need to be defined on the grid and interpolated to the particles, hence the inclusion of magnetic "gather" interpolation in the above figure.

3.3 Electromagnetic Particle-in-Cell (EMPIC)

To solve the full set of Maxwell equations, Finite-Difference Time-Domain discretisation is generally employed on a so-called Yee mesh (Figure 3.5), where the electric field is known at integer time-steps and the magnetic known at the half time-steps. Starting from initial conditions, which may require solving Poisson's equation for the electric field and tabled results for an imposed magnetic field, (2.19) and (2.20) should also remain satisfied in time.

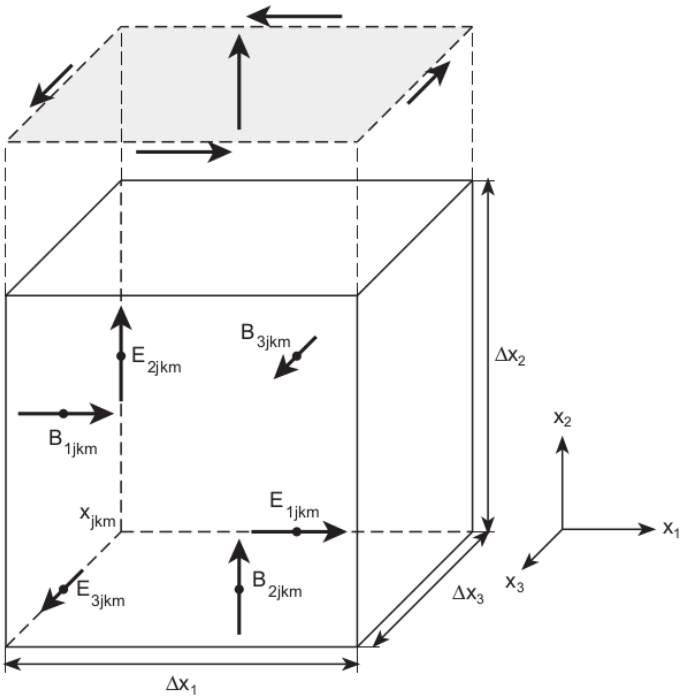

Figure 3.5: Structure of a Yee mesh cell [83]. Note the half cell width staggering of positions at which EM field values are defined, this enables cheap second order accurate spatial finite differences. The numbers in the subscript of each field vector simply refers to the direction shown (1 along x, 2 along y, etc.).

The fields can then be advanced in time using Ampere's and Faraday's laws in general finite-difference format

$$\Delta[\mathbf{E}]_t = \frac{1}{\epsilon}\left(\frac{1}{\mu}\Delta[\mathbf{B}]_{\mathbf{x}} - \mathbf{J}\right), \tag{3.10a}$$

$$\Delta[\mathbf{B}]_t = -\Delta[\mathbf{E}]_{\mathbf{x}}, \tag{3.10b}$$

where the $\Delta[]_t$ and $\Delta[]_{\mathbf{x}}$ operators denote finite difference discretisation of the subject in time and space respectively. Space discretisation in this case means discretisation of the curl terms.

In EMPIC, the field solver needs only the electric current and previous field values to advance in time, as per Ampere's law (2.17). This means the interpolation needed in EMPIC are that of Yee mesh fields to the particles and determining the grid current from particle movement. If the algorithm used to deposit current from the particles to the grid does not make use of the direct charge interpolation from the previous section, or is charge conserving in some other way, explicit reinforcement of (2.20) is required. Failing to do this means there is no guarantee that divergence of the electric field is zero at any point of the domain, and charge can accumulate or dissipate as the simulation progresses.

The general idea of current deposition algorithms in EMPIC which utilise charge weighting similar to ESPIC is as follows. For a 2D Yee mesh with nodes in x, y denoted by the subscripts j, k, the particle motion schematised in Figure (3.6) is observed. Using the definition of charge weight (3.9) from the electrostatic case, the change of charge weighting in time between two mesh nodes is defined as $\Delta W = W^{t+\Delta t} - W^t$. The current along the top edge of the left cell in the x-axis is then

$$I_{1,j+1/2,k+1} = \sum_i \frac{q_i}{\Delta t}\Delta W_1 \overline{W}_2, \tag{3.11}$$

where i again denotes the particles, $\overline{W}$ is the average charge weighting

$$\overline{W} = \frac{W^{t+\Delta t} + W^{t}}{2}, \tag{3.12}$$

and the subscripted weight terms ΔW_1, $\overline{W}_2$ refer to, respectively, change in charge weight along the horizontal and average charge weight along the vertical during the time-step.

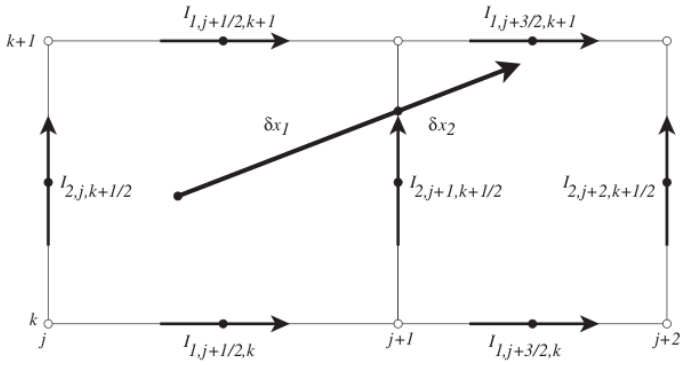

Figure 3.6: Current deposition from moving particle to Eulerian mesh [83].

The currents along the other edges are determined from similar formulae and these are applied for each cell traversed, i.e. overall motion of each particle is broken into the segments falling within each cell. So long as a charge conserving algorithm is used, the divergence relations from the Maxwell equations are automatically satisfied at all time-steps, provided that they are satisfied at the start of the first time-step [83].

When used for motion within a single cell, the method is equivalent to that of Morse and Nielson [53]. For multi-cell movement it becomes equivalent to that of Eastwood [21] as well as Villasenor and Buneman [88]. An alternative scheme based instead on tracking the flux of charge in straight lines was formulated by Esirkepov [24]. Refer to these studies for more information on current interpolation.

3.4 Advances in Kinetic Modelling and PIC

Most of the PIC fundamentals described in the previous section were worked out prior to the mid-1980s, summarised by Birdsall and Langdon [8], and remain relevant to modern PIC. Development has not ceased, and a number of extensions to PIC, enhancing the capabilities and efficiency of the scheme have been proposed. The forthcoming discussion on the more recent developments from 1985 onwards follows that of Verboncoeur [83].

3.4.1 Generalised Weighting

The weighting methods described previously were developed for uniform meshes and use a fixed volume when calculating the charge density. This causes systematic error when used in conjunction with non-uniform volumes, which include curvilinear co-ordinates and variable mesh spacing. The general weighting scheme by Verboncoeur [86] fixes irregular volumes for PIC. Differential volume elements are weighted to nodes to establish the density n_j

$$n_j = \frac{\int_{\mathbf{r}} f(r) W_j(\mathbf{r}dr)}{\int_{\mathbf{r}} W_j(\mathbf{r}dV)}, \tag{3.13}$$

using the same interpolation function W_j as for particle charge. In this manner, the charge density can be computed exactly for cells with non-uniform volumes, provided the number of particles N_q greatly exceeds the number of cells N.

3.4.2 Collisions

The first inclusion of neutral-charge collisions to PIC by Boswell and Morey [12] worked via simple step probabilities. Vahedi and Surendra [81] expanded the method to include energy dependent cross-sections, where the probability for particle i to

collide with a particle of species g is given by

$$P_i = 1 - \exp[-n_g(\mathbf{x})\sigma_T(\varepsilon_i)v_i\Delta t], \tag{3.14}$$

where n_g is the density of the target particles and v_i is the incident speed of the particle. The parameter $\sigma_T(\varepsilon_i)$ is the total cross-section, essentially the sum of individual probabilities of the various possible collision processes between i and g, which is a function of the kinetic energy of the particle ε_i. This is the Monte-Carlo collision (MCC) scheme for PIC.

The MCC method is computationally expensive, as it requires a call to a random number generator for every particle at every time-step. The computation cost per particle of MCC can significantly exceed that of the particle push.

An alternative is the *null collision method*, where a maximum collision frequency is defined

$$\nu_{max} = \max_{\mathbf{x}}(n_g(\mathbf{x}))\max_{\varepsilon}(\sigma_T(\varepsilon)v), \tag{3.15}$$

and the probability for each particle to undergo collision in a given time-step is established cheaply without requiring energy or position information via

$$P_T = 1 - \exp(-\nu_{max}\Delta t). \tag{3.16}$$

The method is not as computationally costly as MCC but it is also less accurate, as the collision parameters of each individual particle are neglected.

Different dynamics are possible for each type of collision. For electron-neutral collisions the three possible events are elastic scattering, excitation and ionisation (latter two being inelastic type events).

A disadvantage inherent to PIC regarding collisions is that the Coulomb force goes to zero as particle separation decreases and particles enter the same cell, as opposed to increasing quadratically as it should. Recall that the Debye length is the distance at which the electrostatic influence (Coulomb force) of a particle is shielded by opposite charges in the plasma. To reduce the error when Coulomb interactions are significant, a mesh resolving scales smaller than the Debye length is thus required.

An alternate scheme by Takizuka and Abe [75] was devised to fix the problem by adding collision modelling to each cell, randomly selecting and pairing particles within each cell and calculating the collision angles. This scheme is computationally expensive as the time-step must be much smaller than the time between collisions. A cheaper method was later proposed by Nanbu [56], wherein only large angle collisions are explicitly calculated by summation of the small angle collisions, allowing independent time-step selection.

3.4.3 Boundary Interaction

The focus on boundary conditions in the 1980s arose from the workings of the plasma devices used in experimentation. While handling PIC boundaries is simple for a fixed imposition of either Neumann or Dirichlet conditions on the outer grid, real plasma devices are often driven by an external electrical circuit, the current in which can and will respond to the behaviour of the plasma. A model for external circuits with reactive impedances based on a simple RLC (resistance-inductance-capacitance) schematic was developed by Verboncoeur et al. [85].

Extending PIC modelling to plasma-surface interactions have spurred the creation of algorithms for secondary emission, when charged particles strike bounding surfaces and create more particles in the process. A general secondary emission model has been described by Gopinath et al. [27].

3.4.4 Noise Filtering

Statistical noise was mentioned as a weakness of the PIC scheme, with a scaling of $N^{-1/2}$ for N particles, i.e. the noise decreases slowly with the number of particles. Fluctuations in the problem quantities which are identified as 'noise' can have adverse numerical effects. The fluctuation levels in PIC simulations can be orders of magnitude higher than the physical systems represented, which can cause numerical heating, in which the particles accelerate due to the numerical methods used rather than any real physical process [83].

To reduce noise, filters can be applied. In planar (2D) models, spatial filters like Fourier space filters have seen much use in smoothing out shorter wave-lengths [8]. These models are not effective in curvilinear coordinates and an alternative for cylindrical coordinates has been formulated by Verboncoeur [87]. Noise is generally affected little by a single pass of these filters, but making dozens or hundreds of passes would be computationally expensive. An alternative is modification of the models to apply a great number of passes in a single operation.

3.4.5 Massively Parallel Processing

As with other computation topics, high performance computing and parallelisation have been a great focus for kinetic plasma simulation schemes. Massive parallelisation allows for more expensive simulation than is currently feasible to perform on even the best individual computers. A number of schemes, like those by Liewer and Decyk [48] or Di Martino et al. [20] have been developed and implemented for particle simulations.

The particle 'push' of the PIC method is especially suitable for parallelisation, as the calculation for each particle is brief and mostly only requires information about the particle itself and localised values (nearby field node values).

Parallel computation of the field mesh, etc. require a scheme similar to many other applications of MPP computing, with interior and neighbouring mesh parts, the latter of which must communicate solution values. Figure 3.7 shows the flow of a parallelised PIC scheme by Luginsland et al. [50].

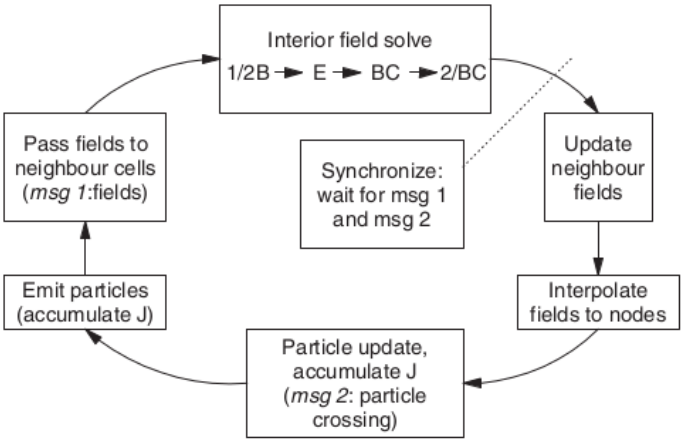

Figure 3.7: Parallel PIC code flow [50].

In the scheme *msg 1* contains the field node values required for solution of Maxwell's equation in neighbouring parts. The other message *msg 2* contains the particles which cross into other parts.

Another consideration to potentially reduce required computational resources greatly is vectorisation of the required calculations. Memory access operations can be greatly reduced by storing particle data in singular large arrays as opposed to individual arrays for each entity/variable. The formation of data structures and variable encapsulation should be informed by these considerations when writing PIC code [83].

Significant areas of improvement for PIC models as predicted by Verboncoeur [83] include improved models for boundary-field and boundary-particle interactions, multi-physics modelling combining different scales and plasma simulation approaches, chemistry models for ionisation and other reactions and the continued increase in complexity and scale of desired simulation. The author noted that a full-scale sim-

ulation of a fusion reactor would require a multi-petaflop computer, which has since been built [13].

3.5 Higher order methods

The topic of higher order methods in all parts of PIC, encompassing integration and interpolation of both particles and field, has received more attention in recent years. As high order particle integration is reviewed in detail in Chapter 4, the discussion here is limited to high order interpolation and field solvers for traditional ESPIC/EMPIC as well as any complete overhauls that extend or modify all components of PIC in unison.

The interpolation functions used to couple particles and grid in PIC directly determine the error of the charge/current distribution used in the solution of the Maxwell system, but also affect the noise of the PIC scheme before any filtering is applied. The noise associated with the macro-particles depend on the chosen shape of the particles, which relates to the order of the interpolation function used in interpolating charge/field between particles and mesh. As an example, for the first order interpolation function (3.9) described in Section 3.2, the shape of the particle charge is triangular w.r.t. the mesh. This is because the charge on a given mesh node increases linearly to a peak depending on the proximity of the particle providing the charge. A further benefit is distributing the charge of a particle over several cells rather than just one as in the linear case. This in turn has a smoothing effect as seen in Figure 3.8 on the charge and current distribution used to solve (3.10). Abe et al. [1] was one of the first studies to propose higher order interpolation functions to improve PIC accuracy. The study further demonstrated theoretically that high order interpolation splines would alleviate the increasing error from self-heating. A more recent application of high order splines was demonstrated by Shalaby et al. [68] for their 1D PIC code "SHARP-1D", which uses up to fifth order interpolation.

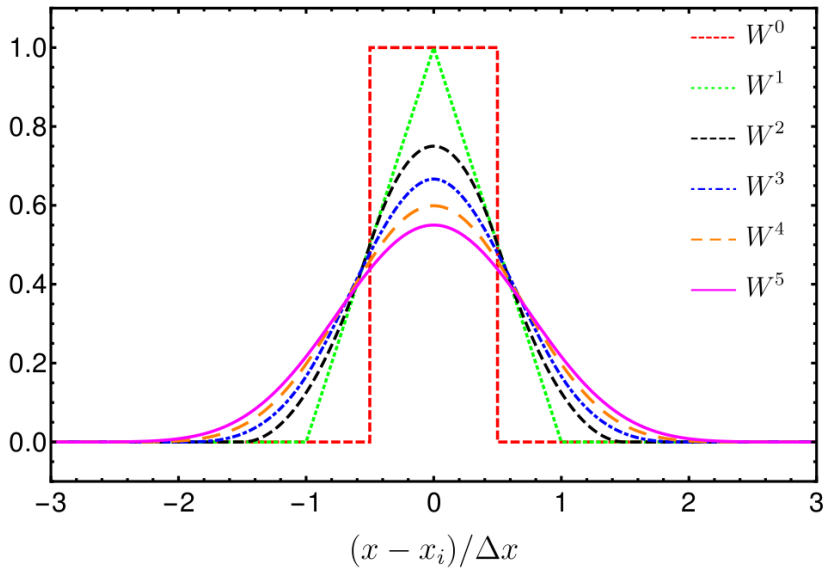

Figure 3.8: 1D particle shapes for a particle at $x = 0$ interpolated to PIC cells with nodes at the integers using interpolation of increasing order [68]. The range of weighting functions shown here include previously mentioned nearest-grid-point W^0 and linear W^1 functions and up, ending at the fifth order function W^5.

The code was tested for Landau damping as well as both non-relativistic and relativistic two-stream instabilities, making the work a spatial counterpart to the current study. SHARP-1D was restricted to second order convergence in time by the use of second order leapfrog scheme to advance the particles, but nonetheless exhibited better energy-conservation and lower numerical heating when fifth order interpolation was used. Shalaby et al. [68] also makes an important note on the meaning of numerical resolution in PIC schemes. The use of macro-particles coupled to a mesh for plasma simulation leads to three resolutions and associated numerical error: Spatial resolution (cell size Δx), momentum resolution (particle count per cell $N_q/\Delta x$) and spectral resolution (periods per domain length $2\pi/L$). The authors omit temporal resolution based on the assumption that it should be tied to the spatial resolution via stability conditions. For the current study of high order time integration of the macro-particle motion, this assumption is discarded in favour of investigating and quantifying the trade-off in performance between spatial and temporal resolution. The tests of the project code in Chapters 5-6 investigate the error and computational efficiency of ESPIC at a range of temporal and spatial

resolutions, while keeping momentum and spectral resolution fixed at the highest values feasible. Another recent PIC code utilizing high order particle interpolation (up to W^3) was introduced earlier by Haugbølle et al. [33] which also included up to sixth order Yee-mesh finite difference solution of the fields. The high order field solution came at the cost of an implicit update for magnetic field, which can cause issues for parallelisation wherein parallel domains must wait for the slowest iterative solver to converge. The authors claimed the in-built solver would minimise impact on parallel simulations and mentioned tests hinting at an increase in computational efficiency from high order particle integration, but no testing was presented in the study and no follow-up studies were found.

Explicit high order finite difference solutions for the fields have been proposed by Benedetti et al. [6] and combined with both high order interpolation and fourth order Runge-Kutta particle integration in the EMPIC code ALaDyn. The combined high order scheme was found to allow a coarser grid and larger time-step size while maintaining the accuracy of a second order scheme, reducing computational effort. Furthermore, the study demonstrated the use of high order interpolation in reducing the particle noise, but at a high computational cost. The fourth order Runge-Kutta does not conserve energy and Londrillo et al. [49] implemented fourth and second order leapfrog integrators within ALaDyn to reduce energy error. The ALaDyn studies did not isolate the impact of the particle time integrator but do provide a clear demonstration of performance improvements in accuracy and computational efficiency by the use of high order methods. The authors also noted explicitly how the time-step convergence of the scheme, and by extension the overall computational effciency, was limited by the second-order nature of the time integration.

Jacobs and Hesthaven [39] proposed a PIC scheme based on nodal discontinuous Galerkin (DG) methods which was capable of both higher order field solution and using unstructured grids, which the finite difference schemes reviewed so far are not. The DG scheme was found to exhibit improved conservation of energy and more

accurately captured high frequency phenomena. Further improvements were made to the DG scheme in [40], where high order convergence in time is demonstrated using an implicit-explicit variant of fourth order Runge-Kutta. The implicit-explicit integrator demonstrated superior computational efficiency compared to standard fourth order Runge-Kutta when hyperbolic divergence cleaning was used to maintain charge-conservation of the implicit-explicit scheme. High order interpolation for DG PIC schemes was investigated in detail by Stindl et al. [71] who confirmed the benefits of smooth particle shapes extend to DG and unstructured grids. The study demonstrates that coarser grids and fewer macro-particles can be used without compromising accuracy, in the presence of high-order interpolation. Note that this computational saving must be offset against the needs of the problem under study, as kinetic effects decrease when particles are accumulated into smooth but large clouds.

Entirely new PIC formulations derived directly from the Vlasov-Maxwell system using techniques from Hamiltonian mechanics were reviewed by Xiao et al. [95]. The review summarises work on the topic of PIC schemes constructed by structure-preserving analysis starting from the fundamental equations for collections of charged particles. These new PIC schemes do not follow the traditional flow seen in Figure 3.1, but are explicit and possess strong numerical conservation of physical invariants such as energy and charge. The schemes also allow high order extensions of each PIC component, i.e. interpolation, field and particle integration. The review highlights the contributions to the field along with the main formulations of Hamiltonian PIC schemes by Xiao et al. [94], Kraus et al. [42] and He et al. [35]. As the focus of the current study is the impact of a self-contained high-order particle integrator on classic PIC schemes, neither Hamiltonian mechanics nor this new type of PIC will be treated in detail. A brief introduction to important concepts from Hamiltonian mechanics and a referral to the appropriate literature is given in the beginning of the next chapter, where more Hamiltonian methods are reviewed.

In summary, the use of high order methods in PIC is well-established and has been demonstrated to yield more accurate results and the potential for increased computational efficiency. These result indicate that high order particle integration is beneficial, but none of the studies investigated the potential performance impact of the integrator in isolation. Comparisons were made to other high order integrators, or to the energy stability of classic low order schemes, but no extensive testing on the performance impact of mesh vs. particle solver was found. The current study will focus on this gap in the literature.

Chapter 4

Particle Time Integration

With the overarching theory of plasma modelling and PIC established in the previous chapters, attention can now turn to the focus of the current study: Numerical solution of particle propagation. Recall that in PIC simulations, the overall plasma distribution is evolved in time by evolving a collection of computational macro-particles using the theory of charged particle dynamics. As such, numerical integration of the particle equations of motion (particle pushing), form a crucial part of any PIC scheme. The simulated movement of charged particles in complicated fields is an interesting topic in its own right which need not go hand-in-hand with PIC, which is essentially just a way of computing the fields arising from the particle motion itself. For this study however, the main interest is in the impact on the PIC scheme of the particle integration schemes used therein. Specifically, determining the effect on computational performance of using the novel particle pusher, Boris-SDC, with both ESPIC and EMPIC.

To put the forthcoming novel research into context, the following sections will thus cover the fundamentals of numerical integration of charged particle motion as relevant to PIC. This will include details on the Boris algorithm, which is the most enduringly popular particle pusher for PIC, as well as other recent integrators meant to compete with Boris in some aspect. Several of these newer integrators have their

roots in Hamiltonian mechanics and a brief introduction to important Hamiltonian concepts is included in the section on fundamentals.

With the proper foundation laid by the discussion of classic Boris, the derivations of both Boris-SDC formulations are demonstrated in Section 4.3, setting the stage for the implementation and testing of the combined PIC/Boris-SDC scheme in Chapters 5 and 6.

4.1 Fundamentals

For the classic PIC method, wherein the particle-field coupling is achieved via interpolation of the fields to particles and charge current to field grid, advancing the fields and particles in time can be handled separately. Recall that the charged particle motion in PIC is governed by the Newton-Lorentz equation

$$m\frac{d}{dt}\mathbf{v} = \mathbf{F}_L(\mathbf{x}, \mathbf{v}) = q(\mathbf{E}(\mathbf{x}) + \mathbf{v} \times \mathbf{B}(\mathbf{x})), \tag{4.1}$$

here in non-relativistic form; the PIC coupling of particle and field means the electric $\mathbf{E}$ and magnetic fields $\mathbf{B}$ are constant at the mesh nodes during the particle time-step. The field values at any stage of the particle time integration are thus found purely as a function of particle position, by interpolating from the nodes.

The motion of the particles can be modelled by rewriting (4.1) as the ODE system

$$\frac{d}{dt}\mathbf{v} = \mathbf{f}(\mathbf{x}, \mathbf{v}) = \frac{q}{m}(\mathbf{E}\big(\mathbf{x}\big) + \mathbf{v} \times \mathbf{B}(\mathbf{x})\big), \tag{4.2a}$$

$$\frac{d}{dt}\mathbf{x} = \mathbf{v}, \tag{4.2b}$$

which lends itself to a variety of numerical time integration methods, which shall be the focus of this section.

As computational power increases and longer simulations become possible, long-term accuracy becomes ever more important as a key performance metric. For a given numerical integration scheme, the natural expectation might be that all important quantities of the physical system are not conserved due to the truncation error in the discretisation and must increase or decrease systematically over the course of a simulation. The discovery of schemes which conserve one or more invariants of the physical system indefinitely is remarkable, with energy-conservation in particular being an important property in numerical integration of (4.2). If energy is not conserved or at least bounded by the integrator, the addition or subtraction of energy at each time-step, however small, will accumulate as error. Without correction, this energy error eventually causes the simulation to diverge completely from the physical problem it attempts to simulate.

Long term conservation of important quantities in the context of particle mechanics goes hand-in-hand with the topic of Hamiltonian systems. While several relevant numerical integration methods have been developed based on classic non-Hamiltonian theory (including Boris and Boris-SDC), many modern methods are derived from Hamiltonian mechanics. The focus on such methods underline the importance of upholding conservation laws for particle integrators. The topic involves solving directly in terms of conserved quantities such as energy and is therefore also useful in analysing and establishing adherence to conservation laws for otherwise unrelated numerical methods.

For a detailed introduction to Hamiltonian mechanics, refer to Leimkuhler and Reich [46]. Following here is a brief definition of important concepts from Hamiltonian mechanics relevant to performance of numerical integrators, i.e. desirable properties that an integrator can possess. Conservation of energy is a fundamental property of physical systems, and thus it is crucial that a numerical scheme also conserves energy or that energy error is at least bounded. A given physical system may have other conserved properties, which are referred to as first integrals in the

analysis of Hamiltonian systems. Examples of other first integrals in the study of particle motion are linear and rotational momentum. Geometric integrators refer to integrators which conserve geometric properties of a differential equation, such as phase-space area or volume. As an example for the phase-space of 1D particle motion, the area of phase-space defined by a square with lengths A_x and B_v along the position/velocity axes may move and deform, but will always maintain an area equal to $A_x B_v$. Conserving phase space volume is typically associated with good energy conservation [46], and so there is a natural limit to the contribution to numerical heating by such a scheme. Symplectic integrators are a subset of geometric integrators and symplecticity is a stronger property than volume conservation, which is a necessary but insufficient condition for symplecticity. A given first integral is not guaranteed to be conserved due to symplecticity, but a given symplectic integrator can ensure conservation of more invariant properties than phase-space volume and energy. Finally, time reversal symmetry is another necessary but insufficient condition for symplecticity and also a desirable integrator property even if symplecticity is not achieved. A symmetric integrator ensures that reversing the momentum history of a particle causes it to travel its original trajectory in reverse. This guarantees energy-conservation in many cases, and symmetry is therefore highly desirable if a given integrator cannot be made fully symplectic [46].

4.2 The Boris Algorithm

4.2.1 Leapfrog Integration using Boris

The Boris algorithm was proposed as a way to resolve an implicit velocity dependence that arises in the leapfrog scheme for numerical integration of (4.2). The algorithm and integrator have been used in combination to the point that leapfrog integration using the Boris algorithm is referred to as the Boris integrator, but the two are distinct methods. The common derivation of the classic Boris integrator (leapfrog

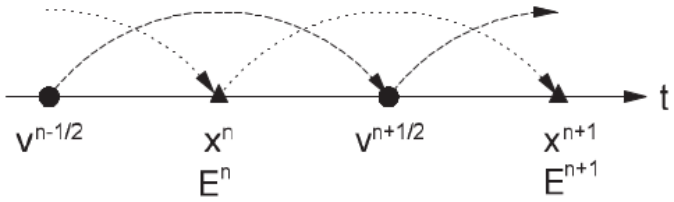

Figure 4.1: Leapfrog scheme [83], the charge density and electric field would be defined at the integer time-steps, with current density and magnetic field defined at the half time-steps.

integration with Boris) is presented in this section following Birdsall and Langdon [8] to demonstrate the traditional use of the algorithm proposed by Boris [11].

The general leapfrog scheme for charged particle motion works by defining the particle positions on the integer time-steps (Figure 4.1). Using staggering of the velocity updates by $\Delta t/2$, a second order accurate central difference scheme is achieved using minimal storage and fewer operations than an equivalent scheme wherein integer time-step values of both velocity and position are tracked. The leapfrog integrator is thus an example of a time-staggered integrator, wherein position and velocity are not known at the same points in time. Contrast this staggering to time-synchronised integration, where the position and velocity are both known at each integer time-step.

When the force does not depend on velocity, the scheme is explicit and similar in form to forward Euler, but with the added benefit of second order accuracy due to the staggered time-stepping. When applied to the Newton-Lorentz system the scheme becomes implicit as

$$\frac{\mathbf{v}_{n+1/2} - \mathbf{v}_{n-1/2}}{\Delta t} = \frac{q}{m}\left(\mathbf{E}_n + \frac{\mathbf{v}_{n+1/2} + \mathbf{v}_{n-1/2}}{2} \times \mathbf{B}_n\right), \tag{4.3a}$$

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} = \mathbf{v}_{n+1/2} \tag{4.3b}$$

where n is a given time-step with time t_n and $n \pm 1/2$ denotes $t_n \pm \frac{\Delta t}{2}$. The velocity at the integer time-step is defined implicitly as the average of the velocity at the next and previous half steps [83].

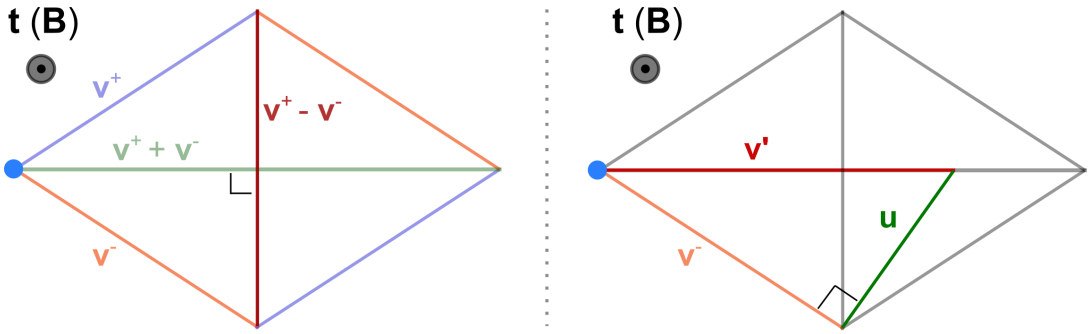

Figure 4.2: Geometric view of the Boris algorithm, adapted from Birdsall and Langdon [8].

To avoid an integration involving the inversion of the above cross-product, Boris [11] devised an efficient integration scheme based on the observation that the magnetic field only ever serves to rotate the particle trajectory. The magnitude of particle velocity will thus remain the same under the influence of $\mathbf{B}$ (Figure 4.2). The scheme is equivalent to solving for the implicit dependence by inverting the 3-by-3 matrix arising from the cross-product of the magnetic field vector with the velocity average.

The derivation of Boris' scheme starts from the position and velocity updates as defined by the leapfrog scheme:

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + \frac{\Delta t q}{m}\left[\mathbf{E}(\mathbf{x}_n) + \frac{1}{2}(\mathbf{v}_{n+1/2} + \mathbf{v}_{n-1/2}) \times \mathbf{B}(\mathbf{x}_n)\right], \tag{4.4}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{v}_{n+1/2}, \tag{4.5}$$

here in the non-relativistic case.

Knowing that the magnetic field serves only to *rotate* the trajectory of a particle, the forcing over the time-step is split into three: Initial electric acceleration, full magnetic rotation and final electric acceleration

$$\mathbf{v}^- = \mathbf{v}_{n-1/2} + \frac{\Delta t q}{2m}\mathbf{E}_n \qquad \text{and} \qquad \mathbf{v}^+ = \mathbf{v}_{n+1/2} - \frac{\Delta t q}{2m}\mathbf{E}_n. \tag{4.6}$$

Physically, $\mathbf{v}^-$ is the velocity after half the electric forcing, pre-rotation and $\mathbf{v}^+$ is the velocity post-rotation before the second half of the electric forcing.

Adding and subtracting these definitions result in the following terms

$$\mathbf{v}^+ - \mathbf{v}^- - \frac{\Delta tq}{2m}\mathbf{E}_n = \mathbf{v}_{n+1/2} - \mathbf{v}_{n-1/2} \qquad \text{and} \qquad \mathbf{v}^+ + \mathbf{v}^- = \mathbf{v}_{n+1/2} + \mathbf{v}_{n-1/2}, \tag{4.7}$$

substituting these expressions into (4.4) result in the electric field terms cancelling out.

The equation (4.4) can now be written in terms of the pre- and post-rotation vectors only

$$\mathbf{v}^+ - \mathbf{v}^- = (\mathbf{v}^+ + \mathbf{v}^-) \times \frac{\Delta tq}{2m}\mathbf{B}_n. \tag{4.8}$$

Recognising from

$$\mathbf{a} \times \mathbf{b} = |\mathbf{a}||\mathbf{b}| \sin(\theta_{ab})\hat{\mathbf{n}}, \tag{4.9}$$

that the orthogonality of any two vectors result in their cross-product reducing to a simple magnitude multiplication, a vector ratio can be established from the similar triangles in Figure 4.2:

$$\frac{|\mathbf{u}|}{|\mathbf{v}^-|} = \frac{|\mathbf{v}^+ - \mathbf{v}^-|}{|\mathbf{v}^+ + \mathbf{v}^-|} = \frac{\Delta tq}{2m}|\mathbf{B}_n| = |\mathbf{t}|. \tag{4.10}$$

The vector $\mathbf{v}'$ about which $\mathbf{v}^-$ is rotated into $\mathbf{v}^+$ can now be expressed in terms of the pre-rotation velocity $\mathbf{v}^-$ and magnetic force vector $\mathbf{t} = (\Delta tq\,\mathbf{B}_n)/2m$

$$\mathbf{v}' = \mathbf{v}^- + \mathbf{u} \qquad \text{and} \qquad \mathbf{u} = \mathbf{v}^- \times \mathbf{t} \qquad \text{so} \qquad \mathbf{v}' = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{t}. \tag{4.11}$$

Refer to Figure 4.2 for the geometry of how vectors $\mathbf{v}^+$, $\mathbf{v}^-$, $\mathbf{u}$ and $\mathbf{v}'$ relate.

Since $\mathbf{v}'$, $\mathbf{v}^+ - \mathbf{v}^-$ and $\mathbf{t}$ are all mutually orthogonal, it follows that one of the

vectors can be expressed as some fraction α of the cross-product of the other two

$$\mathbf{v}^+ - \mathbf{v}^- = \alpha \mathbf{v}' \times \mathbf{t}. \tag{4.12}$$

Applying the Pythagorean theorem on the formula for $\mathbf{v}'$ then yields

$$|\mathbf{v}'|^2 = |\mathbf{v}^-|^2 + |\mathbf{v}^- \times \mathbf{t}|^2 = |\mathbf{v}^-|^2 + |\mathbf{v}^-|^2|\mathbf{t}|^2, \tag{4.13}$$

which can be rearranged to give

$$|\mathbf{v}^-|^2 = \frac{|\mathbf{v}'|^2}{1 + |\mathbf{t}|^2}. \tag{4.14}$$

Using similar triangles, new ratios of length can be established. Starting from

$$\frac{|\mathbf{v}^+ + \mathbf{v}^-|}{2|\mathbf{v}^-|} = \frac{|\mathbf{v}^-|}{|\mathbf{v}'|}, \tag{4.15}$$

and rearranging yields

$$\frac{|\mathbf{v}^+ + \mathbf{v}^-|}{2} = \frac{|\mathbf{v}^-|^2}{|\mathbf{v}'|} = \frac{|\mathbf{v}'|}{1 + |\mathbf{t}|^2}. \tag{4.16}$$

However, as $|\mathbf{v}^+ - \mathbf{v}^-|/|\mathbf{v}^+ + \mathbf{v}^-| = |\mathbf{t}|$, the magnitude relations become

$$|\mathbf{v}^+ - \mathbf{v}^-| = |\mathbf{t}||\mathbf{v}^+ + \mathbf{v}^-| = \frac{2|\mathbf{v}'||\mathbf{t}|}{1 + |\mathbf{t}|^2} = \frac{2}{1 + |\mathbf{t}|^2}|\mathbf{v}'||\mathbf{t}|. \tag{4.17}$$

Returning to vectors rather than magnitudes, the formula for the post-rotation vector becomes

$$\mathbf{v}^+ = \mathbf{v}^- + \frac{2}{1 + |\mathbf{t}|^2}\mathbf{v}' \times \mathbf{t}, \tag{4.18}$$

which is explicit and easily computable using known quantities for the particle velocity and magnetic field via $\mathbf{v}' = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{t}$ and $\mathbf{t} = (\Delta t q\, \mathbf{B}_n)/2m$.

The leapfrog Boris integrator has been in use for over 40 years since it was pro-

posed in 1969 and is the de-facto standard algorithm for integrating the Lorentz force to advance particle position and velocity in time. The popularity stems from the long term bounded energy error of the algorithm observed in many applications as noted by Qin et al. [62], who also showed that Boris is not symplectic but does conserve phase-space volume. The conservation properties of Boris do not hold in all cases, as Hairer and Lubich [31] showed that near-conservation of energy is only seen when the magnetic field is constant or electric potential is quadratic. The debate over the conservation properties of Boris illustrates an important question: How important is the stricter property of symplecticity compared to simply volume-preservation? Both help bound the energy error for any arbitrarily long simulation length but do not in general conserve or bound any other invariants of the system. As an example, neither type of algorithm will bound the global phase error, e.g. the gyration radius of a particle circling in a magnetic field will be accurately predicted at any point of the simulation but the simulated particle will slowly drift out of phase with the equivalent 'real' particle.

4.2.2 The Boris Trick

The use of leapfrog integration using Boris as simply "Boris integration" is ubiquitous in PIC, but a different interpretation is required for Boris-SDC. The derivation (4.6)-(4.18) from the previous section is ultimately a distinct algorithm used to solve the implicit velocity dependence of (4.2) and can be applied to any discretisation of the velocity ODE (4.2a), provided it is in the right form. For a discretised implicit equation of the form

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \alpha\mathbf{E} + \beta\frac{\mathbf{v}_n + \mathbf{v}_{n+1}}{2} \times \mathbf{B} + \mathbf{c}, \tag{4.19}$$

where α, β are scalar constants and $\mathbf{E}$, $\mathbf{B}$, $\mathbf{c}$ are given vectors, Boris' trick as presented in Algorithm 1 can be used to calculate $\mathbf{v}_{n+1}$ explicitly. The terms $\alpha\mathbf{E}$

and $\mathbf{c}$ are kept separate here to allow clarity in the arithmetic differences between classic leapfrog and alternative formulations (as classic leapfrog has $\mathbf{c} = 0$). The solution will be second order accurate if the given constants and vectors correspond to a second order scheme. For the leapfrog Boris integrator, the scalar constants are equal to the time-step Δt multiplied by the charge-to-mass ratio q/m and the vector $\mathbf{c}$ does not feature, i.e. it is set to 0.

Algorithm 1: The Boris trick. Adapted and generalised from [79].

input : $\mathbf{v}_n$, α, β, $\mathbf{B}$, $\mathbf{E}$, $\mathbf{c}$
output: $\mathbf{v}_{n+1}$ solving $\mathbf{v}_{n+1} = \mathbf{v}_n + \alpha\mathbf{E} + \beta\frac{\mathbf{v}_{n+1}+\mathbf{v}_n}{2} \times \mathbf{B} + \mathbf{c}$

1.1 $\mathbf{t} = \frac{\beta}{2}\mathbf{B}$
1.2 $\mathbf{s} = 2\mathbf{t}/\left(1 + \mathbf{t} \cdot \mathbf{t}\right)$
1.3 $\mathbf{v}^- = \mathbf{v}_n + \frac{\alpha}{2}\mathbf{E} + \frac{1}{2}\mathbf{c}$
1.4 $\mathbf{v}^* = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{t}$
1.5 $\mathbf{v}^+ = \mathbf{v}^- + \mathbf{v}^* \times \mathbf{s}$
1.6 $\mathbf{v}_{n+1} = \mathbf{v}^+ + \frac{\alpha}{2}\mathbf{E} + \frac{1}{2}\mathbf{c}$

Using Algorithm 1, Boris integrators can be constructed for time-synchronised discretisation schemes (see velocity-Verlet Boris below). Additionally, the algorithm can be applied as is to relativistic formulations of (4.2), see Verboncoeur [83] for the relativistic leapfrog Boris integrator and Section 5.2 of this thesis for relativistic extensions to velocity-Verlet and Boris-SDC.

An example of a another formulation which retains the form (4.19) is the velocity-Verlet integrator used as a basis for Boris-SDC by Winkel et al. [92]. The scheme is second-order accurate like leapfrog but time-synchronised rather than time-staggered, with the final system of update equations taking the form

$$\mathbf{v}^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2}\mathbf{f}(\mathbf{x}^n, \mathbf{v}^n), \tag{4.20}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1/2}, \tag{4.21}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{q}{m}\Delta t \left[\frac{\mathbf{E}^n + \mathbf{E}^{n+1}}{2} + \frac{\mathbf{v}^n + \mathbf{v}^{n+1}}{2} \times \mathbf{B}^n \right]. \tag{4.22}$$

Here $\mathbf{f}$ is the force given by (4.2a) and the velocity update (4.22) is of the form (4.19). The system can be solved explicitly using Algorithm 1, setting $\mathbf{c} = 0$.

Velocity-verlet was used to produce all performance results for classic Boris presented in this study unless otherwise is noted. The concurrent velocity/position entries simplify simulation setup as well as data handling, since the half-step staggering of velocity need not be accounted for. This is particularly important when comparing simulations of varying time-step length as is done in Chapter 5 and 6. Particle velocities are also directly comparable to Boris-SDC, and no noticeable performance difference was observed compared to the leapfrog integrator.

4.3 Boris-SDC

The point of comparison for all the above methods and focus of the project is the Boris-SDC (Boris with Spectral Deferred Corrections) integration scheme for (4.2) first formulated by Winkel et al. [92]. Boris-SDC was designed to provide a high-order alternative to Boris, with strong conservation properties, improved accuracy and lower computational effort required to attain this accuracy.

Boris-SDC is fundamentally a collocation method solved via spectral deferred corrections, using the same trick as the Boris algorithm to avoid an implicit velocity dependence. To understand Boris-SDC, an appreciation of the collocation scheme and SDC is beneficial, and a short introduction following Ruprecht and Speck [67] for a simple first order ODE will be given before the derivation of the Boris-SDC method proper.

4.3.1 The Collocation Method

A way of solving the ODE

$$\frac{du}{dt} = f(u,t) \tag{4.23}$$

is sought. One way to achieve this is by approximating the RHS in the integral form of the ODE:

$$u(t_n + h) = u(t_n) + \int_{t_n}^{t_n+h} f(u(s))\ ds. \tag{4.24}$$

In the one-step collocation method [29], this is done by establishing the interpolating polynomial l, defined later, that satisfies (4.23) at the interpolation points and integrating this polynomial via quadrature instead. Establishing the interpolating polynomial of order M requires M nodes $\{\tau_1, ..., \tau_M\}$ and the quadrature is then performed at the same (*collocated*) nodes, hence the method name. The update from a time-step t_n to $t_{n+1} = t_n + h$ spanning the M collocation nodes then takes the form

$$u_i = u_0 + h\sum_{j=1}^{M} q_{ij} f(u_j), \qquad i = 1, ..., M, \tag{4.25a}$$

$$u_{n+1} = u_0 + \sum_{j=1}^{M} q_j f(u_j), \tag{4.25b}$$

where $u_i = u(\tau_i)$ and $u_0 = u(t_n)$. Note that indices i and j are used to denote collocation nodes (time sub-steps) to make a clear distinction from the global time-steps denoted by n. The quadrature weights q_{ij} and q_i are then defined as

$$q_{ij} := \int_{t_n}^{\tau_i} l_j(s)ds, \qquad i = 1, ..., M, \tag{4.26}$$

$$q_j := \int_{t_n}^{t_n+h} l_j(s)ds, \qquad j = 1, ..., M. \tag{4.27}$$

The Lagrange interpolating polynomial l_i associated with each node i is given by

$$l_j(s) = \prod_{i=1, i\neq j}^{M} \frac{s - \tau_i}{\tau_j - \tau_i}. \tag{4.28}$$

The polynomial and associated weights are defined on the unit interval $x \in [0, 1]$, which is mapped to the time-step interval $s \in [t_n, t_n + h]$ by $t = t_n + xh$.

The distribution of collocation nodes and associated quadrature weights does not have to be uniform across the time-step. Other types of quadrature can be used to yield a higher order Lagrange polynomial without increasing the number of nodes. Two such schemes are Gauss-Lobatto and Gauss-Legendre quadrature, which result in order $2(M-2)$ and $2M$ polynomials respectively. Gauss-Lobatto quadrature has the added benefit that the first and final nodes coincide with the beginning and end of the time-step, i.e. $\tau_1 = t_n$ and $\tau_M = t_{n+1}$. Refer to Hairer et al. [29] and Abramowitz and Stegun [2] (pg. 887-888) for more detail on Gaussian quadrature and the exact weights used in either scheme for a given number of nodes.

To perform the order M collocation time-step, the non-linear coupled M-dimensional system for F_i in (4.25a) must be solved, followed by the explicit update (4.25b). The collocation solution is symplectic for Gauss-Legendre nodes [29] and symmetric for Gauss-Lobatto nodes [32].

4.3.2 Spectral Deferred Corrections (SDC)

To avoid dealing with the non-linear system, likely to require root finding or inversion, converging approximations of the collocation solution can be made using the iterative SDC scheme. A simpler base method, like implicit or explicit Euler, is used to 'sweep' the time-step, establishing a first-order approximation of the solution at the collocation nodes. For implicit Euler

$$u_i = u_{i-1} + \Delta\tau_i f(u_i), \tag{4.29}$$

where $\Delta\tau_i = \tau_i - \tau_{i-1}$, the corresponding SDC iteration takes the form

$$u_i^{k+1} = u_{i-1}^{k+1} + \Delta\tau_i\big(f(u_i^{k+1}) - f(u_i^k)\big) + \sum_{j=1}^{M} s_{ij} f(u_j^k), \tag{4.30}$$

showing how several sweeps denoted by k are performed to establish and refine the collocation approximation. Here $u_0^k = u_0$ and $s_{ij} := q_{ij} - q_{i-1j}$ with $s_{1j} := q_{1j}$ are the node-to-node quadrature weights.

This SDC iteration converges to the collocation solution as $(f(u_i^{k+1}) - f(u_i^k) \to 0)$, which can be shown by taking the node-to-node update

$$u_i = u_{i-1} + \sum_{j=1}^{M} s_{ij} f(u_j), \tag{4.31a}$$

and taking the sum of all updates up to the node i yielding the 0-to-node update

$$u_i = u_0 + \sum_{l=1}^{i}\sum_{j=1}^{M} s_{ij} f(u_j) = u_0 + \sum_{j=1}^{M} q_{ij} f(u_j), \tag{4.31b}$$

which is identical to that of the collocation update (4.25a).

If Gauss-Legendre quadrature is used, a collocation method with M nodes should become order $2M$. SDC thus provides an explicit or simple implicit method for achieving order $2M$ integration of the original ODE. For each sweep, as SDC converges on the collocation solution, the order and beneficial properties of symplecticity of the collocation method are gradually retrieved. Xia et al. [93] showed that for an Euler base method, the integrator increases in convergence order at a rate of 1 order per sweep, i.e. each sweep increases the order of the method by 1 up to the collocation order.

The SDC solution must be initialised before the scheme can commence, i.e. the nodal arrays must be filled for $k = 0$. The simplest initialisation option is to simply set the values at all nodes $m = 1, ..., M$ equal to the initial value u_0. For a more accurate initialisation, a predictor can be applied, which refers to a single sweep of an explicit or implicit method without the collocation terms. In practice, doing so involves stepping through the collocation nodes using the base method on its own, providing a lower order starting point which is then improved iteratively using SDC.

4.3.3 Boris-SDC (2015)

This section contains the derivation of the first Boris-SDC formulation as proposed by Winkel et al. [92]. The integrator allows explicit high-order integration of (4.2) by using the velocity-Verlet/Boris integrator as a base method for SDC. The result is an explicit method that converges on the high-order collocation solution of the Newton-Lorentz system and maintains the associated energy stability if the collocation solution is retrieved. To apply the collocation method to (4.2), the system can be written in integral form for an arbitrary interval $[t_n, t_{n+1}]$ as

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_n}^{t} \mathbf{f}(\mathbf{x}(s), \mathbf{v}(s))ds, \tag{4.32a}$$

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_n}^{t} \mathbf{v}(s)ds, \tag{4.32b}$$

where initial values $\mathbf{x}(t_n)$, $\mathbf{v}(t_n)$ are denoted $\mathbf{x}_0$, $\mathbf{v}_0$.

To facilitate solving for a large number of particles N, the total degrees of freedom of the system (three coordinates per particle) are arranged in single column vectors as $[x_1^1, x_2^1, x_3^1, ..., x_1^N, x_2^N, x_3^N]^T$, so that $\mathbf{x}, \mathbf{v} \in R^d$ with the degrees of freedom dimension $d = 3N$.

For convenient notation when applying the collocation method to (4.32), the start-

ing point t_n is included in weight matrices and simply considered an un-weighted collocation node $\tau_0 := t_n$. Collecting the desired collocation weights q_{ij} and q_j in matrix $\hat{Q} \in R^{M\times M}$ and vector $\hat{q} \in R^M$ respectively, the weights matrices Q, q actually used include additional zero entries to account for the initial results at the 'node' τ_0, being defined as

$$Q := \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \hat{Q} \end{bmatrix}, \qquad q := [0, \hat{q}]. \tag{4.33}$$

The addition of intermediate nodes along the time-step for collocation, and subsequently SDC, requires storage and arithmetic involving d solution values for both $\mathbf{x}$ and $\mathbf{v}$ at every node, including τ_0. To facilitate this, the expanded solution vectors $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_M)^T \in R^{(M+1)d}$ and $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_M)^T \in R^{(M+1)d}$ are required. The expanded vector of function evaluations follows from this as $\mathbf{F}(\mathbf{X}, \mathbf{V}) = \big[\mathbf{f}(\mathbf{x}_0, \mathbf{v}_0), \mathbf{f}(\mathbf{x}_1, \mathbf{v}_1), ..., \mathbf{f}(\mathbf{x}_M, \mathbf{v}_M)\big] \in R^{(M+1)d}$.

The coupled collocation system for (4.32) then takes the form

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \mathbf{q}\mathbf{V} = \mathbf{x}_0 + \mathbf{q}\mathbf{V}_0 + \mathbf{q}\mathbf{Q}\mathbf{F}(\mathbf{X}, \mathbf{V}), \tag{4.34a}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_0 + \mathbf{q}\mathbf{F}(\mathbf{X}, \mathbf{V}), \tag{4.34b}$$

where the expanded matrix and vector of collocation weights $\mathbf{Q} = Q \otimes \mathbf{I}_d$ and $\mathbf{q} = q \otimes \mathbf{I}_d$ ensures all solution values at each node are multiplied by the appropriate node weight. Here $\otimes$ is the standard Kronecker product and $\mathbf{I}_d \in R^{d\times d}$ is the identity matrix.

The node-to-node SDC formulation for the collocation system (4.34) using velocity-Verlet (Section 4.2.2) as a base method then becomes

$$\begin{aligned}\mathbf{x}_{m+1}^{k+1} = \mathbf{x}_m^{k+1} &+ \Delta\tau_{m+1}\mathbf{v}_0 \\ &+ \sum_{l=1}^{M} S_{m+1,l}^{X}\big(\mathbf{f}(\mathbf{x}_l^{k+1}, \mathbf{v}_l^{k+1}) - \mathbf{f}(\mathbf{x}_l^{k}, \mathbf{v}_l^{k})\big) \\ &+ \sum_{l=1}^{M} SQ_{m+1,l}\big(\mathbf{f}(\mathbf{x}_l^{k}, \mathbf{v}_l^{k})\big),\end{aligned} \tag{4.35a}$$

$$\begin{aligned}\mathbf{v}_{m+1}^{k+1} = \mathbf{v}_m^{k+1} &+ \frac{\Delta\tau_{m+1}}{2}\big(\mathbf{f}(\mathbf{x}_{m+1}^{k+1}, \mathbf{v}_{m+1}^{k+1}) - \mathbf{f}(\mathbf{x}_{m+1}^{k}, \mathbf{v}_{m+1}^{k})\big) \\ &+ \frac{\Delta\tau_{m+1}}{2}\big(\mathbf{f}(\mathbf{x}_{m}^{k+1}, \mathbf{v}_{m}^{k+1}) - \mathbf{f}(\mathbf{x}_{m}^{k}, \mathbf{v}_{m}^{k})\big) \\ &+ \sum_{l=1}^{M} S_{m+1,l}\mathbf{f}(\mathbf{x}_l^{k}, \mathbf{v}_l^{k}),\end{aligned} \tag{4.35b}$$

where the node spacings are given by $\Delta\tau_{m+1} = \tau_{m+1} - \tau_m$, for $m = 0, ..., M-1$ in the case of Gauss-Legendre nodes and $m = 1, ..., M-1$ for Gauss-Lobatto. Subscripts here are simply used to indicate row and column indices used for the matrices, with SQ denoting the product of the $(M+1) \times (M+1)$ matrices S and Q containing the node-to-node and full collocation weights as defined for Q previously.

The various permutations of the node-to-node S matrix are defined as follows, $S_{ij} = Q_{ij} - Q_{i-1j}$, where $S_{1j} = Q_{1j}$. Likewise, $S_{ij}^{X} = Q_{ij}^{X} - Q_{i-1j}^{X}$, with $S_{1j}^{X} = Q_{1j}^{X}$.

The propagation matrix for velocity-Verlet $Q^X := Q^E Q^T + \frac{1}{2}(Q^E \circ Q^E)$ is derived from the propagation matrices for explicit Q^E and implicit Q^I Euler

$$Q^E := \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ \Delta\tau_1 & 0 & 0 & & \\ \Delta\tau_1 & \Delta\tau_2 & 0 & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \\ \Delta\tau_1 & \Delta\tau_2 & ... & \Delta\tau_M & 0 \end{bmatrix}, \qquad Q^I := \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & \Delta\tau_1 & 0 & & \\ 0 & \Delta\tau_1 & \Delta\tau_2 & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & \Delta\tau_1 & \Delta\tau_2 & ... & \Delta\tau_M \end{bmatrix} \tag{4.36}$$

as well as the trapezoidal rule $Q^T = \frac{1}{2}(Q^E + Q^I)$. Here $\circ$ is the Hadamard product,

multiplying the same-sized matrices element by element.

Term two and three of the (4.35b) RHS will be known from the previous iteration k and can be condensed into the known constant term

$$\Delta\tau_{m+1}\mathbf{c}_k := -\frac{\Delta\tau_{m+1}}{2}\left(\mathbf{f}(\mathbf{x}_{m+1}^k, \mathbf{v}_{m+1}^k) + \mathbf{f}(\mathbf{x}_m^k, \mathbf{v}_m^k)\right) + \sum_{l=1}^{M} S_{m+1,l}\mathbf{f}(\mathbf{x}_l^k, \mathbf{v}_l^k). \quad (4.37)$$

For the 2015 formulation, magnetic field was assumed constant to simplify the remaining arithmetic. The extension to inhomogeneous magnetic field is included in both the 2019 and relativistic formulations. Writing out the function evaluations $\mathbf{f}$ from (4.2) with constant $\mathbf{B}$, the velocity update step (4.35b) becomes

$$\frac{\mathbf{v}_{m+1}^{k+1} - \mathbf{v}_m^{k+1}}{\Delta\tau_{m+1}} = \frac{q}{m}\left[\mathbf{E}_{m+\frac{1}{2}}^{k+1} + \frac{\mathbf{v}_{m+1}^{k+1} - \mathbf{v}_m^{k+1}}{2} \times \mathbf{B}\right] + \mathbf{c}_k, \quad (4.38)$$

which becomes the correct form (4.19) for solution by Algorithm 1 when multiplied through by $\Delta\tau_{m+1}$. Velocity-Verlet/Boris is the base method used, where $\mathbf{E}$ rather than $\mathbf{v}$ is averaged over the time-step, calculated linearly as

$$\mathbf{E}_{m+1/2}^{k+1} = \frac{\mathbf{E}(\mathbf{x}_m^{k+1}) + \mathbf{E}(\mathbf{x}_{m+1}^{k+1})}{2}. \quad (4.39)$$

Essentially, the particular SDC formulation chosen for the Lorentz system results in a node-to-node update step for velocity and position with the same kind of (seemingly) implicit dependence on velocity found in the leapfrog and velocity-Verlet schemes. The Boris trick can thus be used to make the equations explicit in terms of previous node values. Implemented, the position is updated first which allows the interpolation of the electric field to the half-node point. The velocity can then be updated using Algorithm 1.

As an SDC method, Boris-SDC will converge towards the underlying collocation

solution with every iteration $k \to K$, eventually regaining symplecticity/symmetry and reaching order $2M$ or $2M-2$ depending on the type of quadrature used. This behaviour as well as favourable computational efficiency compared to classic Boris was demonstrated for a single particle and cloud of particles in a Penning trap using simple imposed electric and magnetic fields by Winkel et al. [92].

4.3.4 Boris-SDC (2019)

The 2019 formulation of Boris-SDC by Tretiak and Ruprecht [79] follows a slightly different derivation, but still applies the principles of collocation and SDC laid out in the beginning of Section 4.3. Starting again from the integral version of (4.2) over the time-step $t_n < t < t_{n+1}$:

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_n}^{t} \mathbf{v}(s) ds, \tag{4.40}$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_n}^{t} \mathbf{f}\left(\mathbf{x}(s), \mathbf{v}(s)\right) ds = \mathbf{v}_0 + \int_{t_n}^{t} \mathbf{f}\left(\mathbf{x}(s), \mathbf{v}(s)\right) ds, \tag{4.41}$$

where $\mathbf{x}_0 = \mathbf{x}(t_n)$.

A number of quadrature nodes are introduced to discretise the equation, denoted $t_n \leq \tau_1 < ... < \tau_M \leq t_{n+1}$, yielding the approximations

$$\int_{t_n}^{t} \mathbf{v}(s) ds \approx \sum_{m=1}^{M} q_m \mathbf{v}_m, \tag{4.42}$$

$$\int_{t_n}^{t} \mathbf{f}\left(\mathbf{x}(s), \mathbf{v}(s)\right) ds \approx \sum_{m=1}^{M} q_m \mathbf{f}\left(\mathbf{x}_m, \mathbf{v}_m\right), \tag{4.43}$$

where $\mathbf{x}_j, \mathbf{v}_j$will denote the approximations at nodes τ_j. The approximate solution

at t_{n+1} are then given by

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + \sum_{m=1}^{M} q_m \mathbf{v}_m, \tag{4.44}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_0 + \sum_{m=1}^{M} q_m \mathbf{f}\left(\mathbf{x}_m, \mathbf{v}_m\right). \tag{4.45}$$

In the 2015 formulation, the velocity in (4.44) is substituted into equation (4.45), which is the main difference. In the 2019 formulation, this substitution is neglected and the SDC treatment continues as in the previous section. The final node-to-node position update for 2019 Boris-SDC is then

$$\mathbf{x}_m^{k+1} = \mathbf{x}_{m-1}^{k+1} + \Delta\tau_m(\mathbf{v}_{m-1/2}^{k+1} - \mathbf{v}_{m-1/2}^{k}) + \mathbf{IV}_m^k \tag{4.46}$$

where

$$\mathbf{IV}_m^k := \sum_{j=1}^{M} S_{m,j} \mathbf{v}_j^k, \tag{4.47}$$

and the half-node velocity can be calculated via

$$\mathbf{v}_{m-1/2}^k = \mathbf{v}_{m-1}^k + \frac{\Delta\tau_m}{2}\mathbf{f}(\mathbf{x}_{m-1}^k, \mathbf{v}_{m-1}^k). \tag{4.48}$$

The velocity update remains unchanged from (4.35b), but is repeated here for completeness sake. In more compact notation the update reads

$$\mathbf{v}_m^{k+1} = \mathbf{v}_{m-1}^{k+1} + \frac{\Delta\tau_m}{2}\left[\mathbf{f}_{m-1}^{k+1} + \mathbf{f}_m^{k+1}\right] - \frac{\Delta\tau_m}{2}\left[\mathbf{f}_{m-1}^{k} + \mathbf{f}_m^{k}\right] + \mathbf{IF}_m^{k+1}, \tag{4.49}$$

where

$$\mathbf{IF}_m^k := \sum_{j=1}^{M} S_{m,j}\mathbf{f}_j^k. \tag{4.50}$$

The equation can be rearranged by expanding the force terms and gathering known values (iteration k) into a constant term, giving the correct form (4.19) for solution by Algorithm 1:

$$\mathbf{v}_{m+1}^{k+1} = \mathbf{v}_m^{k+1} + \Delta\tau_m E + \Delta\tau_m \frac{\mathbf{v}_m^{k+1} + \mathbf{v}_{m+1}^{k+1}}{2} \times B + c_k, \tag{4.51}$$

where

$$E := \frac{q}{m}\frac{\mathbf{E}(\mathbf{x}_m^{k+1}) + \mathbf{E}(\mathbf{x}_{m+1}^{k+1})}{2}, \tag{4.52}$$

$$B := \frac{q}{m}\mathbf{B}(\mathbf{x}_{m+1}^{k+1}), \tag{4.53}$$

$$c_k = c_1 - \frac{\Delta\tau_m}{2}\mathbf{v}_m^{k+1} \times B, \tag{4.54}$$

$$c_1 = \frac{q}{m}\frac{\Delta\tau_m}{2}\mathbf{v}_m^{k+1} \times \mathbf{B}(\mathbf{x}_m^{k+1}) - \frac{\Delta\tau_m}{2}\left[\mathbf{f}(\mathbf{x}_m^k, \mathbf{v}_m^k) + \mathbf{f}(\mathbf{x}_{m+1}^k, \mathbf{v}_{m+1}^k)\right] + \mathbf{IF}_m^k \tag{4.55}$$

The update shown here is slightly different from the 2015 derivation shown in the previous section, which was for a constant magnetic field. Here no such assumption is made and the formulation is valid in the presence of inhomogeneous magnetic fields which change with particle position.

4.4 Alternative Particle Integrators

Various alternative schemes or extensions to Boris are reviewed in this section, starting with integrators not based on Hamiltonian mechanics.

In the work of Stoltz et al. [72], a Boris-like algorithm was devised to better facilitate plasma beam simulations. Beam simulations are often performed using the beam length-wise spatial coordinate as the independent variable rather than time. Thus, an algorithm was developed wherein the z-coordinate is advanced through a number of steps size Δz yielding the time t and beam cross-sectional coordinates x, y for each particle at each step. In practice, the algorithm works much like Boris with the application of a central difference scheme to advance the generalised particle position $s = [x, y, ct]$ w.r.t. z, and thus remains second-order accurate as well. Applied, the spatial Boris-like scheme generally offers no advantage over Boris except for beam-like problems, where retrieving particle data s for some z is generally advantageous when studying spatial structures.

It is of natural interest to match any high order accuracy in the time discretisation of PIC with a high order field discretisation and vice versa. Doing so allows both high order schemes to be used to their full advantage and avoids the error of the lower order method dominating the solution. Quandt et al. [63] proposed a particle integrator based on an expanded Taylor series for high order particle integration. Such an expansion enables the approximation of particle position with arbitrary accuracy just like Boris-SDC, but requires high order derivatives from the electric and magnetic field discretisations to match. In practice, the field discretisation would also have to be of high order and capable of providing these derivatives. As such, free choice in what particle/field solvers to use for the combined PIC scheme is lost as is any advantage in using high order particle integration in isolation.

Relativistic velocities have been discovered as another limitation for the Boris algorithm, with the Newton-Lorentz force becoming non-linear due to the addition

of the velocity dependent relativistic factor γ

$$\frac{d\mathbf{u}}{dt} = \frac{q}{m}\left(\mathbf{E} + \mathbf{B} \times \mathbf{v}\right), \tag{4.56}$$

where $\mathbf{u}$ is the proper velocity as experienced by the particle, $\mathbf{v} = \mathbf{u}/\gamma(\mathbf{u})$ and $\gamma(\mathbf{u}) = \sqrt{1 + \frac{|\mathbf{u}|^2}{c^2}}$.

The relativistic factor must be approximated in some way in order to apply Algorithm 1. In leapfrog Boris this is typically done by using the relativistic factor of the known pre-rotation velocity. See Section 5.2 in the subsequent chapter for more detail on the relativistic case and the consequences for particle integration.

Vay [82] determined that relativistic leapfrog Boris does not correctly compute cancelling of equal $\mathbf{E}$ and $\mathbf{v} \times \mathbf{B}$ terms, leading to spurious forces and unphysical drift. The cause of the problem is the average velocity term and corrected this term for a proposed modified Boris integrator more suitable to applications with cancelling electromagnetic force terms. The resulting second-order integrator was found to preserve the correct interaction $\mathbf{E}$ and $\mathbf{v} \times \mathbf{B}$ terms at the cost of additional calculation complexity. It was later found by Higuera and Cary [36] that this integrator is not phase-space volume conserving, with the volume alternately shrinking and growing in some cases, to give the appearance of conservation. A new integrator was then proposed in which the relativistic factor was calculated via a quadratic expression. The Higuera integrator was compared to the Boris and Vay integrators and found to both be both volume conserving for the problem studied and preserve the correct $\mathbf{E} \times \mathbf{B}$ velocity. In another take on the same issue, Qiang [60] devised a second order algorithm similar to that of Boris, but performing electric and magnetic forcing in the same operation. The new method was similar to Boris and correctly preserved $\mathbf{E} \times \mathbf{B}$ terms, but with simplified algebra and fewer operations per time-step compared to the Vay and Higuera integrators. As a drawback, the new integrator was not volume-preserving, and is thus be susceptible to numerical error for long simulations. However, tests performed by the authors showed

identical results to the more complicated volume-preserving relativistic algorithm in tracking the orbit of an electron for 500,000 periods. In a different study, Qiang [61] also demonstrated a method for extending existing symmetric integrators to higher order. fourth-order variants of Boris, Vay and Higuera-Cary integrators were constructed and their higher order convergence demonstrated. No direct comparison was made between second and fourth-order integrators in terms of either time-step or computational effort however.

A combined high-order and energy-conserving particle integrator scheme was proposed by Li and Wang [47]. Using energy-conserving quadrature analysis similar to collocation, arbitrarily high order integration was achieved (depending on the type of quadrature nodes used) with the high order implicit terms treated using Runge-Kutta methods. Second, fourth and sixth order integrators were demonstrated in comparison with Boris in terms of global error vs simulation runtime. High-order convergence and increased computational efficiency compared to Boris was demonstrated for the fourth and sixth order integrators, albeit using only four simulations per integrator with no cross-over actually visible in the results. Energy was conserved by the new integrators throughout the simulation with energy error orders of magnitude lower than Boris.

Hamiltonian analysis has been used to construct a range of volume-conserving or symplectic numerical integrators for Hamiltonian systems, with methods for constructing explicit volume-conserving schemes of high (even) orders proposed by Yoshida [96]. Recent work has extended this previous methodology to Lorentz force integration for particles in electromagnetic fields. He et al. [34] improved upon existing volume-conserving schemes to construct second and fourth-order accurate particle integration methods. Starting from volume-conserving splits of the Lorentz system (4.2) developed in the previous literature, the authors show how a technique called “processing” can be used to elevate a general Hamiltonian method to higher order without adding multiple time stages. As a demonstration, second and fourth-

order methods are derived and shown to exhibit superior performance to traditional multi-stage higher order volume-conserving methods. Using their approach, constructing volume-conserving methods of arbitrary order based on most Hamiltonian solution techniques for (4.2) should be possible.

A new class of volume-conserving integrators for charged particles were developed by Hairer and Lubich [30], by performing multi-step extensions on a variational version of the Störmer-Verlet integrator which becomes equivalent to Boris for constant magnetic field. The resulting formulation allows the construction of any desired volume-conserving integrator of arbitrary order $2k$ where k is the number of central difference extensions, i.e. the method gains one formal order per added step. The fourth-order method constructed using the scheme was shown to conserve both energy and momentum for a single particle over a minimum of ten million time-steps.

By reformulating three second-order volume-conserving integrators, one being Boris, the above-mentioned study by Qiang [61] used the same methodology as Yoshida [96] to construct explicit fourth-order integrators. No long term study of the energy error was performed, but the construction method from Yoshida [96] should conserve the volume-conserving nature of the integrators in question.

Tao [76] proposed an explicit, fully symplectic integrator for the non-relativistic Lorentz system with time-dependent electric and magnetic fields. Named Explicit Symplectic Shadowed Runge-Kutta (ESSRK) by the author, construction of arbitrary order p symplectic integrators is performed via simple Hamiltonian analysis. An implicit term arises, which can be treated using a p-order Runge-Kutta approximation of the position followed by a shadowed update of the momentum using the position approximation. Arbitrary order is then achieved following the example of previous authors like Yoshida [96] and Suzuki [73]. The author demonstrates that the fourth-order scheme performs as desired in terms of accuracy compared to the equivalent fourth-order Runge-Kutta, with the expected long term conservation of

energy not present using RK4. The performance tests were conducted for a particle trapped in a toroidal magnetic field similar to those in fusion experiments, as well as for a particle brought to resonance in a periodic perturbed magnetic field.

Matsuyama and Furukawa [52] proposed an integrator combining multiple points of the current discussion. The integrator was shown to be explicit, volume-conserving and with proper treatment of $\mathbf{E} \times \mathbf{B}$ drift. Additionally, using the techniques first proposed by Yoshida [96], the integrator was shown to be extendable to higher order and high-order convergence behaviour was demonstrated in comparison with Boris and fourth-order Runge-Kutta.

With these new contributions, Hamiltonian integrators are becoming a promising avenue for particle pushing via Lorentz force integration. These approaches have yet to come into proper use in particle plasma codes however and remain largely untested compared to the Boris algorithm and derivatives thereof. The Hamiltonian integrators also exhibit the common disadvantage of increased complexity compared to traditional integrators. None of the reviewed high-order integrators except potentially that of Li and Wang [47] appeared to allow an increase in convergence order using a single implementation from input parameters alone, as Boris-SDC allows. Several of the new particle integrators were tested, verified for their high convergence order and compared to Boris. However, no investigation was found of the increase in computational effort associated with higher order integration and comparison of this increase with Boris, with the exception of a single test case by Li and Wang [47]. In the context of PIC, improvements to Boris can be construed as improvements to PIC and improvements made in the reviewed studies include high-order convergence, better conservation and fixing unphysical drift. None of these improved integrators were actually tested as part of a PIC scheme however, and the combined performance outlook thus remains uncertain. The same lack of work on PIC with higher order particle integration was noted for the PIC literature reviewed in Chapter 3. Taken as a whole, plenty of work on higher order particle

integration and PIC has been done, but no studies exist in which the impact of the former on the latter is isolated and directly assessed. More specifically, the interplay in accuracy and computational performance of using a high order integrator and standard second order EMPIC/ESPIC scheme remain unexplored.

The lack of studies testing the performance of new integrators within the context of PIC motivated the current work on PIC/Boris-SDC. As mentioned in Chapter 1, improved computational efficiency was demonstrated by [92] for Boris-SDC over classic Boris in a case of pure particle motion in known electromagnetic fields. In PIC however, the field values are approximations and it was unclear how well the favourable performance of Boris-SDC would hold up within a PIC scheme. The same argument applies to the alternative integrators reviewed above, with the noted lack of work in the topic of high-order particle integration performance in the context of PIC. Putting Boris-SDC to the test in PIC will thus contribute to the issue of particle integration performance for PIC in a wider context, not just specifically for Boris-SDC.

Chapter 5

Implementation and Preliminary Results

The current chapter contains an introduction to the structure and functioning of the ESPIC code developed for the project, followed by an overview of the algorithmic modifications required to combine the ESPIC and Boris-SDC methods. To validate and verify the Boris-SDC and velocity-Verlet implementations in isolation, the single-particle Penning trap results of the previous study [92] were replicated. In the subsequent section, performance of Boris-SDC and velocity-Verlet is studied for test cases using increasing levels of ESPIC approximation of the electric field values, providing initial results for the effect of the spatial solver on particle integration. Following the electrostatic implementation and results, Section 5.2 presents the derivation of relativistic velocity-Verlet after a brief introduction to the relevant parts of Special Relativity. Using the relativistic velocity-Verlet as a foundation, a novel generalisation to the relativistic regime is proposed for the 2019 Boris-SDC formulation. The new relativistic Boris-SDC integrator is then verified for high-order convergence using a relativistic reproduction of the Penning trap test used to test the first formulation of Boris-SDC by Winkel et al. [92]. The chosen EMPIC code Runko is an existing, open-source multi-physics code focused on plasma PIC

simulation. The code will be introduced in more detail in Section 5.3, with a focus on the special unit system used for the code. Implementation considerations regarding relativistic Boris-SDC follow along with verification and preliminary results akin to those of Section 5.1, namely testing of the particle integrator implementations in isolation and combined with parts of the PIC functionality.

5.1 Project ESPIC Code: KPPS

5.1.1 Introduction

Performance evaluation of electrostatic PIC using Boris-SDC as the particle pusher was completed using the custom simulation code KPPS. Performance in the context of this study refers to the reduction in error with respect to time-step size and arithmetic complexity. The performance does not refer to real runtime performance, as the test codes have yet to be optimised for computational efficiency. Designed to be as open and simple to expand as possible, the fundamental setup is focused on tracking particles in 3D space continuously and relating these to a finite difference mesh representing the changing electric field. Particles are grouped into species, which defines critical parameters such as charge, mass, and charge-mass ratio for all members. Vectors are defined for species quantities which differ in position, velocity and parameters which depend on these quantities such as the local field values. All other operations were defined to relate to these two fundamental structures, species and mesh. 2D and 1D simulation was achieved by using flat or line shaped meshes respectively as well as enforcing zero velocity in extraneous particle vector components.

KPPS was established in Python as a fully object-oriented program consisting of several separate modules. The software definition of the term 'module' is used here, referring to an independent piece of code with a specific functionality, several

of which form the main program when linked.

The structure of KPPS (Figure 5.1, overleaf) takes the form of a main module, which can be run and passed the model settings, which include the desired species, field grid/mesh and analysis parameters. Example parameters for each of these three categories include particle count for each species, mesh resolution and choice of integration methods, respectively. The main script constructs and ties the other modules together and is responsible for handling the time-loop.

Before the main time-loop, the species and mesh loaders are invoked and set the initial particle and field conditions as well as calculating and enforcing dimensionality, boundary conditions, etc.

The analysis module uses the physical model settings when constructed to form lists of the required methods for achieving the desired analysis, to be run every time-step. For example, particle to field interpolation, followed by field integration, followed by particle integration, repeated every time-step. Fields can be evaluated at the particle position on demand, requiring a full loop of the desired methods providing ESPIC field interpolation, solution and reverse interpolation.

Finally the data handler module can be called to record the data of the current time-iteration, either by dumping the mesh and all species objects to file for use in analysis by scripts or by writing particle position and velocity to .vtk files for use in visualisation software like Paraview.

The passing and storage of data was developed to follow a pipe-line methodology, with the two main structures of interest (particle species and finite difference mesh) being defined by data-carrying objects, passed through the program and manipulated by a pre-defined set of analysis operations.

All particle data specific to a given species, including constants, properties, posi-

tions, velocities and field values, are encapsulated within the "species" object. Any number of species objects can be established and analysed, allowing flexibility in how the necessary particle sets are ordered. The same approach was taken for the field mesh, with the "mesh" object holding all parameters and data at the current time-step for the cartesian mesh. This data includes the charge density, electric potential, electric and magnetic fields.

In addition to the two main groupings of the physical simulation setup, simulation control parameters were combined into a "controller" object, which includes static reference values such as start/end times, time-step size and number. Most importantly, the object holds the current time and simulation status, updated as the program progresses, used to control the simulation.

Thus, the flow arrows of Figure 5.1 indicate the flow of the three data-holding objects through a given simulation run.

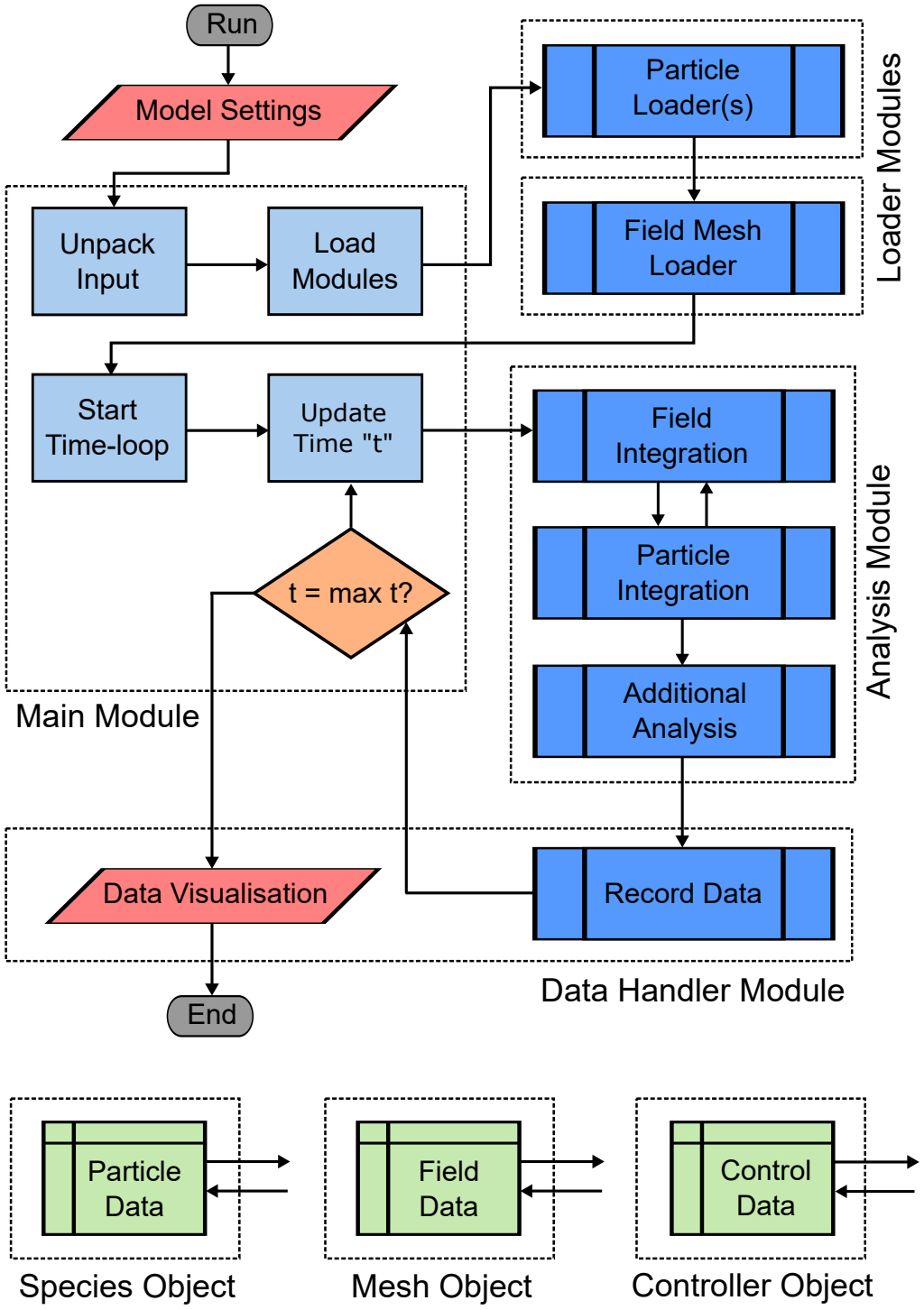

Figure 5.1: KPPS Flow Schematic

5.1.2 Integrating the Boris-SDC and PIC Algorithms

The use of velocity-Verlet as the base of Boris-SDC presents a challenge to its integration within a standard leapfrog-based ESPIC scheme. In the leapfrog integration

scheme, velocity is updated before position in any given time-step. This order requires the field to be updated after the particle time-step, so as to provide the correct, up-to-date field values to the particle integrator during the next time-step. Recall the structure of velocity-Verlet

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{f}(t)\Delta t^2, \tag{5.1}$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t) + \mathbf{f}(t + \Delta t)}{2}\Delta t, \tag{5.2}$$

here without the Newton-Lorentz force $\mathbf{f}(t) = \frac{q}{m}\big(\mathbf{E}(t) + \mathbf{v}(t) \times \mathbf{B}(t)\big)$ expanded in the velocity update.

In velocity-Verlet, the order of the particle update is reversed, and the use of Algorithm 1 to solve the velocity update explicitly requires the updated field values corresponding to the new position. Since the acceleration/fields must be evaluated between the position and velocity updates, velocity-Verlet cannot be implemented for schemes where the particle integrator must act separately within a single function without access to the field solver. In traditional ESPIC codes, the spatial field operations are often performed in one separate routine and the interpolated values are subsequently provided to the particle integrator. The particle integration routine is also self-contained, calculating velocity and position updates immediately based on the field approximations from the spatial solver, after which the cycle can begin again. Directly replacing leapfrog with velocity-Verlet in this setup means that velocity-Verlet does not have access to the updated electric field $\mathbf{E}(t + \Delta t)$, needed in the Boris algorithm, because the spatial solver has not been run to provide updated field values corresponding to $\mathbf{x}(t + \Delta t)$. In general, application of the Boris algorithm requires the field to be approximated using updated positions in order to update the velocity, i.e. the field update comes "between" the position and

velocity updates. In the case of leapfrog, this occurs naturally when the particle integration and field solvers are separate modules within the code, run sequentially. In the case of velocity-Verlet, the field solver must either be accessible from within the particle integrator, or the particle integrator must be split into two separate updates, performed before (position) and after (velocity) the spatial solver runs.

The picture is further complicated when it comes to Boris-SDC, in which the position and velocity integration is performed for every quadrature node calculation in every sweep. For Gauss-Lobatto nodes, this equates to $(M-1)K$ field integrations per time-step, so Boris-SDC with 5 Gauss-Lobatto nodes and 5 sweeps requires 20 field integrations per time-step.

Accepting this cost for Boris-SDC, an ESPIC scheme using velocity-Verlet Boris or Boris-SDC merely requires a slightly different structure for the mathematical operations: Performing the field integration in the middle of every particle integration. Recalling the node-to-node formulation of Boris-SDC and fundamental equations/-operations of ESPIC, the combined ESPIC/Boris-SDC scheme is outlined below, written in terms of the necessary mathematical operations. While the exact arithmetic differs between the two Boris-SDC formulations, the PIC procedure remains the same using either version.

For an arbitrary time-step starting at t_n, the updated values $\mathbf{x}(t_n+\Delta t)$, $\mathbf{v}(t_n+\Delta t)$ and $\mathbf{f}(t_n+\Delta t)$ are sought. First, position, velocity and force values must be initialised for SDC iteration at the collocation nodes. The simplest setup involves no predictor, and the initial values at t_n is copied to all the nodes to give a starting point, SDC iteration $k = 0$ at all nodes m.

$$\mathbf{x}_m^{k=0} = \mathbf{x}(t_n) \quad \text{for } m = 0, 1, \ldots, M, \tag{5.3}$$

$$\mathbf{v}_m^{k=0} = \mathbf{v}(t_n) \quad \text{for } m = 0, 1, \ldots, M, \tag{5.4}$$

$$\mathbf{f}\big(\mathbf{x}_m^{k=0}, \mathbf{v}_m^{k=0}\big) = \mathbf{f}\big(\mathbf{x}(t_n), \mathbf{v}(t_n)\big) \quad \text{for } m = 0, 1, \ldots, M. \tag{5.5}$$

With an initial solution established, node-to-node stepping from node 1 to M is performed for every SDC iteration k.

For $m = 1, 2, \ldots, M$ at $k = 1, 2, \ldots, K$, repeat:

1. Update position to yield $\mathbf{x}_{m+1}^{k+1}$ using (4.35a) and (4.46).

2. Deposit particle charge to the mesh nodes i and calculate charge density, following the procedure in Section 3.2:

$$q_i^{k+1}(\tau_{m+1}) = \sum_{j=1}^{N_q} W_i(\mathbf{x}_{m+1}^{k+1}) q, \tag{5.6}$$

$$\rho_i^{k+1}(\tau_{m+1}) = \frac{q_i^{k+1}(\tau_{m+1})}{\Delta V}. \tag{5.7}$$

3. Solve Poisson's equation to yield the electric potential ϕ_i and differentiate to give field $\mathbf{E}_i$. The two are defined together on every mesh node and correspond to the instantaneous particle distribution, following the procedure introduced in Section 3.2:

$$\triangle \phi_i^{k+1}(\tau_{m+1}) = \frac{\rho_i}{\epsilon}. \tag{5.8}$$

$$\mathbf{E}_i^{k+1}(\tau_{m+1}) = \nabla \phi_i^{k+1}(\tau_{m+1}). \tag{5.9}$$

4. Interpolate the mesh electric field to the particle positions following the pro-

cedure in Section 3.2:

$$\mathbf{E}(\mathbf{x}_{m+1}^{k+1}) = \sum_{i=1}^{N} W(\mathbf{x}_{m+1}^{k+1})\mathbf{E}_i^{k+1}(\tau_{m+1}), \tag{5.10}$$

5. Update velocity to yield $\mathbf{v}_{m+1}^{k+1}$ using (4.35b)-(4.39)/(4.49)-(4.55) and Algorithm 1.

Finally, integrate to new time-step node $t_{n+1} = t_n + \Delta t$. For Gauss-Lobatto nodes, the new time and final collocation node coincide $\tau_M = t_{n+1}$, and thus $\mathbf{x}(t_{n+1}) = \mathbf{x}_M^K$, $\mathbf{v}(t_{n+1}) = \mathbf{v}_M^K$.

Presented in a simplified schematic in Figure 5.2, the flow of operations in Boris-SDC integrated ESPIC follows the general ESPIC flow (Figure 3.4), but with the ESPIC cycle repeated for each collocation node update and with a split particle push.

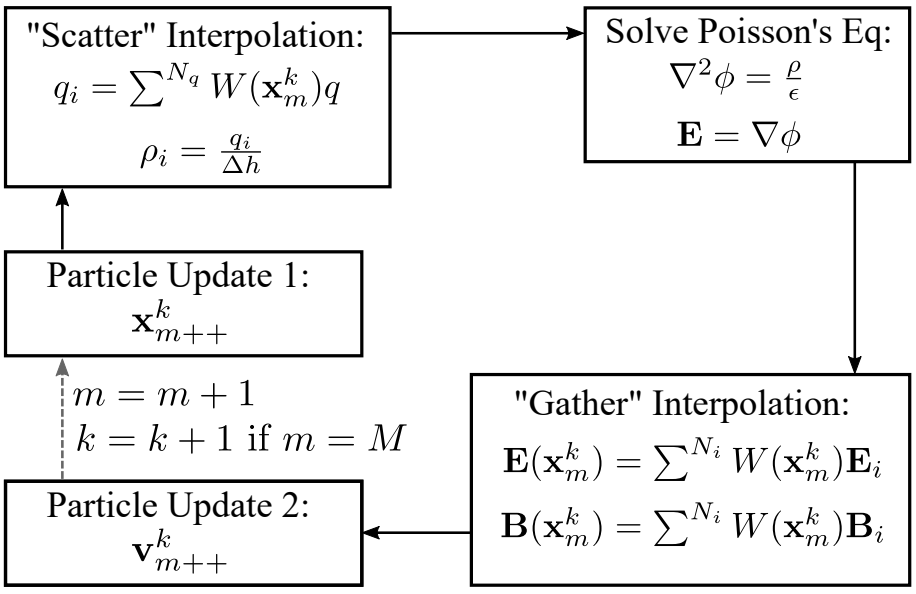

Figure 5.2: Integrated ESPIC/Boris-SDC Algorithm

5.1.3 Boris-SDC: Single-Particle Penning Trap

To verify the Boris-SDC implementation within KPPS separate from PIC, the expected high-order convergence characteristics must be demonstrable. The KPPS/Boris-SDC implementation was therefore compared to the work-precision results of the original study by Winkel et al. [92]. Conducting a work-precision study not only allows the convergence behaviour of the time integrator to be evaluated but shows the relative computational expense between multiple integrators when compared side to side. The original study used a single-particle Penning trap as a work-precision benchmark with the analytical solution used to establish the error for classic Boris and any given Boris-SDC configuration. The theory and simulation setup are reproduced below, with KPPS results based on the setup following.

The electric field

$$\mathbf{E}(\mathbf{x}_i) = -\epsilon\frac{\omega_E^2}{\alpha}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -2\end{bmatrix}\mathbf{x}_i, \tag{5.11}$$

and magnetic field

$$\mathbf{B} = \frac{\omega_B}{\alpha}\hat{\mathbf{z}}, \tag{5.12}$$

are used in combination, with $\mathbf{x}_i$ being the position of particle i, α being the charge to mass ratio and ω_E, ω_B determining the magnitude of the electric and magnetic fields respectively.

From Patacchini and Hutchinson [58] the analytical Penning trap trajectory of a single particle takes the form

$$z(t) = z(0)\cos(\tilde{\omega}t) + \frac{v_z(0)}{\tilde{\omega}}, \tag{5.13a}$$

along the magnetic field lines in z. The frequency $\tilde{\omega} = \sqrt{-2\epsilon} \cdot \omega_E$ for the chosen configuration consists of an ideal quadrupole electric field and constant magnetic field. In the xy-plane, the trajectory follows

$$w(t) := x(t) + iy(t) = (\mathcal{R}_+ + i\mathcal{I}_+)\exp(-i\Omega_+ t) + (\mathcal{R}_- + i\mathcal{I}_-)\exp(-i\Omega_- t), \tag{5.13b}$$

with

$$\Omega_\pm = \frac{1}{2}\left(\omega_B \pm \sqrt{\omega_B^2 + 4\epsilon\omega_E^2}\right). \tag{5.13c}$$

The real $\mathcal{R}_\pm$ and imaginary $\mathcal{I}_\pm$ terms are given by

$$\mathcal{R}_- = \frac{\Omega_+ x(0) + v_y(0)}{\Omega_+ - \Omega_-}, \qquad \mathcal{R}_+ = x(0) - \mathcal{R}_- \tag{5.14}$$

$$\mathcal{I}_- = \frac{\Omega_+ y(0) + v_x(0)}{\Omega_+ - \Omega_-}, \qquad \mathcal{I}_+ = y(0) - \mathcal{I}_- \tag{5.15}$$

Simulation parameters were set to match those used by Winkel et al. [92], see Table 5.1.

The final particle position was then computed in KPPS with both classic Boris and Boris-SDC as the particle time-integrators, for the case of $M = 3$ and $M = 5$ collocation nodes respectively, using a range of SDC iteration counts. The resultant work-precision plots show relative error compared to the analytical solution of the final particle x-coordinate w.r.t time-step size and can be seen in Figures 5.3 and 5.5.

Table 5.1: Validation study parameters

α	1
t_{end}	16
$\mathbf{x}(0)$	$(10, 0, 0)^{\mathrm{T}}$
$\mathbf{v}(0)$	$(100, 0, 100)^{\mathrm{T}}$
ω_E	4.9
ω_B	25.0
ϵ	-1

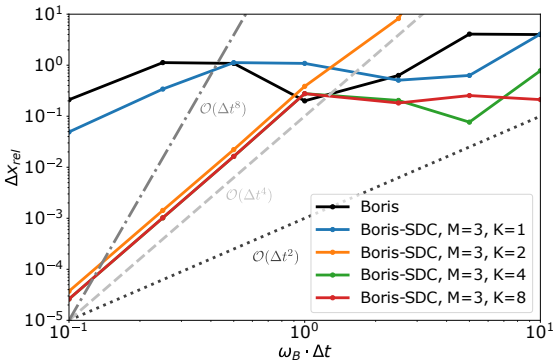

Figure 5.3: KPPS Penning trap work-precision for 3 Gauss-Lobatto nodes produced using KPPS.

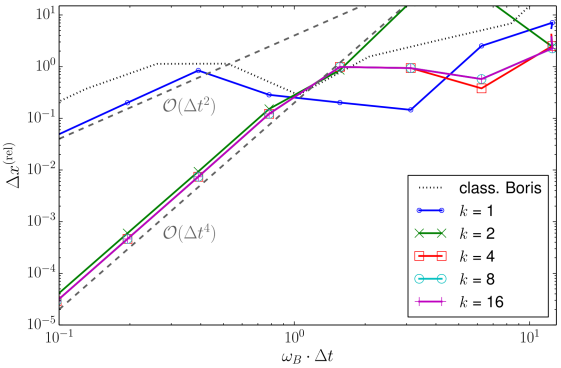

Figure 5.4: Penning trap work-precision for 3 Gauss-Lobatto nodes produced by [92].

Compared to the results of the previous study (Figures 5.4 and 5.6), the high-order convergence behaviour of Boris-SDC is captured well by the new implementation. For one SDC iteration ($K = 1$), the integrator produces results equivalent to the standard Boris algorithm and displays the expected second-order convergence once the time-step is short enough. As more iterations are added, the convergence order increases until the order of the underlying collocation method is reached: 4th order for 3 Gauss-Lobatto nodes and 8th order for 5 nodes. Recall from Section 4.3.2 on SDC that a one order increase per iteration is expected formally for first order Euler methods. In both the current and previous studies, approximately 2 orders are gained per iteration, which improves computational efficiency significantly. The 2 order gain is not guaranteed and was not always observed when using the method for more than 5 collocation nodes.

As an additional visual check that the particle motion is captured accurately by the KPPS implementation, a typical simulated trajectory using Boris-SDC can be

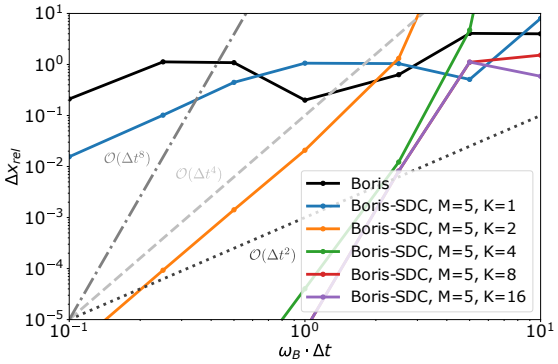

Figure 5.5: KPPS Penning trap work-precision for 5 Gauss-Lobatto nodes produced using KPPS.

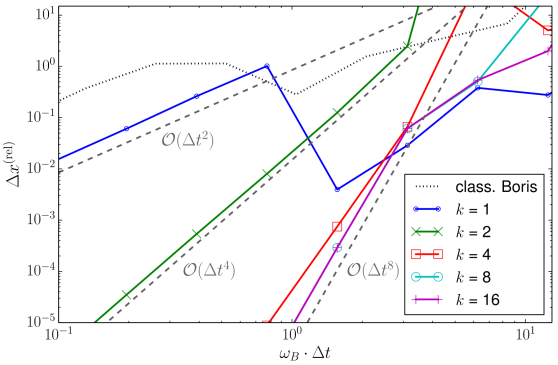

Figure 5.6: Penning trap work-precision for 5 Gauss-Lobatto nodes produced by [92].

seen alongside the analytical solution in Figure 5.7. No visual trajectory differences exist between Boris-SDC in KPPS compared to both the analytical solution and the results from the previous work (Figure 5.8).

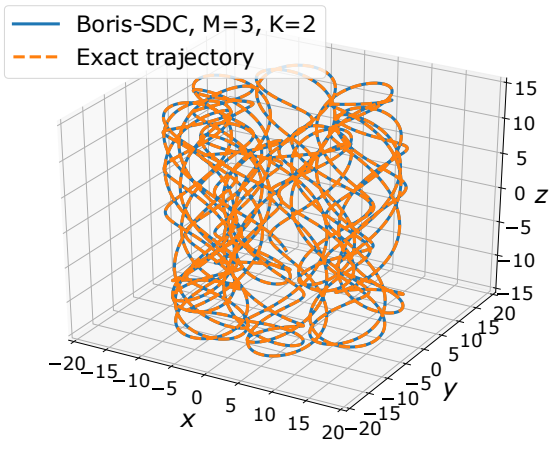

Figure 5.7: KPPS Penning trap particle trajectory $t = [0, 16]$ for 320 time-steps.

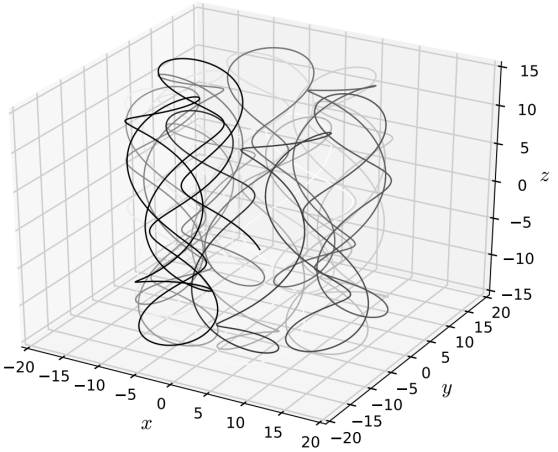

Figure 5.8: Penning trap particle trajectory for $t = [0, 16]$ produced by [92].

With the data produced by KPPS for the single-particle Penning trap matching both the analytical solution and the previous study, the Boris-SDC implementation was considered verified and ready for integration with PIC.

5.1.4 PIC with Boris-SDC: 1D Particle Oscillator

With the implementation of Boris-SDC completed and successfully tested, verification and initial performance evaluation of the full ESPIC implementation were sought. Recall that the PIC scheme consists of several layers of numerical approximation, including interpolation from particles to mesh, mesh to particles and solving Poisson's equation on the mesh. Each of these stages introduces error, the magnitude of which depends on the specific method and spatial resolution of the mesh. The aggregate error of the spatial PIC component methods combined with the error of the particle integrator then forms the combined global error of the full scheme compared to the continuous solution. Whichever component method, including the particle integrator, exhibits the highest error will set a lower bound on achievable total error magnitude for the scheme as a whole. Conducting a time integration performance study done from the bottom up, adding one layer of the ESPIC approximation at a time, will demonstrate the impact of each part of the combined ESPIC scheme on the time convergence and performance of Boris-SDC demonstrated in the previous section. Performance of the particle integrator was evaluated in the same manner as for the Penning trap, by studying the change in error with respect to time-step size. Note that it is now the global error produced by the full ESPIC scheme being evaluated, as opposed to the truncation error caused by the time discretisation of the particle integrator alone. However, as the particle integrator is solely responsible for evolving the PIC scheme in time, the error of the ESPIC solution w.r.t. time-step works as a performance metric for the integrator. This is not the case in EMPIC, wherein the Maxwell equations also require discretisation in time.

By studying how the solution to a simple test problem evolves when each component method of the ESPIC scheme is added to the solution process, each link in the implementation of the scheme was verified. Additionally, the study provides a detailed and quantitative insight into the impact of the numerical errors produced by the components of ESPIC and provides an initial performance comparison of Boris-

SDC vs. classic Boris. To the best of the author's knowledge, such an investigation has yet to be pursued in the literature.

A simple oscillator, consisting of a particle oscillating in 1D due to a quartic potential field ($\phi(x) = -\frac{1}{4}x^4$), was chosen as the test problem. The problem is ideal for study using different components of PIC as the corresponding potential, electric field and charge distributions are all easily calculable via simple differentiation and can then be used to setup the spatial solver routine under study. Crucially, the potential is not exactly solvable by the second-order Poisson discretisation, which would eliminate the spatial error and defeat the point of the study. The four test cases listed below, which neatly straddle the spectrum from no PIC to full ESPIC, were investigated. The parameters for the particle oscillation setup are summarised in Table 5.2.

Exact E-field at particles (no PIC): The exact electric field produced by the quartic potential ($E(x) = \nabla\phi(x) = -x^3$) is evaluated directly at the particle positions without the use of any spatial approximations. This case is equivalent to a simpler version of the Penning trap experiment from the previous section and should yield similar results. This case is a non-PIC solution for the problem.

Interpolated E-field from mesh: The exact electric field is evaluated on the mesh nodes, and the PIC gathering operation (linear interpolation of field to particle) is performed to determine approximated electric field values at the particle positions for the integrator to use. This case demonstrates the basic effect of limiting the accuracy of the higher order time discretisation using a lower order spatial discretisation.

Approximated E-field from Poisson solve: The quartic potential field is established on the mesh by the applying the exact equivalent charge density ($\rho(x) = \nabla^2\phi(x) = -3x^2$) at each mesh node and solving Poisson's equation via second-order finite differences (see Section 3.2). Second-order finite differences is applied again to

Table 5.2: Particle Oscillator Problem Parameters

Domain	X	$[-1, 1]$
End time	t_{end}	1
Initial position	x_0	0.5
Initial velocity	v_0	0
Potential	$\phi(x)$	$-\frac{1}{4}x^4$
Electric field	$E(x)$	$-x^3$
Charge density	$\rho(x)$	$-3x^2$

differentiate the potential into electric field values on the mesh. The electric field is then interpolated to the particles as in the previous case.

Approximated charge density from particle distribution (ESPIC): The potential well is now represented by a charge distribution using a large number of immobile positive charges in continuous space. The PIC scattering operation (linear interpolation of particle to field) is performed to determine the approximate charge distribution on the mesh. The potential at the mesh nodes is then established via a Poisson solve, electric field calculated via differentiation and interpolated back to the particle positions, all as above. This last case gives the closest to an ESPIC solution for the problem, albeit still without dynamic coupling of particles to the mesh, as only static particles are scattered. The number of stationary charges was kept constant at 20 particles per cell to ensure a smooth distribution at all cell counts investigated. The effect of varying particle resolution against the mesh resolution was not studied.

The chosen end time of 1 computational second was found to correspond to roughly 10% of the oscillation period and so the simulations do not cover the full oscillating motion. The short time-frame was selected to conserve computational resources and was deemed sufficient for producing a first look at the effect of the spatial solvers on particle time integration. The mesh used for interpolation and finite difference solving in the simulation consisted of equally sized cells, covering the full domain length, for some cell count NZ, giving a mesh of NZ+1 finite difference nodes. Error was calculated as the relative difference in particle position at t_{end}

$$\Delta z^{(\mathrm{rel})} = \frac{|z_{\mathrm{ref}} - z_{\mathrm{end}}|}{|x_{\mathrm{ref}}|}, \tag{5.16}$$

where $z_{\mathrm{end}} = z(t_{\mathrm{end}})$ and z_{ref} is the same position calculated by the particle integration, no ESPIC case which is exact with respect to the field values. The most accurate temporal accuracy was available was used, in the form of the Boris-SDC simulation with 8 nodes, 8 sweeps and 2000 time-steps.

Exact E Case

The results of the "exact E" work-precision study can be seen on Figure 5.9. The plot again demonstrates the low error and order of accuracy convergence of the Boris-SDC compared to the standard Boris algorithm in terms of both actual time-step sizes used and computational expense. Here computational expense is calculated as the number of evaluations of the Lorentz force at the particle performed by a particle integrator during the course of the simulation. This is the number of evaluations of the right-hand side (RHS) of (2.25), i.e. number of times $\mathbf{f}$ is called during a simulation run. Standard Boris exhibits steady 2nd order convergence for the full run, with the 3 and 5 collocation node Boris-SDC solutions reaching 4th and 8th order convergence for $K = M$ SDC iterations until machine precision is reached. The results of the case thus provide further evidence that Boris-SDC exhibits favourable performance over classic Boris in terms of accuracy at a given number of RHS evaluations.

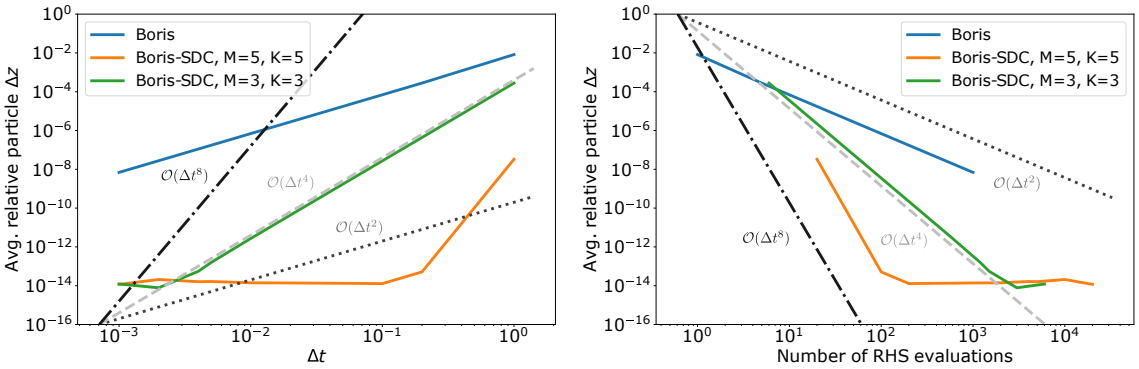

Figure 5.9: Convergence of position of non-linear oscillating particle in Type 1 experiment against Boris-SDC with $M = 8$ and $K = 8$ at 2000 time-steps. Performed for Boris and Boris-SDC in time-step range $1 \rightarrow 1000$. M denotes collocation nodes and K denotes SDC sweeps.

Interpol. E Case

The same convergence study performed for the "interpol. E" case demonstrates the error threshold due to the linear interpolation and the effect on accuracy and convergence order of the particle integrators (Figure 5.10). A minimum possible error floor in relative particle position between 10^{-2} and 10^{-12} can be observed for the range of mesh resolutions 10^1 to 10^6. Above the error threshold, decreasing time-step size improves accuracy at the order expected for each particle integrator. When the threshold is reached, improving the resolution of the time discretisation has no effect and the error saturates at the error of the interpolation method.

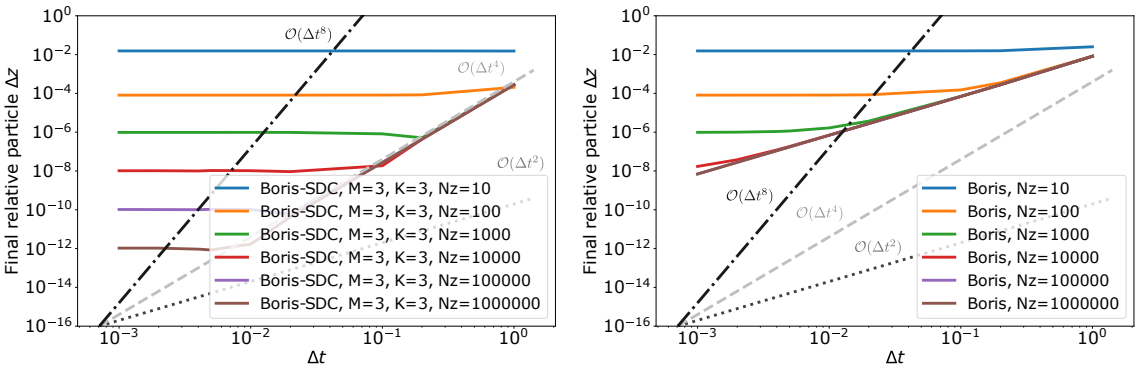

Figure 5.10: Convergence of position of oscillating particle in "interpol. E" experiment for Boris-SDC at $M = 3$, $K = 3$ (left) and standard Boris (right) at a range of cell resolutions ($10 \rightarrow 10^6$).

The error threshold is seen to decrease by two orders of magnitude when time-step size is reduced by one order of magnitude, which is consistent with the use of second-order accurate linear interpolation.

Approx. E Case

For the "approx. E" case, the addition of field solving on top of the field interpolation further raises the minimum error floor for a given spatial resolution. To emphasise the difference between this case and the previous, the results from each were plotted together (Figure 5.11) for both Boris and Boris-SDC. The error threshold is now such that improving the spatial resolution from NZ = 10^5 to NZ = 10^6 in Figure 5.10 only marginally lowers the error. This suggests the error saturation point lies around 10^{-10}, where the NZ = 10^6 simulations cease to converge. The diminishing returns at higher cell counts may be due to round-off error in the differentiation step ($\phi \rightarrow \mathbf{E}$) as numerical differentiation is ill-conditioned. Refer to Section 3.2 for the exact discretisation scheme used. As direct inversion was used to solve the linear scheme, solver tolerance is not believed to have had an effect.

The 4th order convergence expected from the Boris-SDC configuration used is still observed above the error threshold, and the integrator is more accurate than Boris at any given time-step, except for the lowest resolution NZ = 10 setup.

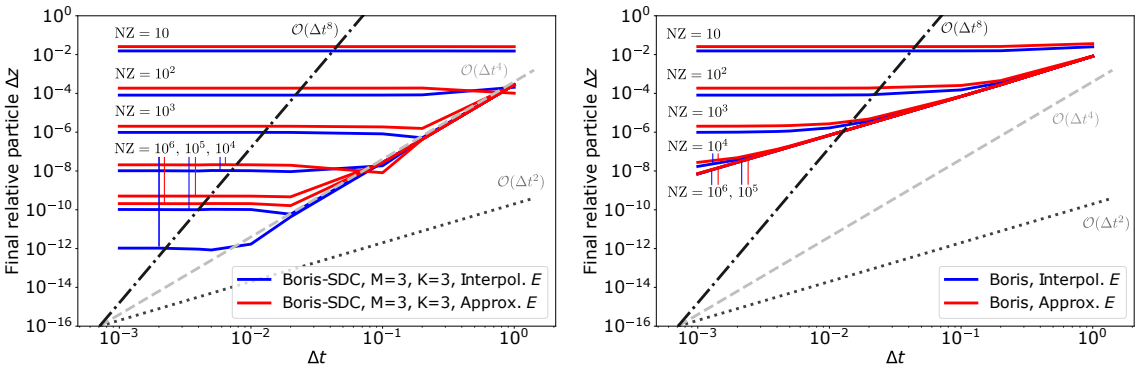

Figure 5.11: Convergence of position of oscillating particle in "approx. E" and "interpol. E" experiments for Boris-SDC at $M = 3$, $K = 3$ (left) and standard Boris (right) at a range of cell resolutions ($10 \rightarrow 10^6$).

ESPIC

Finally, when comparing the work-precision of the full ESPIC scheme to the "approx. E" case, much the same is observed as in the previous comparison. The error threshold for ESPIC is slightly worsened from "approx. E" by the addition of an extra interpolation step. Fortunately, the error saturation does not increase as much with the addition of the final interpolation step as for the addition of field solution.

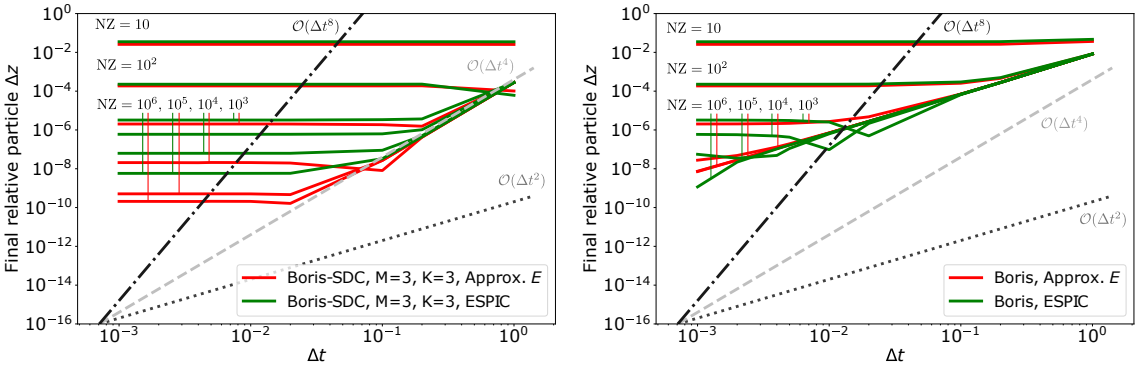

Figure 5.12: Convergence of position of non-linear oscillating particle in Type 3 and 4 experiments for Boris-SDC at $M = 3$, $K = 3$ (left) and standard Boris (right) at a range of cell resolutions ($10 \rightarrow 10^6$).

Overall, the error threshold at the most accurate solution for a given setup ($\mathrm{NZ} = 10^6$) increases from just above machine precision (10^{-14}) to around 10^{-8} as the transition is made from no-PIC to full ESPIC. The individual loss of accuracy for each component of the ESPIC scheme added was on the order 10^{-2} for the high resolution case, with little to no loss for the lowest resolution case. In summary, while the presence of field approximations limit the maximum accuracy achievable by Boris-SDC, it is still found to exhibit high-order convergence until the error saturation associated with spatial discretisation is reached.

5.2 Relativistic Particle Integration

5.2.1 Special Relativity

A relativistic generalisation of the Boris-SDC algorithm was sought to broaden the range of possible applications. A relativistic formulation was deemed particularly useful for EMPIC implementation, as charged particle tracking and EMPIC are often used to model particles or streams at relativistic velocities - see the discussion on $\mathbf{E} \times \mathbf{B}$ drift for relativistic particle integrators in Section 4.4 as well as studies on laser-plasma interaction such as Arber et al. [4]. The PIC code chosen for Boris-SDC implementation in EMPIC, Runko, is also inherently relativistic in its function definitions. No relativistic formulations were found to exist for velocity-Verlet nor Boris-SDC at the time of writing. The following section provides the derivation of relativistic versions of both velocity-Verlet and Boris-SDC, starting with a brief introduction to special relativity to provide the necessary context.

Following Griffiths [28], the original two postulates which form the basis of Einstein's first treatment read as follows:

1. The laws of physics apply in any inertial reference frame (the principle of relativity).

2. The vacuum speed of light is constant in all frames regardless of motion.

The reference frame defines a coordinate system in time and space (referred to henceforth as spacetime). Events inhabit a position $\mathbf{R} = (ct, \mathbf{r})$ in the system, where c is the speed of light and t, $\mathbf{r} = (x, y, z)$ are the coordinates in time and space respectively. Events can thus occur at different times and spatial positions depending on the reference frame.

The vector $\mathbf{R}$ is the "four-position" of an event or object w.r.t. the reference

frame. Using this, the "coordinate" velocity of an object can be defined as

$$\mathbf{v} = \frac{d\mathbf{r}}{dt}, \tag{5.17}$$

yielding the change in position with time of the object as observed from the reference frame.

Any object moving relative to the reference frame represents an inertial reference frame itself and for sufficiently large relative motion $\mathbf{v}$, different four-positions will be seen depending on the choice of frame (coordinate system reference frame or object frame). For instance, for the event of inhabiting a certain spatial position $\mathbf{r}_A$, an observer travelling with the object will report a different time τ to the time t measured by an observer stationary w.r.t. the reference frame. For the purpose of defining the differential equations of motion of a travelling object, time derivatives are generally taken using the proper time τ as it is invariant. Transforming between the two perspectives is done via the Lorentz factor γ, which essentially provides a simple multiplier for the dilation of time experienced between the two frames, with $dt = \gamma d\tau$. The Lorentz factor depends on relative velocity as

$$\gamma(\mathbf{v}) = \frac{1}{\sqrt{1 - \frac{|\mathbf{v}|^2}{c^2}}}. \tag{5.18}$$

For the relativistic Newton-Lorentz system treated in the following sections, the four-vectors will be defined from the perspective of the travelling objects (the particles) rather than the reference frame. The four-position of the particles then becomes

$$\mathbf{X} = (c\tau, \mathbf{x}) = (c\tau, x, y, z), \tag{5.19}$$

where $\mathbf{x}$ is the coordinate position of the particle w.r.t. the reference frame. The

associated four-velocity is then

$$\mathbf{U} = \frac{d\mathbf{X}}{d\tau} = (c, \mathbf{u}) = (c, u_x, u_y, u_z), \tag{5.20}$$

where $\mathbf{u}$ is the proper velocity, i.e. the change of position with time from the perspective of the particle.

Transforming between the coordinate velocity and proper/four-velocity is then performed by simple multiplication using the Lorentz factor

$$\mathbf{u} = \gamma \mathbf{v}. \tag{5.21}$$

From (5.18) and (5.21) it follows that the Lorentz factor can also be calculated using the proper velocity

$$\gamma(\mathbf{u}) = \sqrt{1 + \left(\frac{\mathbf{u}}{c}\right)^2}. \tag{5.22}$$

The effect of special relativity on the Newton-Lorentz system and the particle integration schemes under study can be found by applying (5.21) to Newton's Law

$$\mathbf{F} = m\frac{d\mathbf{u}}{dt} = m\frac{d(\gamma \mathbf{v})}{dt}. \tag{5.23}$$

The change in momentum now includes the relativistic factor, making the equation non-linear.

5.2.2 Relativistic Velocity-Verlet

As an alternative to leapfrog integration, velocity-Verlet can be defined for relativistic particle motion. In velocity-Verlet, both velocity and position are defined at the integer time-steps. Concurrent velocity/position can simplify simulation setup as well as data handling, since the half-step staggering of velocity need not be ac-

counted for. This is particularly important when comparing simulations of varying time-step.

Starting from relativistic Newton-Lorentz system, the coordinate velocity can be written

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} = \mathbf{g}(\mathbf{u}) = \frac{\mathbf{u}}{\gamma(\mathbf{u})}, \tag{5.24}$$

with

$$\frac{d\mathbf{u}}{dt} = \mathbf{F}(\mathbf{x}, \mathbf{g}(\mathbf{u})), \tag{5.25}$$

where

$$\mathbf{F}(\mathbf{x}, \mathbf{g}(\mathbf{u})) = \frac{q}{m}\left(\mathbf{E}(\mathbf{x}) + \mathbf{g}(\mathbf{u}) \times \mathbf{B}(\mathbf{x})\right) = \frac{q}{m}\left(\mathbf{E}(\mathbf{x}) + \frac{\mathbf{u}}{\gamma(\mathbf{u})} \times \mathbf{B}(\mathbf{x})\right). \tag{5.26}$$

In this form, the system is structured as a general second-order initial value problem [29]. The proper velocity $\mathbf{u}$ is now the independent variable of interest and coordinate velocity $\mathbf{v}$ is considered a derived quantity thereof for the purposes of time discretisation. To achieve second-order convergence accuracy like in leapfrog, central difference estimates of velocity and acceleration are applied and contracted to one time-step length giving

$$\mathbf{g}(\mathbf{u}^{n+1/2}) = \frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t} = \frac{\mathbf{u}^{n+1/2}}{\gamma(\mathbf{u}^{n+1/2})}, \tag{5.27}$$

$$\mathbf{F}\left(\mathbf{x}^{n+1/2}, \mathbf{g}(\mathbf{u}^{n+1/2})\right) = \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t}. \tag{5.28}$$

Rearranging these into update form for position and velocity yields

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{g}(\mathbf{u}^{n+1/2}), \tag{5.29}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{F}(\mathbf{x}^{n+1/2}, \mathbf{g}(\mathbf{u}^{n+1/2})). \tag{5.30}$$

The half-step velocity at $t^{n+1/2}$ in the position update can be estimated by half an explicit Euler step of (5.25)

$$\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\frac{1}{2}\Delta t} = \mathbf{F}(\mathbf{x}^n, \mathbf{g}(\mathbf{u}^n)), \tag{5.31}$$

which implies that

$$\mathbf{u}^{n+1/2} = \mathbf{u}^n + \frac{\Delta t}{2}\mathbf{F}(\mathbf{x}^n, \mathbf{g}(\mathbf{u}^n)). \tag{5.32}$$

The update equations for position and velocity are then

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{g}(\mathbf{u}^{n+1/2}) = \mathbf{x}^n + \Delta t \mathbf{g}\left(\mathbf{u}^n + \frac{\Delta t}{2}\mathbf{F}(\mathbf{x}^n, \mathbf{g}(\mathbf{u}^n))\right), \tag{5.33}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \frac{\Delta t}{2}\left(\mathbf{F}(\mathbf{x}^n, \mathbf{g}(\mathbf{u}^n)) + \mathbf{F}(\mathbf{x}^{n+1}, \mathbf{g}(\mathbf{u}^{n+1}))\right). \tag{5.34}$$

To solve the system (5.34) needs to be rearranged to fit the form (4.19) required for the Boris trick. Expanding the force terms gives

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \frac{\Delta t}{2}\frac{q}{m}\left(\mathbf{E}(\mathbf{x}^n) + \mathbf{g}(\mathbf{u}^n) \times \mathbf{B}(\mathbf{x}^n) + \mathbf{E}(\mathbf{x}^{n+1}) + \mathbf{g}(\mathbf{u}^{n+1}) \times \mathbf{B}(\mathbf{x}^{n+1})\right), \tag{5.35}$$

where the first cross-product involves only known quantities and can be calculated explicitly. Labelling the known cross-product as the constant

$$\mathbf{c}_1 = \frac{\Delta t}{2}\frac{q}{m}\mathbf{g}(\mathbf{u}^n) \times \mathbf{B}(\mathbf{x}^n), \tag{5.36}$$

and using the shorthands

$$E := \frac{q}{m}\frac{\mathbf{E}(\mathbf{x}^n) + \mathbf{E}(\mathbf{x}^{n+1})}{2}, \tag{5.37}$$

$$B := \frac{q}{m}\mathbf{B}(\mathbf{x}^{n+1}), \tag{5.38}$$

allows the velocity update to be written

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, E + \frac{\Delta t}{2}\frac{\mathbf{u}^{n+1}}{\gamma(\mathbf{u}^{n+1})} \times B + \mathbf{c}_1, \tag{5.39}$$

where the function $\mathbf{g}(\mathbf{u}^{n+1})$ has been expanded. To put the expression into exactly the right form for solution by Algorithm 1, the relativistic term $\gamma(\mathbf{u}^{n+1})$ is combined with the time-step and the term $\mathbf{u}^n \times B$ is both added and subtracted from the cross-product

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, E + \frac{\Delta t}{2\gamma(\mathbf{u}^{n+1})}(\mathbf{u}^{n+1} \times B + \mathbf{u}^n \times B - \mathbf{u}^n \times B) + \mathbf{c}_1, \tag{5.40}$$

which is equivalent to adding 0. The negative term can now be added to the constant, yielding

$$\begin{aligned} \mathbf{c}_k &= \mathbf{c}_1 - \frac{\Delta t}{2\gamma(\mathbf{u}^{n+1})}\mathbf{u}^n \times B \\ &= \frac{q}{m}\frac{\Delta t}{2\gamma(\mathbf{u}^n)}\mathbf{u}^n \times \mathbf{B}(\mathbf{x}^n) - \frac{q}{m}\frac{\Delta t}{2\gamma(\mathbf{u}^{n+1})}\mathbf{u}^n \times \mathbf{B}(\mathbf{x}^{n+1}). \end{aligned} \tag{5.41}$$

Finally, the positive $\mathbf{u}^n \times B$ term can be combined with the existing cross-product to give a velocity average,

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, E + \frac{\Delta t}{\gamma(\mathbf{u}^{n+1})}\frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2} \times B + \mathbf{c}_k, \tag{5.42}$$

resulting in the correct form for the Boris algorithm, provided $\gamma(\mathbf{u}^{n+1})$ can be calculated or approximated. In the relativistic version of the classic Boris leapfrog integrator, the velocity average is approximated by the expression

$$\bar{v} = \frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2\gamma^{n+1/2}}, \tag{5.43}$$

where $\gamma^{n+1/2} = \gamma(\mathbf{u}^-) = \gamma(\mathbf{u}^+)$. Recall that the plus and minus subscripted velocities indicate the velocities pre and post magnetic rotation, with the former calcu-

lated by performing one half of the electric acceleration. This approximation is thus built on the observation that the magnitude of velocity in invariant under magnetic rotation and by (5.22), so is the relativistic factor.

Using the same approximation for the velocity average here, the velocity update takes the form

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, E + \frac{\Delta t}{\gamma^{n+1/2}} \frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2} \times B + \mathbf{c}_k, \tag{5.44}$$

and can be solved using Algorithm 1 by first calculating

$$\gamma^{n+1/2} = \sqrt{1 + \left(\frac{\mathbf{u}^n + \Delta t\, E/2}{c} \right)^2}. \tag{5.45}$$

5.2.3 Relativistic Boris-SDC

A complete re-derivation is unnecessary to extend the Boris-SDC algorithm to the relativistic regime. Taking the 2019 formulation update equations (4.46)-(4.50) from Section 4.3.4 and the relativistic Newton-Lorentz system (5.24)-(5.25), the Boris-SDC position update can be made relativistic by simple substitution of $\mathbf{v}$ and $\mathbf{g}(\mathbf{u})$, yielding

$$\mathbf{x}_{m+1}^{k+1} = \mathbf{x}_m^{k+1} + \Delta\tau_m \left[\mathbf{g}(\mathbf{u}_{m+1/2}^{k+1}) - \mathbf{g}(\mathbf{u}_{m+1/2}^{k}) \right] + \mathbf{IV}_m^k \tag{5.46}$$

where

$$\mathbf{u}_{m+1/2}^{k+1} = \mathbf{u}_m^{k+1} + \frac{\Delta\tau_m}{2} \mathbf{F}\left(\mathbf{x}_m^{k+1}, \mathbf{g}(\mathbf{u}_m^{k+1}) \right), \tag{5.47}$$

$$\mathbf{IV}_m^k := \sum_{j=1}^{M} s_{m,j} \mathbf{g}(\mathbf{u}_j^k). \tag{5.48}$$

The form of the implicit velocity equation is similarly unaffected

$$\begin{aligned} \mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} &+ \frac{\Delta\tau_m}{2} \left[\mathbf{F}(\mathbf{x}_m^{k+1}, \mathbf{u}_m^{k+1}) + \mathbf{F}(\mathbf{x}_{m+1}^{k+1}, \mathbf{u}_{m+1}^{k+1}) \right] \\ &- \frac{\Delta\tau_m}{2} \left[\mathbf{F}(\mathbf{x}_m^{k}, \mathbf{u}_m^{k}) + \mathbf{F}(\mathbf{x}_{m+1}^{k}, \mathbf{u}_{m+1}^{k}) \right] + \mathbf{IF}_m^k, \end{aligned} \tag{5.49}$$

$$\mathbf{IF}_m^k := \sum_{j=1}^{M} s_{m,j}\mathbf{F}(\mathbf{x}_j^k, \mathbf{u}_j^k), \tag{5.50}$$

with $\mathbf{u}$ now being the unknown and the force term now including an evaluation of $\mathbf{g}$. The velocity update must be rewritten to apply the Boris algorithm, starting by expanding the force term to define the constant

$$\mathbf{c}_1 = \frac{q}{m}\frac{\Delta\tau_m}{2}\mathbf{g}(\mathbf{u}_m^{k+1}) \times \mathbf{B}(\mathbf{x}_m^{k+1}) - \frac{\Delta\tau_m}{2}\left[\mathbf{F}(\mathbf{x}_m^k, \mathbf{u}_m^k) + \mathbf{F}(\mathbf{x}_{m+1}^k, \mathbf{u}_{m+1}^k)\right] + \mathbf{IF}_m^k \tag{5.51}$$

giving the new update form

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m \frac{q}{m}\left(\frac{\mathbf{E}(\mathbf{x}_m^{k+1}) + \mathbf{E}(\mathbf{x}_{m+1}^{k+1})}{2} + \frac{1}{2}\mathbf{g}(\mathbf{u}_{m+1}^{k+1}) \times \mathbf{B}(\mathbf{x}_{m+1}^{k+1})\right) + \mathbf{c}_1. \tag{5.52}$$

Using the shorthands

$$E := \frac{q}{m}\frac{\mathbf{E}(\mathbf{x}_m^{k+1}) + \mathbf{E}(\mathbf{x}_{m+1}^{k+1})}{2}, \tag{5.53}$$

$$B := \frac{q}{m}\mathbf{B}(\mathbf{x}_{m+1}^{k+1}), \tag{5.54}$$

the SDC update can be written

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m E + \frac{\Delta\tau_m}{2}\frac{\mathbf{u}_{m+1}^{k+1}}{\gamma(\mathbf{u}_{m+1}^{k+1})} \times B + \mathbf{c}_1. \tag{5.55}$$

To close the equation, the relativistic factor $\gamma(\mathbf{u}_{m+1}^{k+1})$ must be calculated or estimated like in the classic Boris algorithm as seen in Section 5.2.2. Recall how γ is typically calculated in Boris from the pre-rotation velocity $\gamma(\mathbf{u}^-)$ and included in the β parameter of Algorithm 1. This works fine for classic Boris, but as $\mathbf{u}^-$ is not tied to the SDC iteration it will not converge for increasing k if applied to Boris-SDC. The approximation thus contains some constant error δ w.r.t. the desired quantity $\mathbf{u}_{m+1}^{k+1}$, giving

$$\left\| \frac{\mathbf{u}_{m+1}^{k+1}}{\gamma(\mathbf{u}^-)} - \frac{\mathbf{u}_{m+1}^{k+1}}{\gamma(\mathbf{u}_{m+1}^{k+1})} \right\| \geq \delta > 0. \tag{5.56}$$

Here δ depends on time-step size and how relativistic the problem is. The quantity equals zero in the non-relativistic case, thus allowing the approximation to be used in the non-relativistic regime as it was in Section 4.3.

A different treatment of γ is required to close the system and produce a new relativistic Boris-SDC formulation that retains the high-order convergence of the non-relativistic scheme. The iterative nature of SDC can be leveraged to provide an approximation $\gamma(\mathbf{u})$ at the next node τ_{m+1} by using the value computed at the node in the previous iteration k. Using as an estimate the relativistic factor at τ_{m+1} for the previous iteration $\gamma := \gamma(\mathbf{u}_{m+1}^k) \approx \gamma(\mathbf{u}_{m+1}^{k+1})$ allows γ to converge with each iterative correction to $\mathbf{u}$. This guarantees convergence as

$$\left\|\gamma(\mathbf{u}_{m+1}^{k+1}) - \gamma(\mathbf{u}_{m+1}^k)\right\| \to 0 \quad \text{as} \quad \left\|\mathbf{u}_{m+1}^k - \mathbf{u}_{m+1}^{k+1}\right\| \to 0, \tag{5.57}$$

allowing the relativistic factor to reach the correct value at τ_{m+1} as k increases.

Some additional arithmetic is required to put the update into exactly the right form to be solved by the Boris algorithm. Letting $\gamma := \gamma(\mathbf{u}_m^k)$ while adding and subtracting the same term from (5.58), equivalent to adding 0, gives

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m E + \frac{\Delta\tau_m}{2\gamma}\mathbf{u}_{m+1}^{k+1} \times B + \frac{\Delta\tau_m}{2\gamma}\left(\mathbf{u}_m^{k+1} \times B - \mathbf{u}_m^{k+1} \times B\right) + \mathbf{c}_1, \tag{5.58}$$

where one of the addition terms can be incorporated into the integral constant

$$\mathbf{c}_k = \mathbf{c}_1 - \frac{\Delta\tau_m}{2\gamma}\mathbf{u}_m^{k+1} \times B, \tag{5.59}$$

allowing the update equation to be written in the correct form (4.11) for solution by Boris' trick (Algorithm 1):

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m E + \Delta\tau_m \frac{\mathbf{u}_m^{k+1} + \mathbf{u}_{m+1}^{k+1}}{2\gamma} \times B + \mathbf{c}_k. \tag{5.60}$$

Initialisation options are the same as for non-relativistic SDC desribed in Section 4.3.2, i.e. using either a simple copythrough of the initial value or applying a predictor step.

5.2.4 Relativistic Penning Trap Test

To verify the relativistic Velocity-Verlet and Boris-SDC formulations derived in Sections 5.2.2 and 5.2.3, the integrator schemes were implemented in a simple test code for the Penning trap setup outlined in Table 5.1. Relativistic effects were achieved by changing the speed of light c rather than the initial velocity of the particle, which was set to a proper velocity of $\mathbf{u} = (100, 0, 100)^{\mathrm{T}}$. The fields were once again calculated exactly at the particle positions using the values from Section 5.1.3. With this methodology, the following four test cases seen in Table 5.3 were studied. Ratio of velocity to lightspeed $\beta = |\mathbf{v}|/c$ has been included for an intuitive idea of particle velocity.

Table 5.3: Relativistic test cases

Setup	c	γ	β	$\lvert\mathbf{v}\rvert$
A	441.726	1.05	0.304911	134.687
B	126.491	1.5	0.745356	94.2809
C	9.44911	15	0.997775	9.42809
D	0.942830	150	0.999978	0.942809

Figures 5.13 - 5.16 present the work-precision results of the relativistic test study. Relative position was used again as the error measure and calculated at the final test time with an 8th order collocation solution using 1000 time-steps as the reference. Predictor initialisation was used for all Boris-SDC cases in the test.

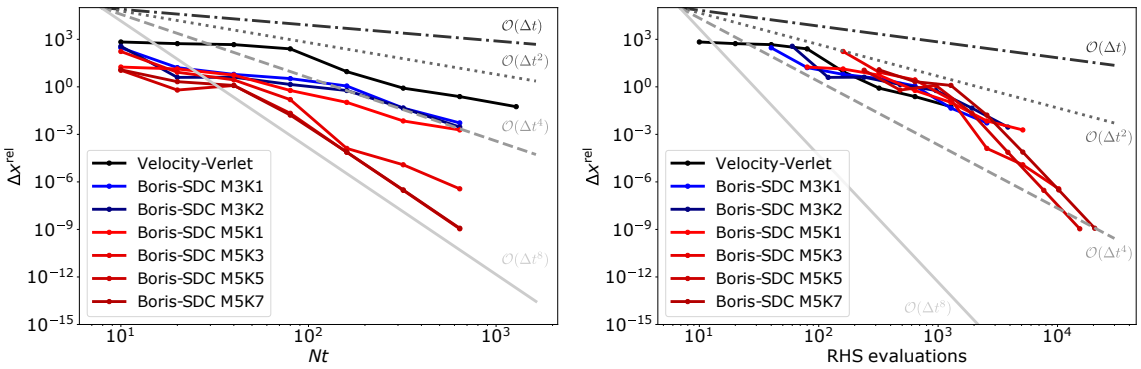

Figure 5.13: Convergence of relative position of relativistic single particle in a Penning trap, test case A: Particle at 30% of light speed.

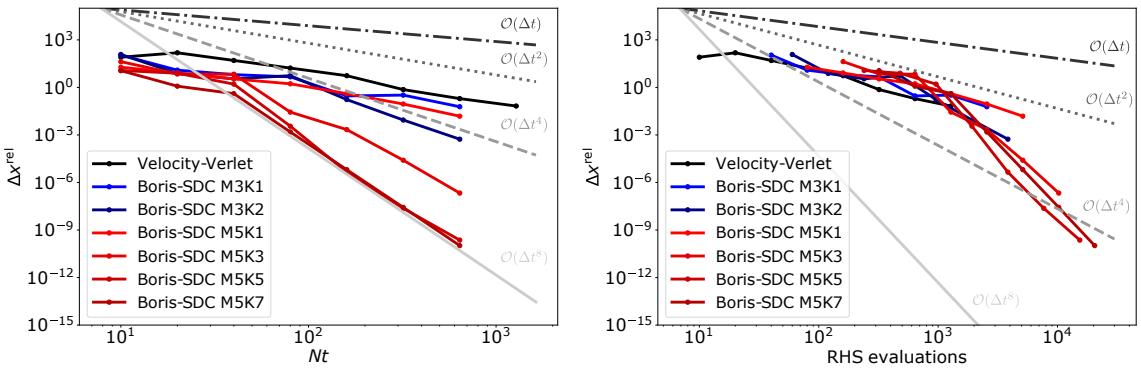

Figure 5.14: Convergence of relative position of relativistic single particle in a Penning trap, test case B: Particle at 75% of light speed.

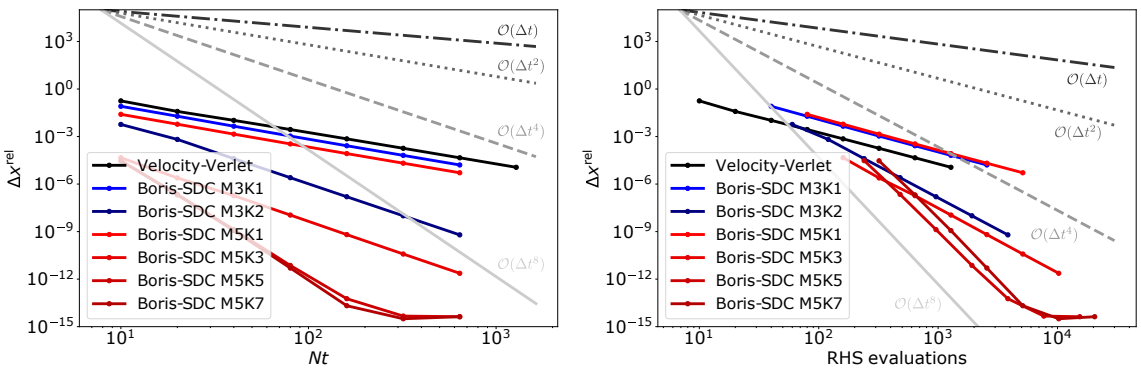

Figure 5.15: Convergence of relative position of relativistic single particle in a Penning trap, test case C: Particle at 99.78% of light speed.

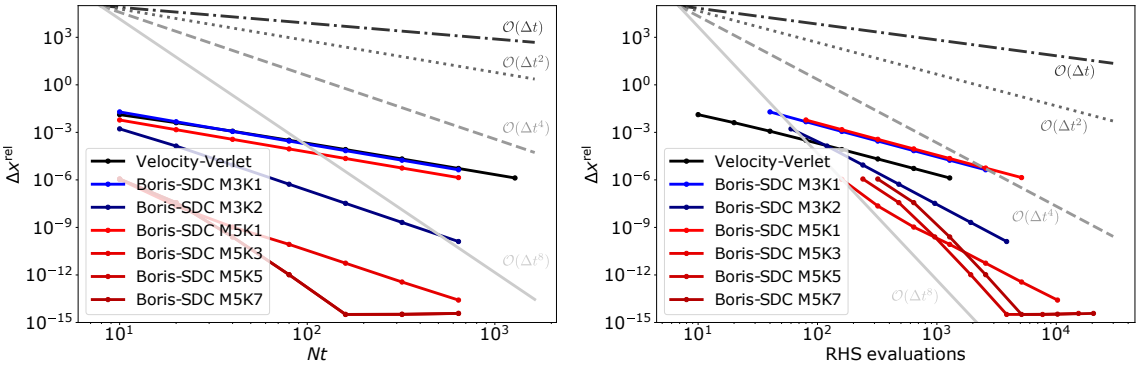

Figure 5.16: Convergence of relative position of relativistic single particle in a Penning trap, test case D: Particle at 99.9978% of light speed.

The high-order convergence with time-step of Boris-SDC was seen to be maintained even in highly relativistic test cases. Plotting the relative error against function evaluations for a given integrator setup as in previous tests, Boris-SDC still exhibits superior performance to velocity-Verlet at all values of c. Note that the seeming increase in performance with the more extreme relativistic cases is due to proper velocity being kept constant. As c is decreased while proper velocity is kept static, the coordinate velocity of the particles decreases accordingly. As a result the trajectories (Figure 5.17) in test case A are much longer and more complicated than test case D, the dynamics are effectively more coarsely resolved by a given time-step size for the higher coordinate velocities.

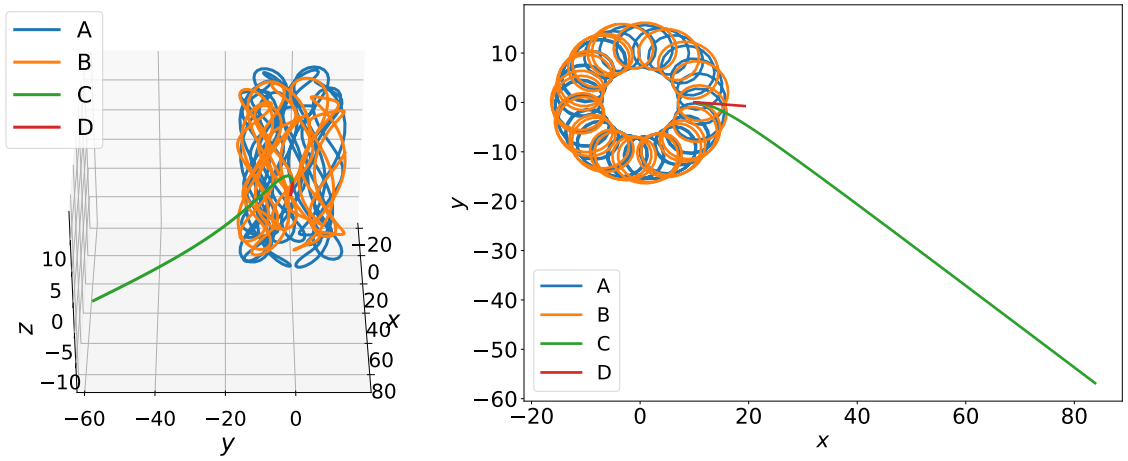

Figure 5.17: Convergence of relative position of relativistic single particle in a Penning trap, test case D: Particle at 99.9978% of light speed.

The test cases studied here show that the relativistic formulations of velocity-Verlet and Boris-SDC work as intended, as they are consistent with one another as well as consistent with the relativistic Newton-Lorentz system as solved by a high accuracy collocation solution.

5.2.5 Comparison with Boris Alternatives

In Section 4.4, the problem of unphysical drift produced by the Boris algorithm when $\mathbf{E}$ and $\mathbf{v}\times\mathbf{B}$ forces are close to equal was discussed briefly along with proposed alternatives to Boris for solving this issue (Vay and Higuera-Cary integrators). The performance of relativistic Boris-SDC under these conditions is unknown and affect the relevance of the integrator for use in ultra-relativistic beam simulations [82] and any other case of closely balanced electric/magnetic forces. In this section, the ability of Boris-SDC to accurately capture the dynamics of the force balanced scenario is discussed and tested. Comparison between existing integrators has already been made in a review by Ripperda et al. [65], where Boris, Vay and Higuera-Cary are tested for four single-particle problems. The most pertinent of these is the force-free test, in which a single particle drifts in uniform electric and magnetic fields precisely calibrated so that the sum of electric and magnetic contributions to the Lorentz force is zero. The particle should thus continue to drift at the same velocity, but for most simulations won't since any error in the particle velocity update will accumulate, however slight. The test is very stringent and even Vay/Higuera-Cary integrators produce drift under these conditions due to computational round-off error. The relativistic Boris algorithm however produces exceedingly high drift beyond that attributable to round-off error due to the way the implicit velocity term in the Lorentz force is handled. Taking the velocity update for time-synchronised leapfrog Boris provided by [65]

$$\frac{u^{n+1}-u^n}{\Delta t} = \frac{q}{m}\left(\mathbf{E}(\mathbf{x}^{n+1/2}) + \mathbf{v}^{n+1/2}\times\mathbf{B}(\mathbf{x}^{n+1/2})\right) \tag{5.61}$$

where $\mathbf{v}^{n+1/2}$ is the average proper velocity over the time-step and recalling that $\mathbf{u}^n = \gamma^n \mathbf{v}^n$ with γ^n being the relativistic factor at time-step n. This equation cannot be solved immediately using the classic (non-relativistic) Boris algorithm from Section 4.2, but requires an approximation for the average velocity which becomes a choice in how to handle the relativistic factor. Recall that in the relativistic form of classic Boris, the velocity average chosen is

$$\mathbf{v}^{n+1/2} = \frac{u^{n+1} + u^n}{2\gamma^{n+1/2}}, \tag{5.62}$$

where $\gamma^{n+1/2} = \gamma^- = \gamma^+$, i.e. the relativistic factor is constant during the rotation step of the algorithm. The factor can thus be calculated directly from the proper velocity $\mathbf{u}^-$ obtained from the explicit electric acceleration half-step.

For such a velocity average, Vay [82] shows that the force is only balanced correctly (requiring $\mathbf{E} + \mathbf{v}^n \times \mathbf{B} = \mathbf{E} + \mathbf{v}^{n+1} \times \mathbf{B}$ and $\gamma^n \mathbf{v}^n = \gamma^{n+1}\mathbf{v}^{n+1}$ in the force-free scenario) when $\mathbf{E}^{n+1/2} = \mathbf{B}^{n+1/2} = 0$. In the general case with non-zero fields, a spurious acceleration results.

The integrator proposed by [82] fixes this by using the velocity average

$$\mathbf{v}^{n+1/2} = \frac{\mathbf{u}^{n+1}/\gamma^{n+1} + \mathbf{u}^n/\gamma^n}{2}, \tag{5.63}$$

and modifying the Boris algorithm to include explicit solution for γ^{n+1}.

This velocity average is the same used in deriving the relativistic Boris-SDC integrator, see Section 5.2.3. Recall however, that the correct value is obtained through SDC iteration rather than direct calculation. Regardless of the iterative method used, the form of the Lorentz force used in the two integrators is the same and Boris-SDC is expected to produce similar drift results to Vay.

In the following two sections, relativistic Boris-SDC is tested against velocity-

Verlet and Vay for both the relativistic gyration and force-free test cases from [65]. The former case has been included to verify the Vay integrator implementation as well as providing some insight into comparative performance and stability of the three integrators.

Relativistic Larmor gyration

Following [65], the motion of a single particle undergoing relativistic gyration in a uniform magnetic field is defined by Larmor radius, angular rate, period and phase. The Larmor gyration radius in a relativistic setting is given by

$$R_c = \frac{\gamma m v_\perp}{q B_z}, \tag{5.64}$$

where $v_\perp$ is the coordinate velocity perpendicular to the gyration circle and γ the associated relativistic factor. For a gyration velocity in the xy-plane, the associated uniform magnetic field must be oriented along the z-axis, denoted here by B_z. For this Larmor radius, the associated angular rate is given by

$$\omega_c = \frac{q B_z}{m\gamma}, \tag{5.65}$$

where ω_c is also called the cyclotron frequency. This expression gives a gyration period of

$$T_c = 2\pi \frac{m\gamma}{q B_z}, \tag{5.66}$$

as well as a function of gyration phase vs. time given by

$$\theta_c(t) = -\omega_c t, \tag{5.67}$$

here defined for the frame wherein the particle begins gyration at the zero angle of the unit circle with the gyration velocity pointed along the negative y-axis at $t = 0$. For this setup, the analytical time-evolution in position (x, y components as the

Table 5.4: Gyration test parameters

Parameter	Value
c	1
q/m	1
γ	10^6
B_z	10^6
$v_\perp$	$\sqrt{c^2 \cdot (1 - \frac{1}{\gamma^2})}$

magnetic field axis is in z) becomes

$$\mathbf{x}(t) = [R_c \cos(-\omega_c t), R_c \sin(-\omega_c t), 0]. \tag{5.68}$$

The physical parameters for the gyration test case were set to equal those of [65], summarised in Table 5.4.

Note that the calculation for perpendicular coordinate velocity is determined from simple rearranging of the equation for relativistic factor as a function of velocity (5.18).

The test case was simulated using the Vay, velocity-Verlet and Boris-SDC integrators, as well as by implicit collocation (4.25) solved using iterative methods from the Python package SciPy [89] instead of SDC iteration. The simulations were run for 100 gyration periods (T_c) at a time-step of $\Delta t = T_c/100 = 2\pi/100$ and error calculated as relative difference with respect to the analytical values (5.68) and $v_\perp$ at any given point in time.

Figure 5.18 show the comparative resulting time-evolution for position and velocity error

The long-term stability of each integrator is in-line with expectations, with the second-order velocity-Verlet and Vay methods producing much higher phase error than both fourth and eighth order Boris-SDC. In terms of velocity error, Vay and velocity-Verlet capture the rotation exactly while Boris-SDC converges to a level

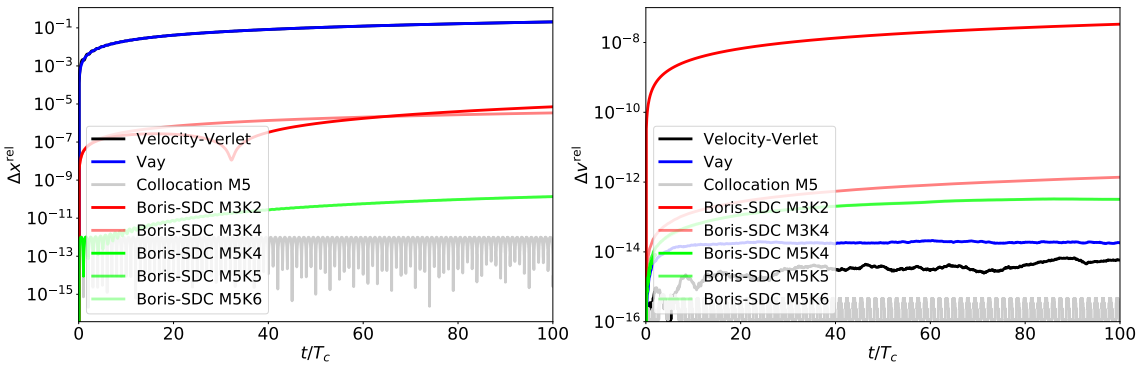

Figure 5.18: Relative error in position and relativistic factor for 100 periods of gyration for Boris-SDC, collocation, velocity-Verlet and Vay integration. Note: The Vay line covers velocity-Verlet and all $M = 5$ node Boris-SDC solutions are also almost identical.

close to machine precision as the number of SDC sweeps are increased. Boris-SDC with $M = 5$ nodes does not completely converge on the underlying 5-node collocation solution, but in this case the results are close enough to machine precision for the difference to attributable to the additional arithmetic required by Boris-SDC.

Force-free condition drift

The force-free test case studied by [65] has a single particle travelling at some initial velocity along the y-axis v_y in a uniform magnetic field. For a magnetic field oriented along the z-axis, the arising $\mathbf{v} \times \mathbf{B}$ force will be entirely in x and can be cancelled out entirely by an appropriately scaled electric force given by E_x. The Lorentz force is then

$$\mathbf{F} = \mathbf{E} + \mathbf{v} \times \mathbf{B} = 0, \tag{5.69}$$

and the particle should continue to drift at constant velocity, unaffected on the whole by the electromagnetic fields.

For a particle with initial velocity entirely along the y-axis in a magnetic field oriented along the z-axis, the uniform electric field required to balance the force

must be entirely in the x-direction. The electric, magnetic and velocity terms are then related by

$$\mathbf{E} = -\mathbf{v} \times \mathbf{B} = \begin{pmatrix} v_y B_z \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} E_x \\ 0 \\ 0 \end{pmatrix}. \tag{5.70}$$

By choosing two out of three terms, the third can now be calculated to provide a balanced pair of forces giving the required force-free setup. The physical parameters used for the force-free test simulations are summarised in Table 5.5, again set to mirror the original study.

Table 5.5: Force-free test parameters

Parameter	Value
c	1
q/m	1
γ	10^6
B_z	1
v_y	$\sqrt{c^2 \cdot (1 - \frac{1}{\gamma^2})}$
E_x	$-v_x B_z$

The setup was studied using Vay, velocity-Verlet, Boris-SDC and collocation using both $M = 3$ and $M = 5$ node configurations for the two latter integrators. As the unphysical drift in x and v_x are initially very small, the simulations were run to a time of $t_E = 10^5$ to allow the error to properly accumulate. A range of time-steps were studied, including but not limited to those used by [65].

Figure 5.19 shows the comparative time-evolution of unphysical drift along the x-axis at a time-step of 0.01, where all integrators produce drifts of similar magnitude, making them visible on a linearly scaled plot.

At this and higher time-resolutions, the drift produced by velocity-Verlet appears to be close to that of Vay, with Boris-SDC producing identical drift to Boris when

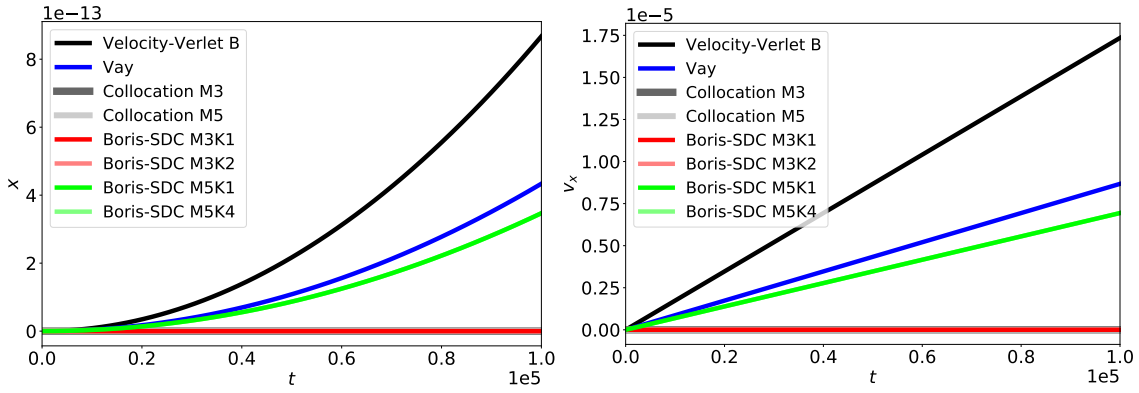

Figure 5.19: Linear scale position and velocity in the force-free case for Boris-SDC, collocation, velocity-Verlet and Vay integration using a time-step of 0.1. This corresponds to 10^6 time-steps to reach the final time of $t = 10^5$.

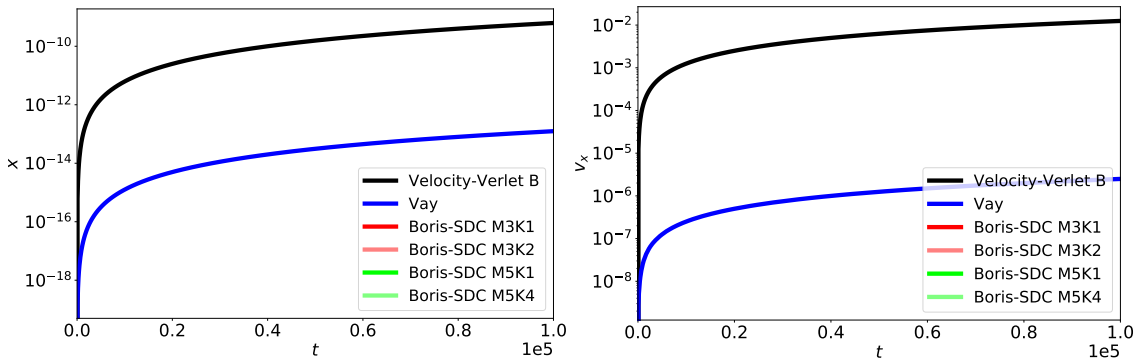

Figure 5.20: Position and velocity in the force-free case for Boris-SDC, collocation, velocity-Verlet and Vay integration using a time-step of 1. Note that the Boris-SDC simulations produced no drift and so not visible on the logarithmically scaled plot.

using 3 collocation nodes. The pure collocation results were calculated iteratively using SciPy [89] and produced no drift in any test. While comparable drift is seen for all 3 explicit integrators at this time-step size, neither Vay nor Boris-SDC suffer from the inaccurate Lorentz force cancellation found in Boris, and all drift should be caused by accumulated round-off error. The drift being caused by the limitations of machine precision also serve to explain why the Boris-SDC results are nearly identical regardless of the number of SDC sweeps used. The additional drift produced by velocity averaging in velocity-Verlet increases rapidly with time-step, becoming orders of magnitude greater at the coarsest time-step of 1 (Figure 5.20).

Note that where the line for an integrator is not visible, zero drift was produced by this integrator at this time-step. The occasional zero-drift cases, which were seen for

Vay as well as both configurations of Boris-SDC, probably arise from particularly fortuitous sets of input parameters which avoid floating point number rounding. Both decimal $(0.1, 0.01, ...)$ and fractional $(1, 1/2, 1/4, ...)$ time-step sizes were used in the following results. This was to see if the zero-drift cases were consistently linked to the type of time-step size used, as decimals are inexactly represented in floating point arithmetic (and as binary numbers generally). Unlike decimal numbers, fractions defined by $1/2^n$ where $n = 0, 1, 2, 3, ...$ can be exactly defined in base-2 number systems like binary. The change in drift with time-step for all integrators can be seen below in Figure 5.21, which cover a range of time-steps sizes from 1 down to 0.001. As the collocation solution produced zero drift in all tested cases and the Boris-SDC results did not change for sweep counts higher than $K = 1$, these configurations were omitted.

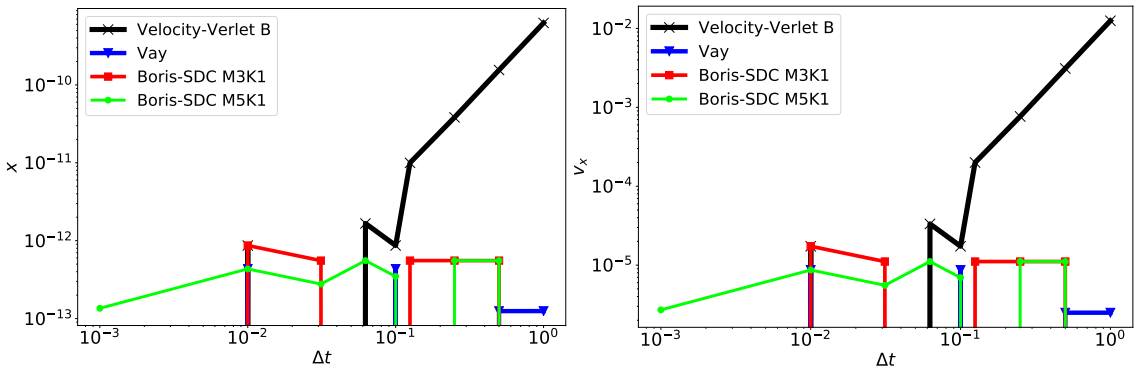

Figure 5.21: Position and velocity in the force-free case for Boris-SDC, velocity-Verlet and Vay integration for a range of time-steps $\Delta t = [1, 0.5, 0.25, 0.125, 0.1, 0.0625, 0.03125, 0.01, 0.001]$. Note again that where data points drop below the x-axis, these points are zero.

For all time-steps, the drift in x-velocity produced by both Vay and Boris-SDC stays around 10^{-5} or below, remaining nearly constant regardless of time-step. Velocity-Verlet however, can be seen to produce a drift several orders of magnitude higher for the coarsest time-step sizes. This drift then decreases with time-step size until the three integrators produce roughly equal drift due to round-off error. No clear link was found between zero-drift cases and time-step size. Despite using the same average velocity term in the integration of the Lorentz force, Boris-SDC

does not consistently produce exactly the same results as Vay. While drift is seen to be either 0, the same as Vay or slightly higher, the integrators remain within an order of magnitude where drift is seen. Additional drift can be attributed to the additional computation required in Boris-SDC, leading to increased round-off error, the main cause of drift for these two integrators.

5.3 EMPIC Code: Runko

5.3.1 Introduction

To study the characteristics and performance of Boris-SDC applied to a fully electromagnetic PIC code, an existing EMPIC code was sought for Boris-SDC implementation. The open-source EMPIC code "Runko" introduced by Nättilä [57] was chosen due to its modular design and collaborative ties to the main contributor.

The main design features of Runko include the use of C++ data containers and solvers wrapped in a Python interface, allowing their use in the "driver" scripts which are also written in Python. The drivers contain the main structure of a simulation, bringing the disparate solvers and modules together and include the main time-loop. While Runko was designed with the typical flow of PIC operations in mind (see discussion in Section 5.1.2), the design of the driver script and modularity of the code allows modification to fit the needs of velocity-Verlet and Boris-SDC integrators.

One of the main differences between Runko and the codes presented in the above sections is a particular scaling of the system quantities, yielding a unique unit system which is designed to reduce floating point operations[57]. In practice, the resolution of particle time integration is controlled via the Courant-Friedrichs-Lewy (CFL) number, combining the grid spacing and time-step size into one term. Boris-SDC and velocity-Verlet must be converted to this unit system before implementation with Runko, see Sections 5.3.2 and 5.3.3. The unit system details and conversions

are included here due to the normalisation and cancellation of the time-step term, which presents a problem on two counts: First, as Boris-SDC operates via the use of time sub-steps, it is not immediately clear how the scheme will appear in the new unit system. Secondly, even when Boris-SDC has been properly converted, the work-precision studies used to gauge performance rely on the ability to explicitly control the time-step size without changing the fundamental physics.

The remainder of this section presents outlines the unit system and the required changes to the field and particle update equations of EMPIC. Rather than the SI units used in this Thesis so far, Runko uses CGS (Gaussian) units. The choice does not affect the fundamentals of the EMPIC scheme, but does change the constants of the arithmetic presented in the remainder of this section. The Maxwell equations in CGS units are

$$\nabla \cdot \mathbf{E} = 4\pi\rho, \tag{5.71}$$

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi\mathbf{J}, \tag{5.72}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{5.73}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}, \tag{5.74}$$

with the corresponding Lorentz acceleration

$$\frac{d\mathbf{u}}{dt} = \frac{q}{m}\left(\mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B}\right). \tag{5.75}$$

The unit system in Runko follows the convention used in the PIC code TRISTAN-MP. The variables are normalised w.r.t. appropriate fiducial values. The most noteworthy scalings are those for time and distance, which are normalised w.r.t. the grid spacing $\mathbf{x} = \hat{\mathbf{x}}\Delta x$ and time-step $t = \hat{t}\Delta t$, where the hat superscript denotes quantities in Runko units.

As velocity is the derivative of time with distance, it follows that the system

velocity is expressed in terms of the spatial and temporal step sizes. The coordinate velocity is given by

$$\mathbf{v} = \frac{d\mathbf{x}}{dt} = \frac{d\hat{\mathbf{x}}}{d\hat{t}}\frac{\Delta x}{\Delta t} = \hat{\mathbf{v}}\frac{\Delta x}{\Delta t}, \tag{5.76}$$

with the corresponding proper velocity

$$\mathbf{u} = \hat{\mathbf{u}}\frac{\Delta x}{\Delta t} = \hat{\mathbf{v}}\gamma(\mathbf{v})\frac{\Delta x}{\Delta t}. \tag{5.77}$$

To maintain stability and ensure the fidelity of the simulation, the fraction $\Delta x/\Delta t$ is set to equal c normalised so that $\Delta x/\Delta t = c/\hat{c}$. Here $\hat{c}$ is then both the CFL number and the speed of light in Runko units. The maximum cell length a particle with velocity near c can travel in a single time-step is now easily limited through the single parameter $\hat{c}$, and the CFL condition can be enforced via this parameter.

The fields are scaled via the fiducial value B_0, so $\mathbf{E} = \hat{\mathbf{E}}B_0$ and $\mathbf{B} = \hat{\mathbf{B}}B_0$. Charge, mass and current are similarly scaled, giving $q = \hat{q}q_0$, $m = \hat{m}m_0$ and $\mathbf{J} = \hat{\mathbf{J}}J_0$ where all numerical variables are denoted with a hat.

Given a set of scaling factors (B_0, J_0, etc.) these conventions are applied to equations (5.74), (5.72) and (5.75), to form the system of update equations in Runko units. With careful choice of the scaling factors B_0 and J_0, the update equations can be made effectively unitless. Defining the scaling factors

$$B_0 := \frac{m_0 c}{q_0 \hat{c}\Delta t} = \frac{m_0 c}{q_0 \hat{c}}\left(\frac{c}{\hat{c}\Delta x}\right), \tag{5.78}$$

and

$$\frac{J_0}{B_0} := \frac{1}{4\pi\Delta t}, \tag{5.79}$$

the discretised update equations can be established in Runko units as

$$\Delta[\hat{\mathbf{E}}]_t = c\frac{\Delta t}{\Delta x}\Delta[\hat{\mathbf{B}}]_x - \frac{4\pi J_0}{B_0}\Delta t\hat{\mathbf{J}} = \hat{c}\Delta[\hat{\mathbf{B}}]_x - \hat{\mathbf{J}}, \tag{5.80}$$

$$\Delta[\hat{\mathbf{B}}]_t = -c\frac{\Delta t}{\Delta x}\Delta[\hat{\mathbf{E}}]_x = -\hat{c}\Delta[\hat{\mathbf{E}}]_x, \tag{5.81}$$

$$\Delta[\hat{\mathbf{u}}]_t = \frac{\hat{q}q_0\hat{c}}{\hat{m}m_0c}B_0\Delta t\left(\hat{\mathbf{E}} + \frac{\hat{\mathbf{v}}c}{c\hat{c}} \times \hat{\mathbf{B}}\right) = \frac{\hat{q}}{\hat{m}}\left(\hat{\mathbf{E}} + \frac{\hat{\mathbf{v}}}{\hat{c}} \times \hat{\mathbf{B}}\right). \tag{5.82}$$

Here $\Delta[]_t$ and $\Delta[]_x$ indicate an appropriate finite difference operator. The final expressions on the RHS are the equations implemented by Runko.

Note also for the velocity update that $\Delta[\hat{\mathbf{u}}]_t$ is a numerical expression for acceleration and is thus scaled using $\Delta x/\Delta t^2$, which is changed to $\Delta t c/\hat{c}$ and moved to the right hand side. The update equation for position is defined by the integral over time of proper velocity $\mathbf{v}$, which takes the following form once scalings are introduced

$$\Delta[\hat{\mathbf{x}}]_t = \frac{\Delta t}{\Delta x}\frac{\hat{\mathbf{u}}}{\gamma(\hat{\mathbf{u}})}\frac{\Delta x}{\Delta t} = \frac{\hat{\mathbf{u}}}{\gamma(\hat{\mathbf{u}})}. \tag{5.83}$$

The current density $\mathbf{J}$ is now scaled as

$$\mathbf{J} = J_0\hat{\mathbf{J}} = \hat{q}\frac{q_0}{\Delta x^3}\hat{\mathbf{v}}\frac{\Delta x}{\Delta t} = \hat{q}\hat{\mathbf{v}}\frac{q_0}{\Delta x^2\Delta t}, \tag{5.84}$$

which implies $J_0 = q_0/\Delta x^2\Delta t$. Combining this scaling with 5.78 and 5.79 yields

$$\frac{q_0}{\Delta x^2} = \frac{m_0 c}{4\pi q_0\hat{c}\Delta t}, \tag{5.85}$$

which can be rearranged for Δx to form

$$\Delta x = 4\pi\frac{q_0^2}{m_0}\left(\frac{\hat{c}}{c}\right)^2, \tag{5.86}$$

using the substitution $\Delta x/\Delta t = c/\hat{c}$.

For a macro-particle containing N electrons/positrons, the reference charge and

mass will equal $q_0 = Nq_e$, $m_0 = Nm_e$ and the expression becomes

$$\Delta x = 4\pi N \frac{q_e^2}{m_e} \left(\frac{\hat{c}}{c}\right)^2 = 4\pi N \hat{c}^2 r_e, \tag{5.87}$$

where m_e, $q_e = e$ are the electron rest mass and elementary charge respectively, resulting in the appearance of the classical electron radius $r_e = e^2/m_e c^2 \approx 2.82 \cdot 10^{-13}$ cm. The field scaling in Gaussian units then becomes

$$B_0 = \frac{m_e c}{e\hat{c}} \left(\frac{c}{\hat{c}\Delta x}\right) = \frac{e}{r_e \hat{c}^2 \Delta x}. \tag{5.88}$$

Converting from code to CGS units now just requires selecting a length scale Δx:

$$\mathbf{B} = \hat{\mathbf{B}} B_0 = 1.705 \cdot 10^3 \frac{\hat{\mathbf{B}}}{\hat{c}^2} \frac{1\,\mathrm{cm}}{\Delta x} \mathrm{G}, \tag{5.89}$$

$$\mathbf{E} = \hat{\mathbf{E}} B_0 = 1.705 \cdot 10^3 \frac{\hat{\mathbf{E}}}{\hat{c}^2} \frac{1\,\mathrm{cm}}{\Delta x} \mathrm{statvolt\,cm^{-1}}, \tag{5.90}$$

$$\mathbf{J} = \frac{B_0}{4\pi\Delta t}\hat{\mathbf{J}} = \frac{ec}{4\pi r_e \hat{c}^3 \Delta x^2}\hat{\mathbf{J}} \approx 4.056 \cdot 10^{12} \frac{\hat{\mathbf{J}}}{\hat{c}^3} \left(\frac{1\,\mathrm{cm}}{\Delta x}\right)^2 \mathrm{statcoul\,s^{-1}}, \tag{5.91}$$

$$q = \hat{q} \frac{e\Delta x}{4\pi \hat{c}^2 r_e} \approx 1.356 \cdot 10^2 \frac{\hat{q}}{\hat{c}^2} \frac{\Delta x}{1\,\mathrm{cm}} \mathrm{statcoul}, \tag{5.92}$$

5.3.2 Relativistic Velocity-Verlet with Runko Scaling

To implement the relativistic velocity-Verlet scheme with Runko, conversion to the Runko unit system is performed by substituting Runko scalings into the update equations (5.29)-(5.44). Note that CGS units are now used, which simply introduces a factor $1/c$ to the cross-product of magnetic field and velocity. The chosen expression for the field scaling B_0 leads to cancellation of all other scalings, yielding the much simplified Runko-scaled update equations

$$\hat{\mathbf{u}}^{n+1/2} = \hat{\mathbf{u}}^n + \frac{k_{\Delta t}}{2}\frac{\hat{q}}{\hat{m}} \left(\hat{\mathbf{E}}^n + \frac{\hat{\mathbf{u}}^n}{\hat{c}\gamma(\hat{\mathbf{u}}^n)} \times \hat{\mathbf{B}}^n\right), \tag{5.93}$$

$$\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{x}}^n + k_{\Delta t}\frac{\hat{\mathbf{u}}^{n+1/2}}{\gamma(\hat{\mathbf{u}}^{n+1/2})}, \tag{5.94}$$

$$\hat{\mathbf{u}}^{n+1} = \hat{\mathbf{u}}^n + E + \frac{\hat{\mathbf{u}}^n + \hat{\mathbf{u}}^{n+1}}{2\hat{c}\gamma^{n+1/2}} \times B + \mathbf{c}_k, \tag{5.95}$$

using the shorthands

$$E := k_{\Delta t}\frac{\hat{q}}{\hat{m}}\frac{\hat{\mathbf{E}}^n + \hat{\mathbf{E}}^{n+1}}{2}, \tag{5.96}$$

$$B := k_{\Delta t}\frac{\hat{q}}{\hat{m}}\hat{\mathbf{B}}^{n+1}, \tag{5.97}$$

and with the constant term given by

$$\mathbf{c}_k = \frac{\hat{q}}{\hat{m}}\frac{k_{\Delta t}}{2\gamma(\hat{\mathbf{u}}^n)}\hat{\mathbf{u}}^n \times \hat{\mathbf{B}}^n - \frac{\hat{q}}{\hat{m}}\frac{k_{\Delta t}}{2\gamma(\hat{\mathbf{u}}^{n+1})}\hat{\mathbf{u}}^n \times \hat{\mathbf{B}}^{n+1}. \tag{5.98}$$

As before, Algorithm 1 can be applied to calculate $\hat{\mathbf{u}}^{n+1}$ explicitly from (5.95) using $\gamma^{1/2} \approx \gamma(\hat{\mathbf{u}}^-)$.

The factor $k_{\Delta t}$ is introduced to maintain direct control over the time-step size in Runko, which is normalised in the unit system and tied to spatial resolution via the CFL number $\hat{c} = \Delta t/\Delta x$ when the speed of light $c = 1$. To allow simulations of the same dynamics with varying time-step in Runko, each simulation must be normalised w.r.t. a common reference time-step Δt rather than the individual simulation time-step, which is expressed as $k_{\Delta t}\Delta t$ to distinguish the two. The normalisation procedure of Section 5.3, which removes time-steps entirely by cancelling out the Δt terms, now leave the $k_{\Delta t}$ factors behind which can be used to control the time-step without changing the mesh spacing via $\hat{c}$. For the reference simulation/time-step where $\Delta t = k_{\Delta t}\Delta t$ the setup is identical to standard Runko as $k_{\Delta t} = 1$.

5.3.3 Relativistic Boris-SDC with Runko Scaling

Applying the same procedure to the relativistic Boris-SDC update (in CGS) yields the scaled update equations

$$\hat{\mathbf{u}}_{m+1/2}^{k+1} = \hat{\mathbf{u}}_m^{k+1} + \frac{\Delta\tau_m}{2}\frac{\hat{q}}{\hat{m}}\left(\hat{\mathbf{E}}(\hat{\mathbf{x}}_m^{k+1}) + \frac{\mathbf{g}(\hat{\mathbf{u}}_m^{k+1})}{\hat{c}} \times \hat{\mathbf{B}}(\hat{\mathbf{x}}_m^{k+1})\right), \tag{5.99}$$

$$\hat{\mathbf{x}}_{m+1}^{k+1} = \hat{\mathbf{x}}_m^{k+1} + \Delta\tau_m\left[\mathbf{g}(\hat{\mathbf{u}}_{m+1/2}^{k+1}) - \mathbf{g}(\hat{\mathbf{u}}_{m+1/2}^{k})\right] + \textstyle\sum_{j=1}^{M} s_{m,j}\mathbf{g}(\hat{\mathbf{u}}_j^k)\ , \tag{5.100}$$

$$\begin{aligned}\hat{\mathbf{u}}_{m+1}^{k+1} = \hat{\mathbf{u}}_m^{k+1} + \Delta\tau_m\frac{\hat{q}}{\hat{m}}&\left[\frac{\hat{\mathbf{E}}(\hat{\mathbf{x}}_m^{k+1}) + \hat{\mathbf{E}}(\hat{\mathbf{x}}_{m+1}^{k+1})}{2} + \frac{\hat{\mathbf{u}}_m^{k+1} + \hat{\mathbf{u}}_{m+1}^{k+1}}{2\gamma(\hat{\mathbf{u}}_{m+1}^{k})\hat{c}} \times \hat{\mathbf{B}}(\hat{\mathbf{x}}_{m+1}^{k+1})\right] \\ &+ \mathbf{c}_k,\end{aligned} \tag{5.101}$$

with

$$\begin{aligned}\mathbf{c}_k = &\frac{\hat{q}}{\hat{m}}\frac{\Delta\tau_m}{2\hat{c}}\mathbf{g}(\hat{\mathbf{u}}_m^{k+1}) \times \hat{\mathbf{B}}(\hat{\mathbf{x}}_m^{k+1}) \\ &- \frac{\Delta\tau_m}{2}\frac{\hat{q}}{\hat{m}}\left[\hat{\mathbf{E}}(\hat{\mathbf{x}}_m^{k}) + \frac{\mathbf{g}(\hat{\mathbf{u}}_m^{k})}{\hat{c}} \times \hat{B}(\hat{\mathbf{x}}_m^{k}) + \hat{\mathbf{E}}(\hat{\mathbf{x}}_m^{k+1}) + \frac{\mathbf{g}(\hat{\mathbf{u}}_m^{k+1})}{\hat{c}} \times \hat{\mathbf{B}}(\hat{\mathbf{x}}_m^{k+1})\right] \\ &+ \sum_{j=1}^{M} s_{m,j}\frac{\hat{q}}{\hat{m}}\left[\hat{\mathbf{E}}(\hat{\mathbf{x}}_j^{k}) + \frac{\mathbf{g}(\hat{\mathbf{u}}_j^{k})}{\hat{c}} \times \hat{\mathbf{B}}(\hat{\mathbf{x}}_j^{k})\right].\end{aligned} \tag{5.102}$$

The factor $k_{\Delta t}$ to control time-step is included more easily for Boris-SDC, as the update equations already contain time-step modifying factors in the form of the node spacings $\Delta\tau_m$ and collocation matrix $s_{m,j}$. These are typically defined in general

form, normalised to span the unit time-step $[0, 1]$, and multiplied with Δt for use in any given SDC configuration. For a default Runko setup these can simply be left as they are or multiplied by $k_{\Delta t}$ when the same dynamics must be studied at a range of time-steps, as is the case in the next section.

5.3.4 Runko Implementation and Particle Helix Test

Implementation of both velocity-Verlet and Boris-SDC within Runko required some modification based on the observations made in Section 5.1.2. The standard setup of the code assumes a single round per time-step of field interpolation and solution of the field equations, i.e. a straight-forward implementation of the typical PIC operational diagram.

Fortunately, the Runko framework is modular and each subroutine can be used as required without interrupting the overall flow of operations. For velocity-Verlet the particle integration had to be split into position and velocity integration respectively. The field solution and interpolation must then occur between the two parts of the particle update. Additionally, for EMPIC the electric and magnetic field values are staggered in time by half a time-step, updated in a leapfrog manner just like the particles. An additional half time-step integration must thus be added to define both fields at the integer time-steps for use in the particle integration.

These changes are also required for Boris-SDC and two additional loops are required for the collocation nodes and sweeps. The final setup then has the particle integration, field solution and field interpolation occur with every step in time, nodes and iterations just as Figure 5.2. Saving the additional information from previous nodes and iterations required in SDC is done separately from the standard Runko C++ data containers. Finally, communication of particles between parallel subdomains is done once per time-step, avoiding the need to communicate the additional SDC data between different subdomains and data containers.

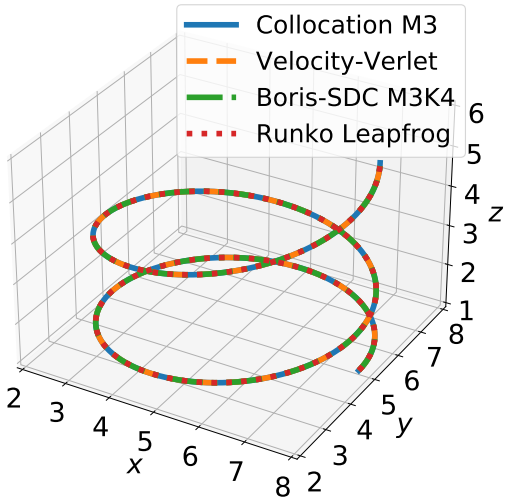

Figure 5.22: Inbuilt leapfrog, velocity-Verlet, 3-node collocation and Boris-SDC M3K4 trajectories in RUNKO helix setup for $t = [0, 45]$ using 1000 time-steps.

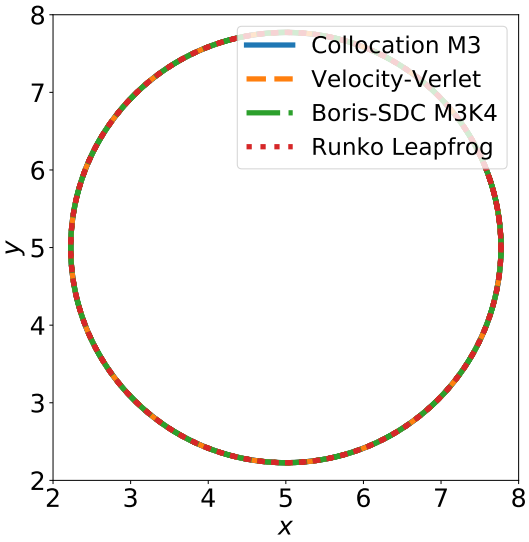

Figure 5.23: Inbuilt leapfrog, velocity-Verlet, 3-node collocation and Boris-SDC M3K4 xy-plane trajectories in RUNKO helix setup for $t = [0, 45]$ using 1000 time-steps.

To verify velocity-Verlet and Boris-SDC implementations in RUNKO separate from the spatial field solver, another simple electrostatic test case was devised with a constant electric and magnetic field. Fixed, spatially homogeneuos fields of $\mathbf{E} = (0, 0, -1.5 \cdot 10^{-3})^{\mathrm{T}}$ and $\mathbf{B} = (0, 0, 0.1)^{\mathrm{T}}$ were applied to every node on the grid. A single particle was used with the initial position $\hat{\mathbf{x}} = (7.75, 5, 1)^{\mathrm{T}}$ and coordinate velocity $\hat{\mathbf{v}} = (0, 0.95\hat{c}, 0)^{\mathrm{T}}$. The CFL number and effective speed of light was fixed at $\hat{c} = 0.45$. Current interpolation from particles to grid was disabled, effectively disabling the field solver as well. The constant fields then ensure no spatial error from the linear interpolation from grid to particles and the particle integrators can be tested in isolation from the remaining PIC functionality.

The setup creates a simple particle gyration about the magnetic field in the xy-plane with accelerating motion along the magnetic field lines (z-axis). The resulting helix trajectory and top-down circular trajectory can be seen in Figure 5.22 and Figure 5.23 respectively.

The trajectories produced by RUNKO leapfrog, velocity-Verlet and Boris-SDC

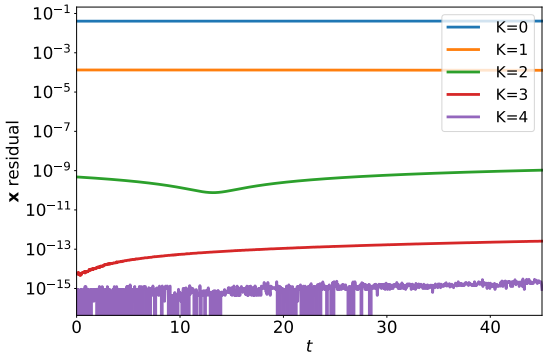

Figure 5.24: RUNKO and Boris-SDC M3K4 particle position residual for $t = [0, 45]$.

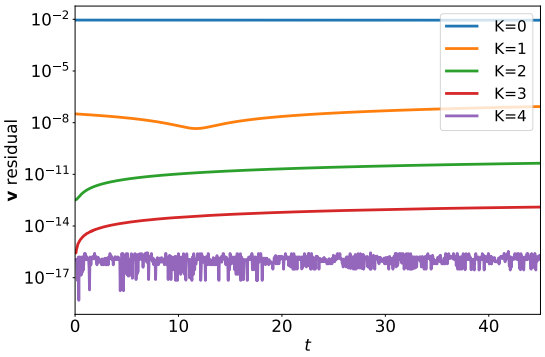

Figure 5.25: RUNKO and Boris-SDC M3K4 particle velocity residual for $t = [0, 45]$.

using $M = 2, 3, 5$ setups were all visually identical, with no discrepancies seen in the simulated dynamics.

The residuals shown in Figures 5.24 and 5.25 for an $M = 3$, $K = 4$ Boris-SDC solution showed iterations converging to machine precision. This demonstrates that Boris-SDC in RUNKO converges to the collocation solution and the associated higher order convergence should be maintained in the absence of spatial errors. All results from Boris-SDC in RUNKO were produced without a predictor, i.e. the "simple" initialisation routine mentioned in Section 4.3,2, due additional work required to implement the predictor in RUNKO.

Finally, a work-precision study was undertaken to further verify the new implementation. Relative error in position $\mathbf{x}$ was evaluated for a range of temporal resolutions using velocity-Verlet, 3-node and 5-node Boris-SDC. Error was calculated by comparison with a reference 5-node collocation solution using NT = 1600 time-steps, as no analytical solution was available.

The convergence order and performance outlook for the relative error in position as seen on Figure 5.26 came out as expected. The results exhibit the same higher order error reduction with time-step size and convergence with SDC iterations K seen in both relativistic and non-relativistic Penning trap tests.

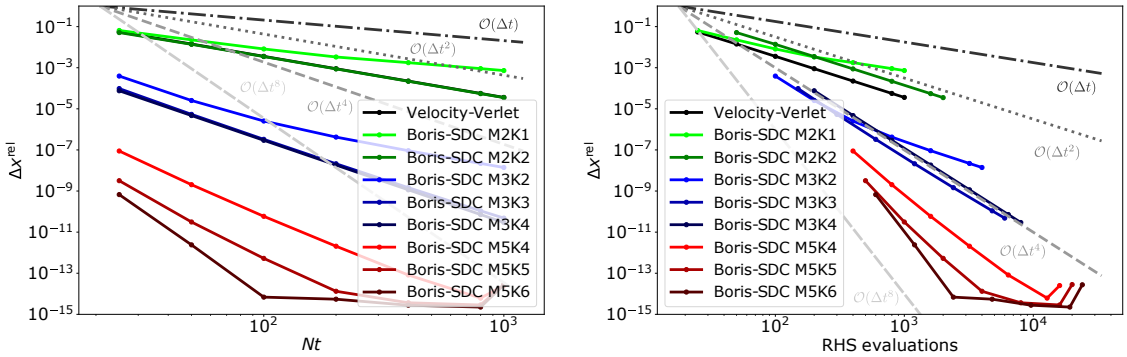

Figure 5.26: Work-precision of particle integrators for Runko helix trajectory constant field test.

5.3.5 Runko Penning Trap Test

As a final implementation test of Boris-SDC within Runko, work-precision results were sought for a Penning trap setup similar to Sections 5.1.3 and 5.2.4. Due to restrictions on domain boundary definition within Runko as well as the change of unit system, new initial values for the field and particles were set to create an appropriate trajectory, one influenced significantly by both the magnetic and electric field. The simulation parameters can be seen in Table 5.6, with the following fields applied to every node on the PIC grid

$$\mathbf{E}_{i,j,k} = |E| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix} x_{i,j,k}, \tag{5.103}$$

$$\mathbf{B}_{i,j,k} = |B|, \tag{5.104}$$

where $x_{i,j,k}$ is the position of the grid node with index i, j, k.

The main purpose of the study was to test the Boris-SDC implementation w.r.t. the field interpolation from grid to particle of Runko, making it the EMPIC counterpart to the second particle oscillator experiment (Figure 5.10) of Section 5.1.4.

Table 5.6: Validation study parameters

$\hat{c}$	0.45
t_{end}	45
$\hat{\mathbf{x}}(0)$	$(7.5, 5, 7.5)^{\mathrm{T}}$
$\hat{\mathbf{v}}(0)$	$(0.7\hat{c}, 0, 0.7\hat{c})^{\mathrm{T}}$
$\lvert E\rvert$	0.1
$\lvert B\rvert$	1

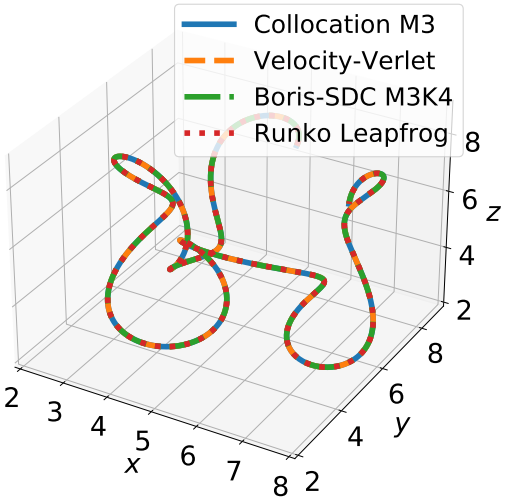

Figure 5.27: Inbuilt leapfrog, velocity-Verlet, 3-node collocation and Boris-SDC M3K4 trajectories in Runko Penning trap for $t = [0, 45]$ using 1000 time-steps.

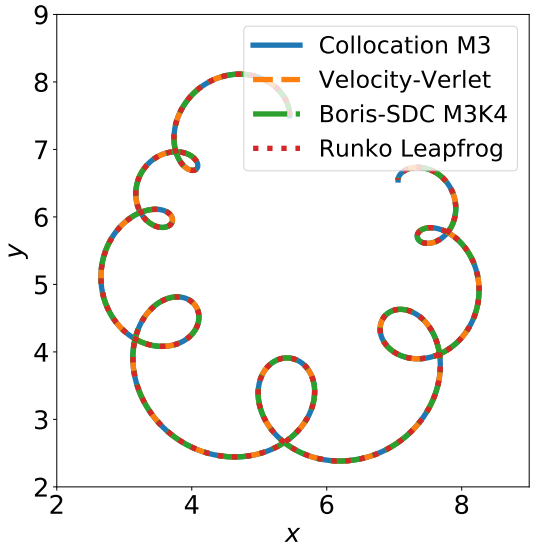

Figure 5.28: Inbuilt leapfrog, velocity-Verlet, 3-node collocation and Boris-SDC M3K4 trajectories in Runko Penning trap setup for $t = [0, 45]$ using 1000 time-steps.

The interpolation of exact field values defined on the PIC grid (as opposed to exact field evaluations at the particles) is not expected to introduce a spatial error threshold in this test, as the spatially linear fields should be captured exactly by the linear interpolation used in Runko.

The resulting trajectory and trajectory projection on the xy-plane can be seen in Figures 5.27 and 5.28 respectively.

The trajectories produced using Runko leapfrog, velocity-Verlet and Boris-SDC were again visually identical and while no analytical trajectory is available for comparison, the trajectories are reminiscent of the dynamics seen in the previous Penning trap setup. The implementation is further supported by the residuals seen in Figures

5.29 and 5.30, which again shows that the particle position and velocity produced by the Boris-SDC implementation converge to the collocation solution.

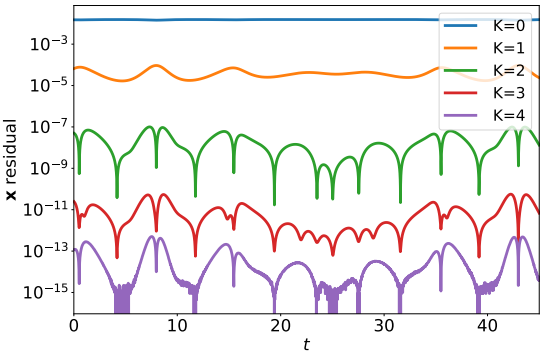

Figure 5.29: RUNKO and Boris-SDC M3K4 Penning trap particle position residual for $t = [0, 45]$.

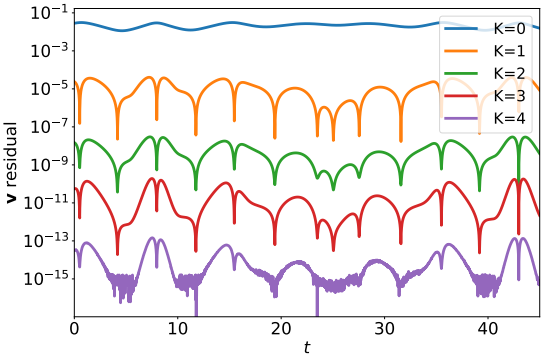

Figure 5.30: RUNKO and Boris-SDC M3K4 Penning trap particle velocity residual for $t = [0, 45]$.

The work-precision results for the RUNKO Penning trap were produced in the same fashion as the helix case in the previous section. Error was once again calculated by comparison with a reference 5-node collocation solution, using NT = 3200 time-steps.

Presented in Figure 5.31, the favourable convergence characteristics of Boris-SDC are still evident. The error threshold seen here is not due to the spatial interpolation which is exact for this problem, but rather the use of mixed single/double precision calculations in the particle calculations of the RUNKO interpolator.

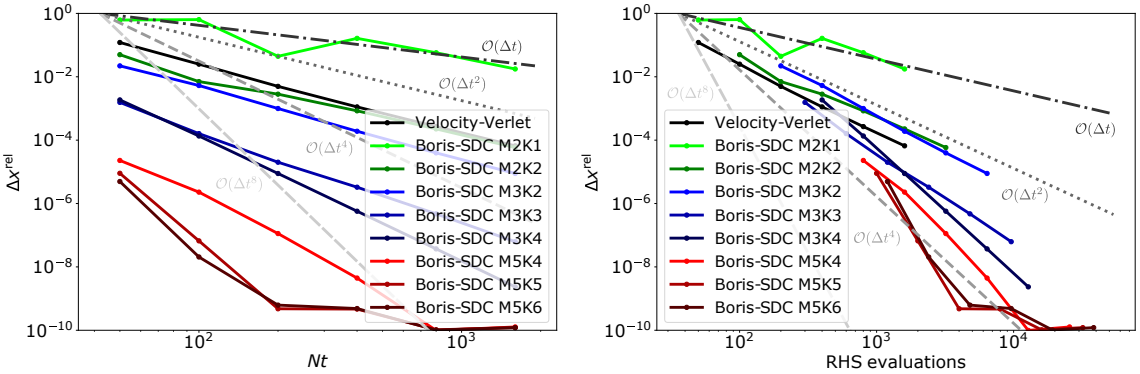

Figure 5.31: Work-precision of particle integrators for RUNKO Penning trap test.

These results provide an initial demonstration of how Boris-SDC can be success-

fully integrated to work in tandem with an existing PIC code such as Runko. At the time of writing, missing features which require implementation and testing include the current interpolation and field solver, which are required for proper electrostatic and electromagnetic problems with dynamic field coupling.

5.4 Conclusions

A custom electrostatic PIC code was written with the 2015 formulation of Boris-SDC implemented as the particle integrator for the first time. A different operational structure was used for the project code when compared to classic ESPIC, in which control of the interpolation and field solution routines rested with the particle integration module.

The combined ESPIC/Boris-SDC scheme was then tested using a simple particle oscillator setup, with a single negatively charged particle trapped in potential well defined by a fourth order polynomial. Work-precision results were produced for increasing spatial approximation of the electric field and demonstrated the impact of spatial error from ESPIC on the time convergence of the particle integrator. The high-order convergence of Boris-SDC was observed for all time-step sizes above the point where error becomes dominated by spatial saturation. The steeper time convergence above the error saturation points means the increased computational efficiency of Boris-SDC compared to Boris is still achievable where spatial error is sufficiently low. The setup was missing dynamic coupling of particles and field, but still provided valuable insight on the interaction of the temporal and spatial errors, setting the expectations for the proper ESPIC tests of the following chapter.

A novel relativistic generalisation of the Boris-SDC integrator was formulated by applying the 2019 Boris-SDC derivation to the relativistic Newton-Lorentz system and incorporating the relativistic factor as one of the converging quantities. The relativistic Boris-SDC integrator was found to retain the accuracy, high-order conver-

gence and performance benefits of the non-relativistic formulation in the relativistic regime, tested up to 99.9975% the speed of light.

Relativistic Boris-SDC was implemented and verified as part of the relativistic EMPIC code Runko, with high-order time convergence demonstrated both for the particle integrator in isolation and combined with the field interpolation routines of Runko. In both cases, Boris-SDC was found to be more accurate at a given time-step size and more computationally efficient than classic Boris. The implementation demonstrates that Boris-SDC can be applied as a replacement or supplemental integrator for classic Boris in existing PIC codes, provided these allow for modification of the order of operations. While the performance for real ESPIC/EMPIC applications are still to be explored, these initial results demonstrate that nothing about the relativistic regime or PIC codes inherently inhibit the desired convergence properties of Boris-SDC. The results of this chapter provide verification of the individual and combined schemes under study, with everything working as expected for simple settings.

Chapter 6

ESPIC with Boris-SDC Performance

With initial verification and implementation checking of KPPS completed in the previous chapter, performance evaluation for actual ESPIC case studies could commence. These evaluations begin with work-precision studies in which the accuracy of Boris and Boris-SDC was tested at a range of time-steps and ESPIC mesh resolutions, in identical fashion to the oscillator results from the previous chapter. Additionally, to compare performance of the two integrators against each other, a study on convergence of error against computational expense like that of Winkel et al. [92] was performed as well, now for the full ESPIC scheme.

Two benchmark problems were selected for testing, both 1D in nature. To cover a wider range of dynamics, a cold plasma problem was selected for the first case, while a warm plasma was studied for the second case. The two-stream instability and Landau damping are two classic problems for each of these general cases respectively [16]. Being well-studied and one dimensional made them the ideal next-step benchmarks for a new ESPIC code.

6.1 The Two-Stream Instability

The two-stream instability is an example of a *streaming instability*, wherein a beam containing one species of charged particles streams through another beam. Such counter-streaming beams are inherently unstable, as any perturbation in density or velocity distribution is reinforced by the charge bunching this induces in the other beam and vice-versa [16].

For example, consider fast electrons streaming past slower ions. If the two streams can be kept perfectly free of density or velocity perturbations, no instability occurs. In reality, any perturbation in either beam reinforces perturbations in the other. A perturbation in density represents periodic bunching of charge, which exudes an increased electric force compared to the surrounding fluid. Where two perturbed streams propagate relative to each other, such bunches are mutually reinforced by their increased forces of attraction or repulsion. This represents an instability growing exponentially in time, following the simple relation $\Delta n_1 \propto n_1$, where n_1 is the beam density [8].

The dispersion relation for a sinusoidally perturbed two-stream instability involving ions and electrons in one spatial dimension takes the form

$$1 = \omega_p^2 \left[\frac{m_e/m_i}{\omega^2} + \frac{1}{(\omega - k v_0)^2} \right], \tag{6.1}$$

where k is the mode of the perturbation, v_0 is the electron velocity, ω is the frequency of the oscillating instability, ω_p is the plasma frequency introduced in Chapter 1 and m_e/m_i is the ratio of electron to ion mass [16].

Multiplying through with a common denominator yields a fourth-order equation for ω, where real roots ω_r indicate a possible wave in the electric field of the form

$$\mathbf{E}_1 = E e^{i(kx - \omega_r t)} \hat{\mathbf{x}}. \tag{6.2}$$

Complex roots would occur in conjugate pairs $\omega_j = \alpha_j \pm i\gamma_j$ changing the time-dependent wave expression to

$$\mathbf{E}_1 = Ee^{i(kx-\alpha_j t)}e^{\pm\gamma_j t}\hat{\mathbf{x}}, \tag{6.3}$$

where a positive or negative value of the imaginary part γ_j indicate an exponentially growing or damped wave respectively. Unless all roots are real, at least one waveform will thus be unstable, as complex roots occur in pairs and one of such a pair must have a negative imaginary part.

If resolving the disparate time-scales of ion and electron movement isn't feasible, as is the case here, the instability can also be demonstrated for a time-frame where ion motion is negligible [8]. In this scenario, two counter-streaming electron beams are simulated on a background of immobile, neutralising ions. If the beams are of equal strength, i.e. the beams have the same plasma frequency $\omega_{p,1} = \omega_{p,2} = \omega_p$ and velocity magnitude $v_{0,1} = v_{0,2} = v_0$, the roots of the fourth-order equation for ω in this case become

$$\omega = \pm\left[k^2v_0^2 + \omega_p^2 \pm \omega_p(4k^2v_0^2 + \omega_p^2)^{1/2}\right]^{1/2}. \tag{6.4}$$

When solved for ω, the negative imaginary component γ_j of any pair of complex roots is then the analytical growth rate of the instability.

Figure 6.1 shows the evolution in time of the particles of a two-stream instability in phase-space, where two negatively charged beams are counter-streaming on a background of neutralising positive charge, following the setup and parameters presented in Section 6.1.1. In short, the sinusoidal perturbation in each beam interacts with the perturbation of the other beam, growing both amplitudes exponentially. This corresponds to an increasing 'bunching' of charge, which eventually becomes strong enough to decelerate and inverse the electron motion. Note the coherent

phase-space structure forming as the instability progresses, which shows particles oscillating back and forth across the phase-space 'eye'.

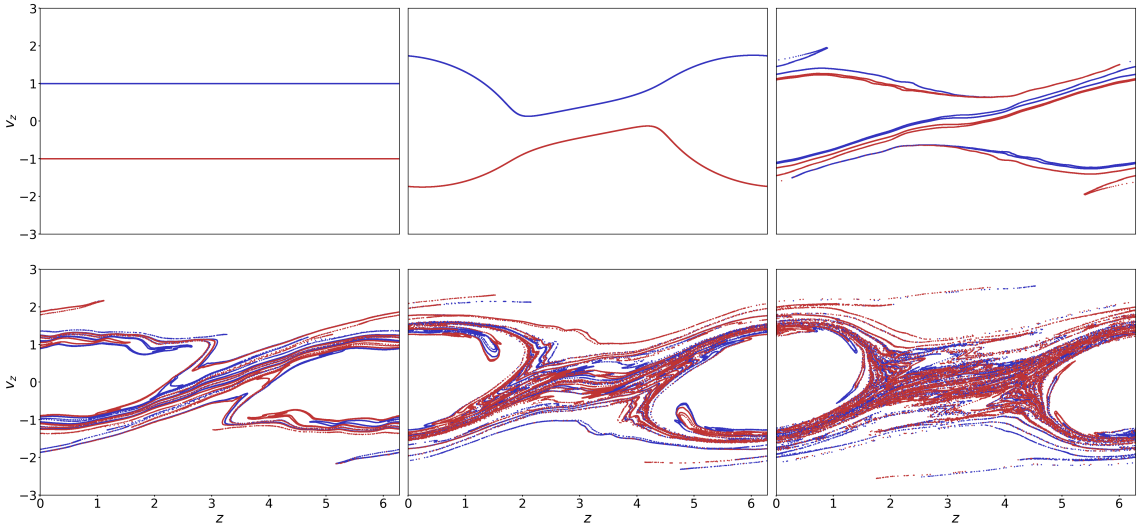

Figure 6.1: Two-Stream Instability particle position-velocity phase-space at $t = 0, 60, 120, 180, 240, 300$ for a half period sinusoidal perturbation in the charge density of each beam (density perturbation magnitude $A = 10^{-1}$). Data and visualisation produced via KPPS.

6.1.1 KPPS Computer Experiment Setup

As with the oscillator experiment in Chapter 5, KPPS was used to compare the performance of Boris and Boris-SDC in a set of benchmark problems. For the two-stream instability, two 1D test cases were conceived, one for weak and strong initial perturbation respectively. The weak case was designed to adhere to the linear theory outlined above, to serve both as a validation study and to test the performance of the integrators in simulating the linear dynamics prior to the instability onset.

The strongly perturbed case was constructed to make the study of performance in simulating the non-linear dynamics after the onset of the instability more convenient, by using a shorter growth phase.

The general setup of both cases was as follows: Two cold beams of negatively charged particles were initialised on a uniform background of positive, neutralising charge (defined on the mesh). Each beam was treated as a separate species of

particles to facilitate different beam configurations and visualisation. Each beam was defined by a distribution of NQ particles with charge q, and charge to mass ratio $a = q/m$. To target a specific solution of the linear theory given in the previous section, which depends on the plasma frequency ω_p, the charge q was calculated for a given input value of ω_p using

$$q = \frac{\omega_p^2}{na\epsilon}, \tag{6.5}$$

where $n = \mathrm{NQ}/L$ is the average number density of the beam, using the domain length L and number of computational particles NQ, and ϵ is the permittivity.

The two beams were given uniform initial velocities v_0 equal in magnitude and opposite in direction. The instability was driven by an initial sinusoidal perturbation in the charge density of each beam $\rho_\delta = A\sin(kz)$.

The perturbation was created by defining the uniformly spaced particle positions z_0 and perturbing these sinusoidally appropriately according to

$$z = z_0 \pm A\cos(2\pi k z_0/L), \tag{6.6}$$

where the plus-minus sign indicates that the beams are perturbed in opposite directions. The resulting perturbations cancel out, leaving the system at net zero charge globally at the beginning of the simulation.

Using these equations, a two-stream instability could be setup in KPPS for any desired combination of the parameters of interest: ω_p, k, v_0 and A. While the perturbation magnitude A does not appear in the dispersion relation for the instability, it is nevertheless crucial in the setup as it must be small for the linear theory to hold ($A \ll 1$ [8]), and the value of it is the only difference between the weak and strong cases studied here. Table 6.1 presents the values used to setup the weak and

strong cases according to the procedure above.

Table 6.1: Physical setup parameters for the two-stream instability cases.

Parameter	**Weak**	**Strong**
L	2π	2π
ω_p	1	1
k	1	1
A	10^{-4}	10^{-1}
$\lvert v_0 \rvert$	1	1
ϵ	1	1

6.1.2 Results

To ensure that the two cases were correctly implemented and to provide additional verification of KPPS overall, the evolution of the L2 norm of the electric field was studied for comparison with the linear theory. The L2 norm of the E-field as defined on the finite difference mesh was calculated as

$$||E||_{L2} = \sqrt{\Delta z \sum_{i=1}^{N_i} E_i^2}, \tag{6.7}$$

where Δz was the mesh spacing, i denotes a given finite difference node on the 1D mesh and N_i is the total number of nodes.

This initial study was performed using the simulation specific parameters found in Table 6.2, and represent relatively low fidelity simulations. If the dynamics of the system are accurately captured using these parameters, the code can be used with confidence for the more extensive work-precision and performance studies below.

Figure 6.2 shows the results of both cases and includes comparison between the best fit of the weak simulation data with the predicted growth rate from theory. The best fit was made over the time interval $t = [12.5, 17.5]$, which was determined by visual inspection of the electric field growth. The chosen interval covered most of

Table 6.2: Simulation setup parameters for the two-stream instability dynamical study.

Parameter	Key	Weak	Strong
Particle integrator	-	Boris-SDC M3K2	Boris-SDC M3K2
Time steps	NT	500	200
Mesh resolution	NZ	100	100
Particle count	NQ	$2 \cdot 10^4$	$2 \cdot 10^4$
End time	T_E	50	20

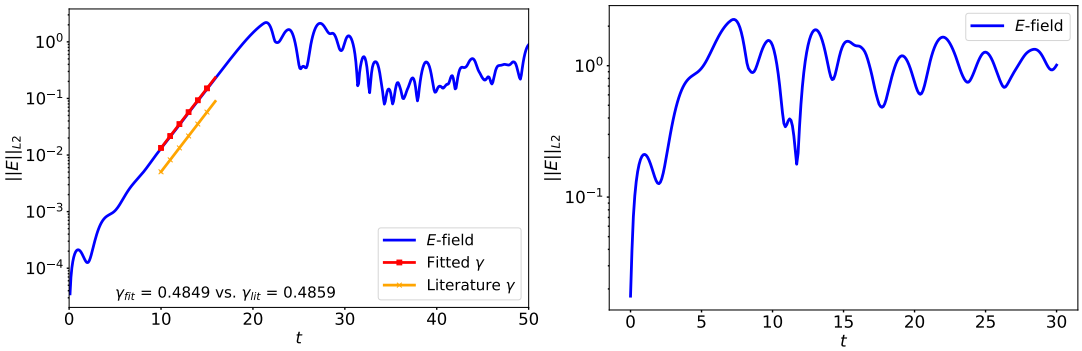

Figure 6.2: Evolution of L2 norm of electric field for weak (left) and strong (right) two-stream instability including fitted and literature growth rate.

the linear growth phase while leaving out ambiguous regions close to the oscillatory initial/instability phases. The simulated growth rate $\gamma = 0.4839$ matches the theoretical value $\gamma = 0.4859$ nicely, giving a relative error of 0.4 %. All relative errors in this study were calculated using

$$\mathrm{Err}^{\mathrm{rel}} = \frac{|x_{\mathrm{ref}} - x|}{|x_{\mathrm{ref}}|}, \tag{6.8}$$

where x denotes the simulated quantity of interest (electric field norm here) compared against x_{ref}, denoting the appropriate analytical or reference solution.

No linear theory was available for the strong two-stream instability, but the magnitude of the electric field grows in the same manner as the weak case and settles around the same value. Combined with the fact that the results were produced using the same code and general setup as the weak case, the above results were considered sufficient verification to proceed.

To evaluate the performance of Boris-SDC in simulating the two-stream instability, work-precision studies akin to those in Chapter 5 were conducted. The two instability cases were simulated for a range of time-step sizes at three levels of mesh resolution, covering three orders of magnitude, using both Boris and Boris-SDC. Evaluation was performed using the relative error (6.8) in E-field norm (6.7) when compared to a high accuracy reference solution (Table 6.3).

Table 6.3: Simulation setup parameters for the two-stream instability reference solutions.

Parameter	**Key**	**Weak**	**Strong**
Integrator	-	Boris-SDC	Boris-SDC
Collocation nodes	M	3	3
SDC iterations	K	3	3
Time steps	NT	5000	5000
Mesh resolution	NZ	5000	5000
Particle count	NQ	$2 \cdot 10^5$	$2 \cdot 10^5$
End time	T_E	10	10

Each series of simulations (see Table 6.3) was run for 10 seconds of simulation time, and sampled at each second mark for a total of 10 sets of work-precision data for each integrator/mesh resolution combination. The 10 second simulation time was chosen to gather performance data in the linear regime of the weak case and linear plus non-linear regime of the strong case.

Table 6.4: Simulation setup parameters for the two-stream instability work-precision study.

Parameter	**Key**	**Weak**	**Strong**
Integrator	-	[Boris-SDC (M=3, K=2), Boris]	-
Time steps	NT	$[10, 20, 40, 50, 80, 100, 200, 400, 500, 1000]$	-
Mesh resolution	NZ	$[10, 100, 1000]$	-
Particle count	NQ	$2 \cdot 10^5$	-
End time	T_E	10	-

Figure 6.3 presents the progression of work-precision for both Boris and Boris-SDC at 4 sampled times of the weak case, evenly spaced from 1 to 10 seconds. The linearity of the dynamics inherent to this simulation setup allows the desirable convergence behaviour of Boris-SDC to show even 10s into the simulation. Much

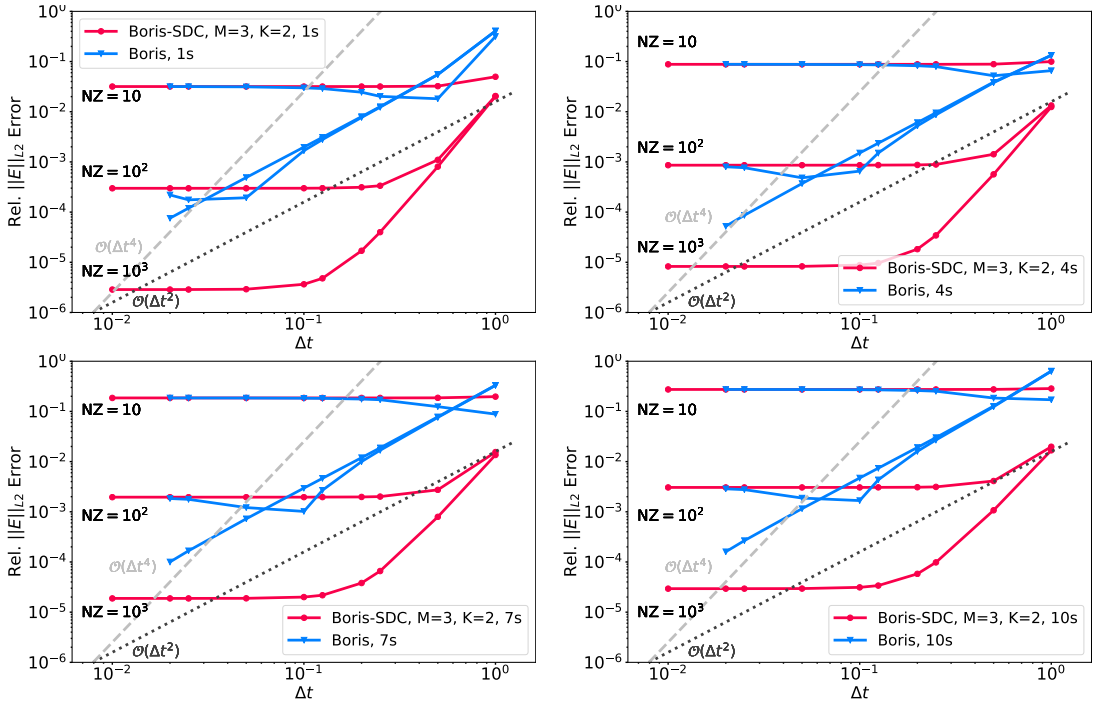

Figure 6.3: Convergence of the E-field L2 norm error of Boris and Boris-SDC for weak two-stream instability at 1, 4, 7 and 10s simulation time.

like the oscillator study from the previous chapter, the accuracy associated with the spatial resolution enforces a hard error threshold, below which no increase in temporal resolution is beneficial. Boris-SDC is seen to be more accurate than the equivalent Boris simulation at any time-step size above this error saturation level.

To properly gauge the performance of the two integrator types, the above data with respect to time-step size can be shifted to account for the increased computational expense of Boris-SDC during each time-step. The KPPS implementation of Boris-SDC performs one evaluation of the RHS of the Newton-Lorentz force on the particles per species update, which requires a full ESPIC cycle. In other words, the particle charge is interpolated, field solution found and electric field interpolated back to the particles once per node update $x_m \to x_{m+1}$. The number of nodal updates per time-step depends on the number of nodes and SDC sweeps as

$$\frac{\text{RHS evals}}{\Delta t} = (M-1)K. \tag{6.9}$$

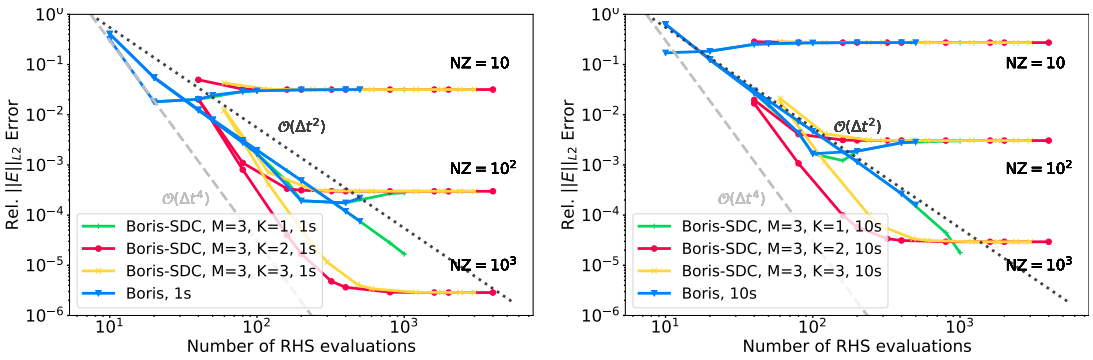

Figure 6.4: Performance comparison of Boris and Boris-SDC for weak two-stream instability at 1 and 10s simulation time.

The majority of the computational effort in PIC essentially lies in this evaluation of the Newton-Lorentz force; for standard Boris, this evaluation occurs only once per time-step. For a setup like the one studied here, with 3 nodes and 3 SDC sweeps, the computational expense of Boris-SDC is thus on the order of 6 times that of classic Boris. Transforming the x-axis of the work-precision plots using (6.9) yields the error achieved versus RHS evals, a good proxy for computational expense. Figure 6.4 presents the performance comparison of Boris and Boris-SDC at the beginning and end of the weak case simulations. When accounting for computational expense, it is clear that using a 4th order Boris-SDC setup is unnecessary if the spatial error is high, as was the case in the previous chapter. More interesting is the cross-over point for the 1000 cell count simulations around a relaive error of 10^{-3}, below which Boris-SDC produces a lower error at a cheaper cost in terms of RHS evaluations. For completeness sake, the work-precision trends of a series of Boris-SDC simulations with $M = 3$ and $K = 1, 3$ are also plotted. These demonstrate that Boris-SDC becomes equivalent to standard Boris in terms of error and cost if set to be 2nd order accurate, but once the order of the collocation solution is reached, further iterations do not noticably improve accuracy.

In summary, for a linearly evolving ESPIC problem, Boris-SDC shows a better performance than Boris below a relative error of 10^{-3}, which the ESPIC spatial accuracy must be able to match. Thus, for a desired maximum error lower than this value, an ESPIC code using Boris-SDC as the particle integrator can be setup to

achieve this error level at reduced computational cost in terms of force evaluations.

The results for the weak two-stream instability shown above were for a predominantly linear problem and it was thought likely that the performance outlook might change once the instability properly set in and the system becomes increasingly chaotic.

The strong case work-precision results in Figures 6.5 and 6.6 confirm this suspicion, with the minimum achievable error rising as the instability progresses. The increasing non-linearity can be seen to affect not just the convergence order, but the smoothness of convergence as well. While the solutions eventually do converge at the spatial error threshold, the reliability of the solutions at coarser time-step sizes are clearly affected greatly by the non-linear dynamics at this point in the simulation. The jagged nature of the lines and sudden jumps in precision were observed for several of the samples in the latter part of the strong case simulations. This adds some difficulty in making more general comments on the performance outlook under these conditions. At these times, the instability has stopped growing and entered a phase of stable energy exchange between particles and field, which proves itself to be highly oscillatory.

The sudden gains in precision observed for both integrators at the samples close to 10 seconds might then be due to the rapid changes in the electric field across a single time-step. If the sampling time is near a sharp local maximum, minimum or saddlepoint, the computed solution for a given time-step can be noticeably different for Boris vs. Boris-SDC since the latter samples the electric field multiple times per time-step while the former does not. The additional field sampling allows for a correction of error that could have resulted in resolving some features of the time-evolution better compared to the more accurate reference solution.

The residuals produced for Boris-SDC (Figure 6.5) indicate that this is indeed the case. In SDC, the residuals at a given node m compares calculated node values to

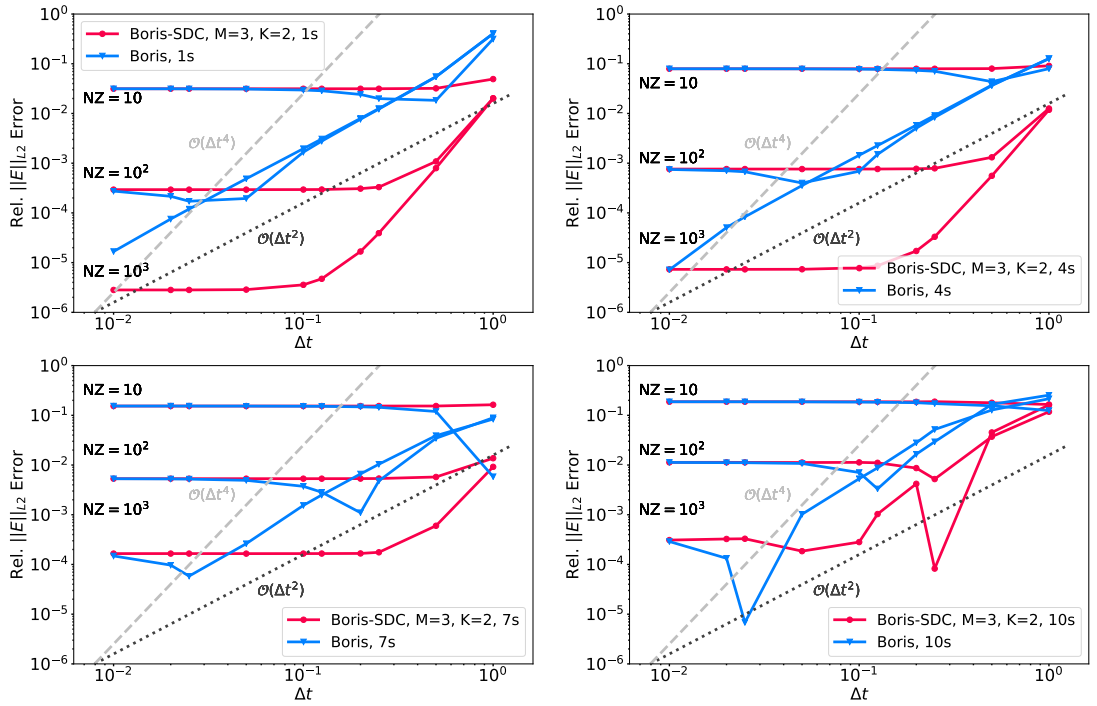

Figure 6.5: Convergence of the E-field L2 norm error of Boris and Boris-SDC for strong two-stream instability at 1, 4, 7 and 10s simulation time.

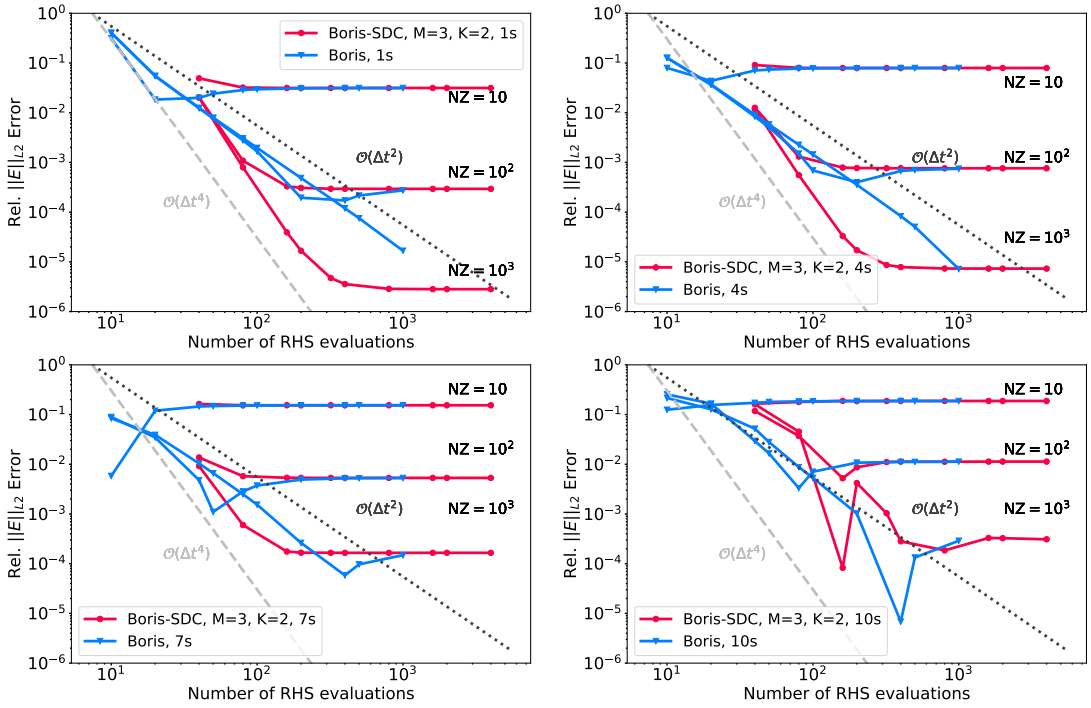

Figure 6.6: Performance comparison of Boris and Boris-SDC for strong two-stream instability at 1,4, 7 and 10s simulation time.

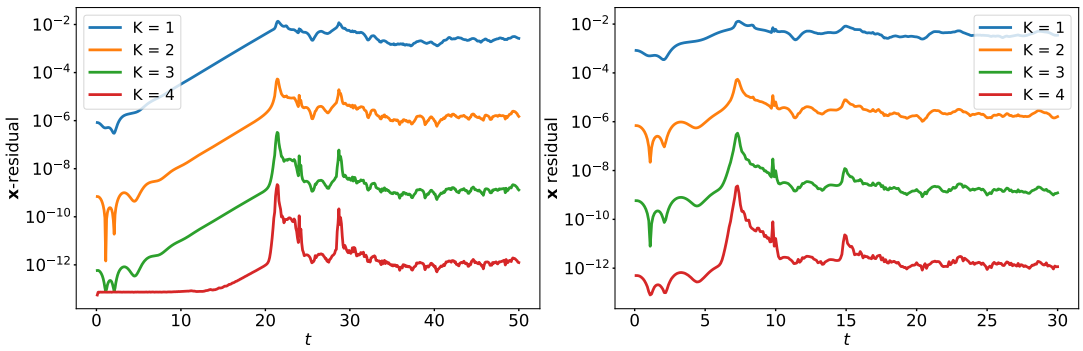

Figure 6.7: Average L2 norm of x-residual over all nodes and particles for weak (left) and strong (right) two-stream instability.

those given by the final SDC solution including all nodes

$$\mathbf{R}_{x,m} = \mathbf{x}_0 + \sum_{j=1}^{M} q_{m,j}\mathbf{v}_j - \mathbf{x}_m \tag{6.10}$$

$$\mathbf{R}_{v,m} = \mathbf{v}_0 + \sum_{j=1}^{M} q_{m,j}\mathbf{F}(\mathbf{x}_j, \mathbf{v}_j) - \mathbf{v}_m \tag{6.11}$$

recalling that q defines the collocation weights. A spike in residual indicates a mismatch between the final high order collocation polynomial and the node-to-node values, i.e. the solution is changing rapidly across the time-step.

It is evident from the residuals that the time interval wherein the linear growth regime ends might prove problematic when studying the work-precision, as particle integrator accuracy varies wildly by several orders of magnitude. The interval 10 - 50 seconds is outside the range of times sampled for the weak two-stream instability work-precision plots and should not affect the study. For the strong two-stream instability however, the peaks appear to lie very close to the times sampled for the work-precision study. These lapses are therefore disregarded w.r.t. performance comparisons here and in the following sections, as both integrators can be seen to converge to the same error as the temporal resolution increases.

All things considered, the performance of Boris-SDC is not observed to be clearly

superior to Boris in the non-linear regime in terms of computational efficiency. Boris-SDC does achieve lower error for any given time-step before the spatial error saturation, but in the non-linear regime the difference is not enough to offset the additional computational cost of the SDC. That being said, this appears to be due to the limited accuracy of the spatial solution, as the convergence of Boris-SDC is clearly steeper than that of Boris at times 1, 4 and 7 seconds. Without the error saturation, it is likely that regions of superior performance would be observed at lower errors, in the range $10^{-4} \rightarrow 10^{-6}$, matching the results from the weak two-stream instability. Even the results at 10 seconds indicate a faster decrease in error for Boris-SDC before the jump in precision towards the saturation point.

The results of the two-stream instability work-precision study indicated a definite benefit in performance of Boris-SDC over Boris in achieving higher accuracy results (relative error 10^{-3} and below) during the linear growth regime. When non-linearity sets in and phase-space vortices form the picture becomes less clear, with Boris-SDC and Boris appearing to be more or less equivalent in performance when accounting for computational expense. It should be noted however, that in the non-linear results, the lowest error allowed by the highest accuracy spatial solution was higher than the range in which favourable performance for Boris-SDC was observed. If spatial accuracy was further raised via increased resolution or higher order methods, there is no reason why Boris-SDC should not exhibit the same comparative performance as in the linear results.

6.2 Landau Damping

One of the more surprising results from plasma physics is that of collisionless damping, termed Landau damping after Landau [44], who predicted it. For a thermal plasma evolving with no physical collisions according to Vlasov's equation, one might intuit that electrostatic waves would propagate freely, with no damping, due to the

lack of an obvious energy transfer mechanism. In reality, damping occurs by the transfer of energy between particles and electric field. Consider an electrostatic wave with phase velocity $v_\phi = \omega/k$. For a thermal plasma adhering to the Maxwell-Boltzmann distribution, there is a set of particles with velocities around the phase velocity $v_\phi \pm v_{tr}$. The particles within this range are trapped by the wave and oscillate around the phase velocity. For a standard Maxwellian, the slower part of trapping range ($v_\phi - v_{tr}$ to v_ϕ) contains more particles than the fast part range ($v > v_\phi$) and the energy gain required to accelerate them into the fast range is supplied by the wave. Thus energy dissipates from the wave and it is damped with no collisions required between the particles. Figure 6.8 shows visually the initial setup (and subsequent evolution) of phase space density affected by Landau damping, constructed using a distribution of negatively charged particles on a neutralising background. While difficult to observe in the phase-space evolution, the plasma in the figure slowly returns towards a thermal equilibrium state as the simulation progresses due to the mechanism of Landau damping.

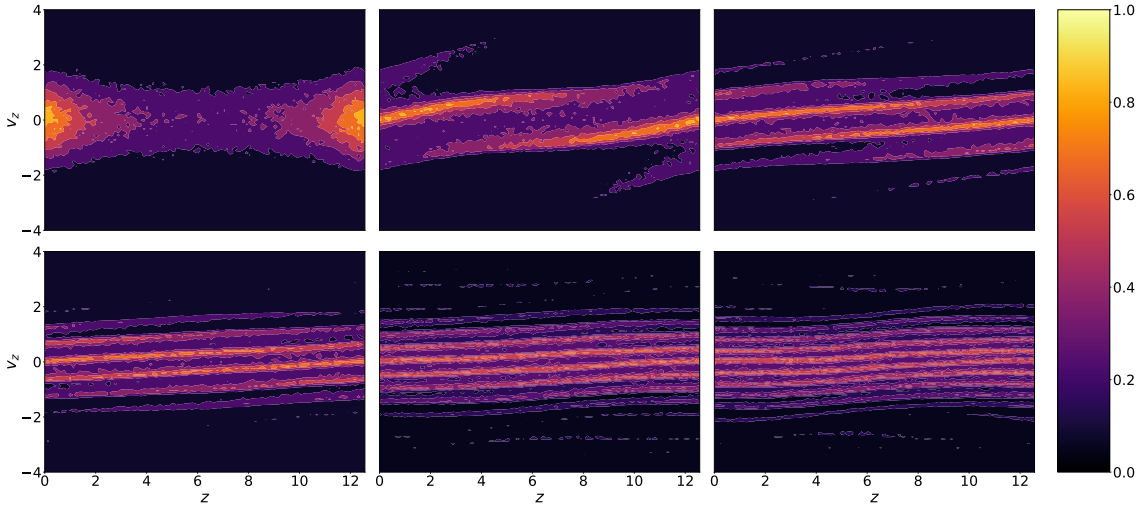

Figure 6.8: Landau damping density distribution function at $t = 0, 60, 120, 180, 240, 300$ for a single period sinusoidal perturbation in the charge density of each beam (magnitude $A = 0.5$). Data and visualisation produced via KPPS.

Landau damping was predicted theoretically before it was observed, in the seminal paper by Landau [44] "on the vibrations of the electronic plasma", and is essentially

the inverse of the two-stream instability. In the two-stream instability, the energy transfer from the particles to the electric field cause a growing electrostatic wave whereas in Landau damping the lost energy does not fuel an instability but rather damps the initial wave.

6.2.1 KPPS Computer Experiment Setup

Similar to the two-stream instability, Landau damping displays a rate of damping action (as opposed to growing action) which can be measured. Landau damping has been studied extensively in literature for both weak and strong perturbations, and a handful of previous studies (see Table 6.5) conveniently use an identical pair of benchmark setups, one for the weak and strong perturbation cases respectively. To mirror the two-stream instability tests and for ease of validation, it was decided to emulate the literature cases. These use the same initial phase space density and velocity distributions, parameterised in terms of the perturbation mode and magnitude. To study the problem in KPPS the setup must be expressed in terms of a particle distribution, which is derived here. The density distribution studied takes the form of a steady state value perturbed sinusoidally

$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}} \left(1 + \alpha \cos(Kx)\right) e^{-\frac{v^2}{2}}, \tag{6.12}$$

where α and K are the perturbation magnitude and mode respectively. This density distribution function is for the particle species of interest, with an assumed neutralising background. Ayuso de Dios and Hajian [5] notes that the setup is defined so that the charge density equals the density function in configuration space, i.e. $\rho = f(x)$:

$$n(x, 0) = \rho(x, 0) = \int f(x, v) dv = 1 + \alpha \cos(Kx), \tag{6.13}$$

which implies a particle charge of 1 according to (2.20). In a similar fashion, integrating over position yields the velocity distribution of the species

$$f(v,0) = \int f(x,v)dx = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}}. \tag{6.14}$$

For a thermal plasma, the velocity distribution takes the form of a standard Maxwell-Boltzmann distribution

$$f(v) = \left(\frac{m}{2\pi k_B T}\right)^{\frac{1}{2}} \exp\left(-\frac{mv^2}{2k_B T}\right), \tag{6.15}$$

here in 1D, where k_B is the Boltzmann constant, T is the absolute temperature and m is the particle mass. This distribution can be rewritten in terms of the thermal speed, defined as the root mean square of the 1D velocity

$$v_{th} = \sqrt{\frac{k_B T}{m}}. \tag{6.16}$$

The velocity distribution written in terms of the thermal speed then becomes

$$f_v(v) = \int f(x,v)dx = \frac{1}{\sqrt{2\pi}} \frac{1}{v_{th}} e^{-\frac{v^2}{2v_{th}^2}}, \tag{6.17}$$

which implies that the density distribution under study (6.12) represents a plasma uniform in configuration space until perturbed sinusoidally, with a standard Maxwell-Boltzmann velocity distribution at a thermal speed of 1.

Note that all integrations of $n(x) = \rho(x)$ over L conveniently equal L for any combination of domain length and perturbation mode wherein KL equals an integer multiple of 2π. Where this is the case the effect of perturbation on the density cancels out over the domain and the effective density is simply 1.

Table 6.5: Collected literature results for Landau damping Vlasov simulation codes. The theoretical growth rate γ is reported by multiple of the studies below, calculated from linear analysis of the weak case.

Author	**Weak** γ	**Rel. Error** (%)	**Strong** γ
Theoretical	-0.1533		
Ayuso de Dios and Hajian [5]	-0.1532	0.065	-0.292
Canosa et al. [15]	-0.145	5.4	
Cheng and Knorr [17]	-0.1532	0.065	-0.281
Nakamura and Yabe [54]	~ -0.1533	~ 0	-0.280
Rossmanith and Seal [66]	~ -0.1533	~ 0	-0.292

The parameters used for the KPPS Landau damping simulations can be seen in Table 6.6. The independent, problem-defining parameters are listed at the top with the important but dependent physical parameters listed at the bottom.

In the existing literature mentioned above, the problem was used as a benchmark for codes in which the Vlasov and Poisson equations are solved directly. Most of the studies opted to study (6.12) with two different perturbation magnitudes $\alpha = 0.01$ and 0.5, producing predicted damping rates for weak and strong damping cases respectively. Their combined results have been compiled in Table 6.5.

To construct the same setup in an ESPIC code, the charge density distribution (6.13) must be translated into an initial particle distribution, which can then be assigned velocities such that the distribution samples the Maxwellian (6.14) as close as possible at every point along the position axis.

The initial particle distribution was established in KPPS using a uniform distribution z_0 perturbed using

$$z = z_0 - \frac{\alpha}{n_0 K} \cos(K z_0), \tag{6.18}$$

where z is the perturbed position of a given particle, the perturbation magnitude α and mode K. The average unit density n_0 can be calculated from the continuous problem (6.13) using

$$n_0 = \frac{1}{L}\int^{L} n(x,0)dx = \frac{1}{L}\int^{L} 1 + \alpha\cos(Kx)dx. \tag{6.19}$$

Since the charge density of the particle species is given globally, the charge of the macro-particles must be assigned based on this and unlike the two-stream instability, the problem does not use the plasma frequency as an independent input parameter. The particle charge was calculated for a given simulation by dividing the global charge of the species by the desired quantity of computational particles NQ

$$q = \frac{Q}{\mathrm{NQ}} = \frac{\int^{L}\rho(x)dx}{\mathrm{NQ}} = \frac{\int^{L} 1 + \alpha\cos(Kx)dx}{\mathrm{NQ}}. \tag{6.20}$$

The compiled PIC settings used to reproduce the weak and strong Landau damping cases in KPPS are summarised in Table 6.6.

Table 6.6: Physical setup parameters for the Landau damping cases.

Parameter	**Weak**	**Strong**
L	4π	4π
K	0.5	0.5
α	0.05	0.5
v_{th}	1	1
ϵ	1	1
ω_p	1	1
n_0	1	1
q	L/NQ	L/NQ

The setup parameters used are identical to those in the studies listed in Table 6.6, except for the perturbation magnitude used in the weak case which was set to $\alpha = 0.01$ in the literature studies. The nature of the particle setup meant that very small perturbations in position become drowned out by noise due to the relatively low number of particles. This noise is inherent to PIC codes, most significantly for the low order particle shapes used in KPPS, see the discussion in Section 3.5. The noise would not be present in any of the collected literature studies as these are all based on direct discretisation of the Vlasov equation, see the relevant discussion in

Section 2.3.

Little to no observable damping was observed for a perturbation magnitude of $\alpha = 0.01$ at the computationally desirable particle quantities, so the magnitude was increased slowly until damping was observed at $\alpha = 0.1$.

6.2.2 Results

Again, the L2 norms of the electric field (6.7) in the simulations were compared to reference values, here the literature results.

The necessary simulations were conducted using the parameters found in Table 6.7, which are of similar fidelity to the two-stream study except for the use of ten times more particles. This number was found to be the lowest number of computational particles necessary to effectively resolve the dynamics of weak Landau damping.

Table 6.7: Simulation setup parameters for the two-stream instability dynamical study.

Parameter	**Key**	**Weak**	**Strong**
Time steps	NT	300	300
Mesh resolution	NZ	100	100
Particle count	NQ	$2 \cdot 10^5$	$2 \cdot 10^5$
End time	T_E	30	30

Figure 6.9 shows the results of both cases and includes a comparison between the damping and growth rates observed for the simulation data and the literature values (Table 6.5). The simulation damping/growth rates were calculated as the linear best fit across the peaks of the oscillating electric field norm. The weak Landau simulation resulted in a damping rate of $\gamma = -0.147$, which gives a relative error of 4.11 % compared to the theoretical value and the most accurate literature predictions. While an order of magnitude less accurate than the two-stream instability prediction, the discrepancy is explained by the noise issues discussed at the

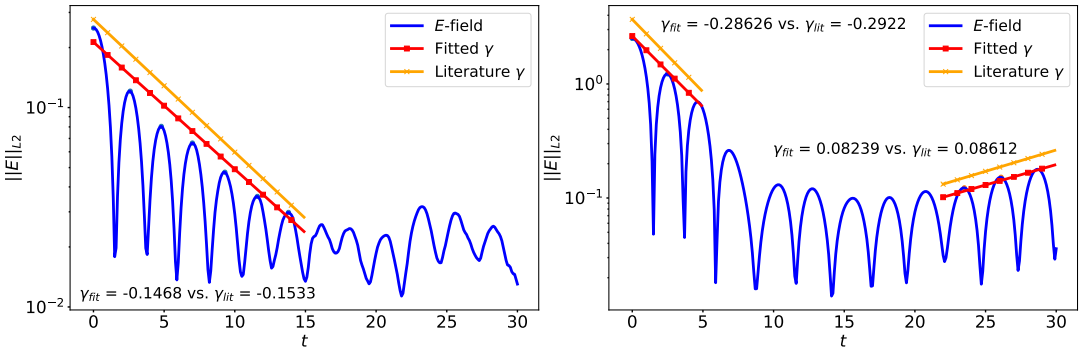

Figure 6.9: Evolution of L2 norm of electric field for weak (left) and strong (right) Landau damping including fitted and literature damping/growth rates.

end of the previous section. The increased perturbation magnitude that had to be used to see damping in the PIC simulation changes the dynamics away from the weakly perturbed setup known from literature and brings the simulation closer to the strongly perturbed case. It is thus unclear whether the weak damping simulation is still strictly in the regime described by the theoretical damping rate, which is determined from linear theory on the assumption that $\alpha \ll 1$ [44]. In light of these difficulties, the weak damping exhibited is remarkably close to the theoretical and literature values. The damping rate obtained for the strong case, $\gamma = -0.28825$ compared to the average of the literature values $\gamma_{\text{lit}} = -0.2863$ indicates a relative error of 0.68 %. This agreement with literature, considering these were not PIC results, demonstrate that the damping dynamics were captured correctly by the code.

The SDC residuals for the Landau damping dynamics (Figure 6.10) did not highlight any regions of special interest in the solution time, with expected fluctuations accompanying the rapid oscillations of the electric field.

Comparison was again made against a high accuracy reference solutions, with identical simulation parameters to those used for the two-stream instability, applied to the Landau damping setup in Table 6.8.

Each series of simulations (see Table 6.9) was again run for 10 seconds of simulation time and sampled at each second. The 10 second simulation time once more

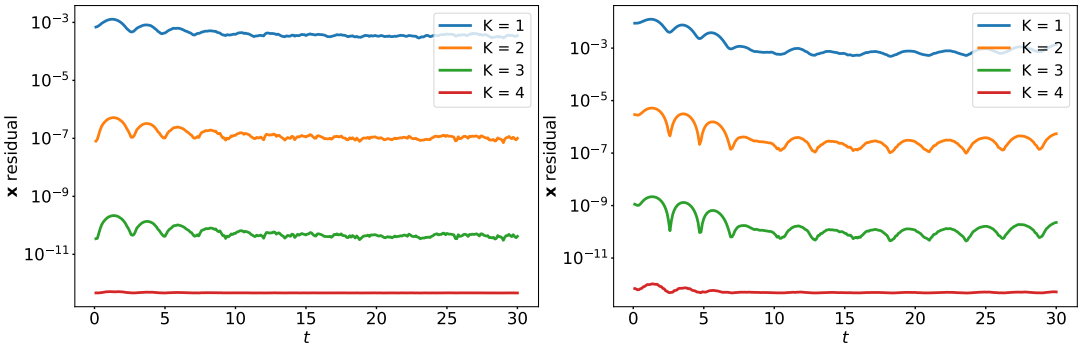

Figure 6.10: Average L2 norm of x-residual over all nodes and particles for weak (left) and strong (right) Landau damping.

Table 6.8: Simulation setup parameters for the Landau damping reference solutions.

Parameter	**Key**	**Weak**	**Strong**
Integrator	-	Boris-SDC	Boris-SDC
Collocation nodes	M	3	3
SDC iterations	K	3	3
Time steps	NT	5000	5000
Mesh resolution	NZ	5000	5000
Particle count	NQ	$2 \cdot 10^5$	$2 \cdot 10^5$
End time	T_E	10	10

proved sufficient in capturing the growing and saturated dynamics in the shortest amount of time.

Table 6.9: Simulation setup parameters for the Landau damping work-precision study.

Parameter	**Key**	**Weak**	**Strong**
Integrator	-	[Boris-SDC (M=3, K=2), Boris]	-
Time steps	NT	$[10, 20, 40, 50, 80, 100, 200, 400, 500, 1000]$	-
Mesh resolution	NZ	$[10, 100, 1000]$	-
Particle count	NQ	$2 \cdot 10^5$	-
End time	T_E	10	-

Figure 6.11 presents the progression of work-precision for both Boris and Boris-SDC at 4 sampled times of the weak case. The effect of the simulated plasma starting in a thermal state can be observed immediately. The spatial error threshold at the very beginning of the simulation (1 second) is now about two orders of magnitude higher for even the highest mesh resolution, 10^{-4} vs 10^{-6} for the two-stream instability. The reduced spatial accuracy significantly narrows the range in

which the convergence of the particle time integration is visible. Clear fourth-order vs. second-order convergence for Boris-SDC vs. Boris is only seen for the initial decrease(s) in time-step size. Boris-SDC is still seen to be more accurate than the equivalent Boris simulation at any time-step size above the spatial error saturation level.

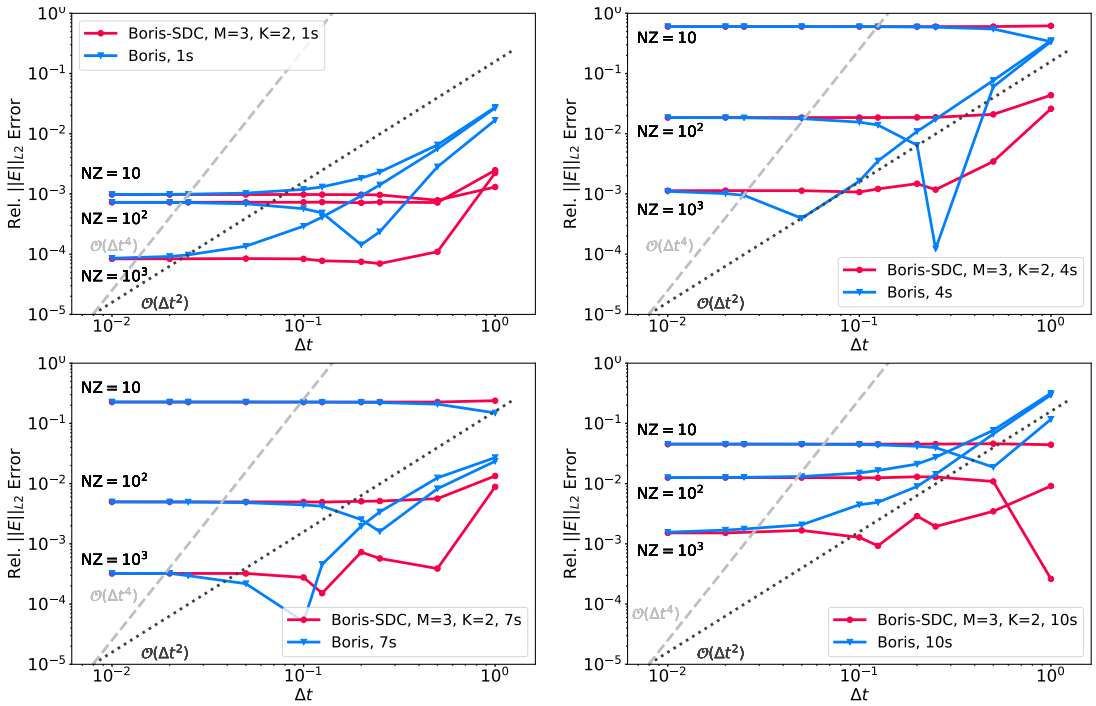

Figure 6.11: Convergence of the error in E-field norm of Boris and Boris-SDC for weak Landau damping at 1, 4, 7 and 10s simulation time.

In terms of computational cost however (Figure 6.12), Boris-SDC suffers from the reduced spatial accuracy. Even at the very beginning of the simulation, Boris-SDC barely overtakes Boris due to the higher convergence order before the error limit is reached.

Interestingly, the outlook is as good or better in the strong (Figure 6.13) case, the opposite from what was expected after the two-stream instability results. Here the expected orders of convergence, or close thereto, are observed throughout the simulation.

In terms of cost, Boris-SDC shows gains in performance for very marginal areas. Deviations like those from the previous results make the picture unclear for

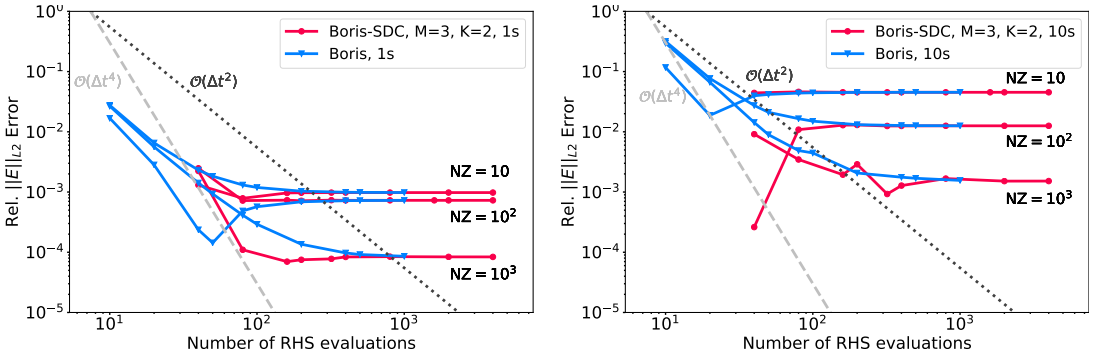

Figure 6.12: Performance comparison of Boris and Boris-SDC for weak Landau damping at 1 and 10s simulation time.

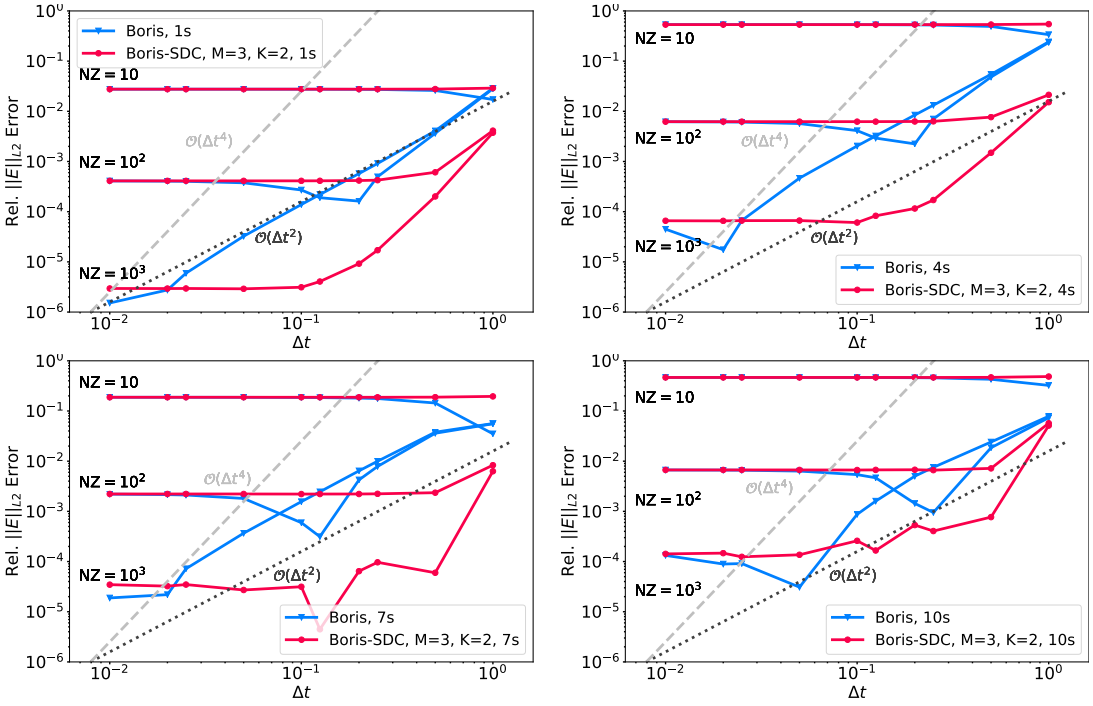

Figure 6.13: Convergence of the error in E-field norm of Boris and Boris-SDC for strong Landau damping at 1, 4, 7 and 10s simulation time.

the crucial cross-over zone between the integrators. Considering the comparative work-precision at 8 seconds however, it appears that these marginal gains persist throughout the simulation.

Results were broadly similar to the two-stream instability for both Landau damping cases, but with higher spatial error saturation overall. The added error is likely due to the thermalised nature of the problem with the resulting dynamics non-linear and rapidly oscillating from the start of the simulation. The resulting higher spatial error saturation significantly reduced or eliminated the possibility for superior

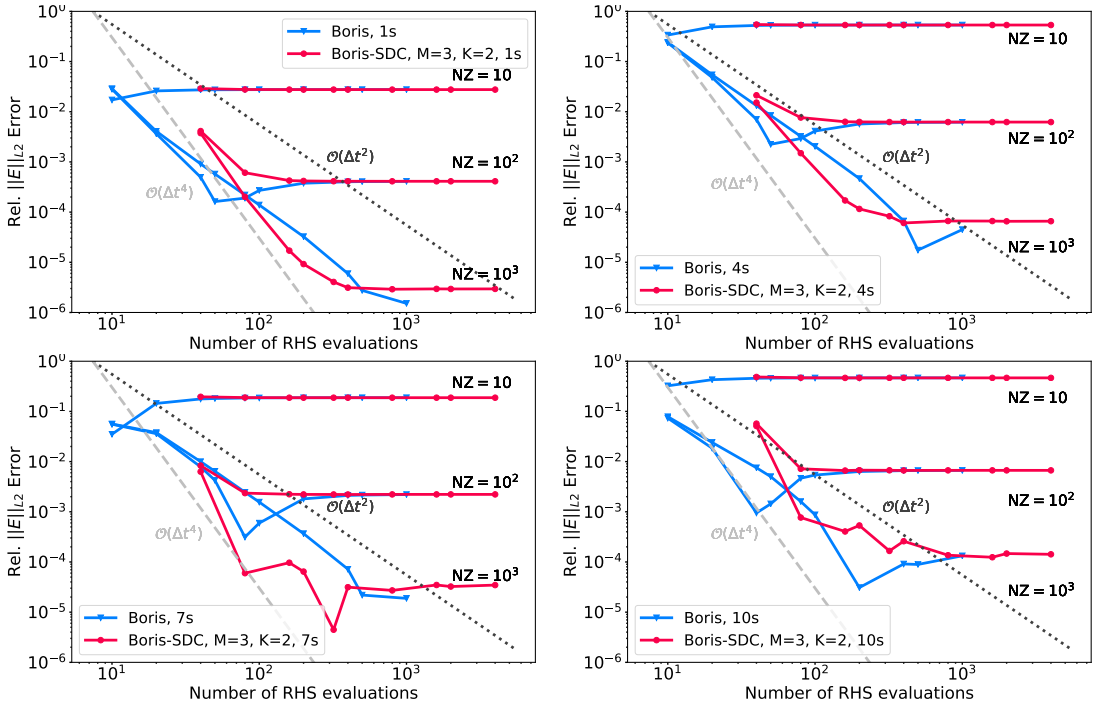

Figure 6.14: Performance comparison of Boris and Boris-SDC for strong Landau damping at 1, 4, 7 and 10s simulation time.

Boris-SDC computational efficiency, even for the initial sample times. In terms of error for a fixed time-step size however, Boris-SDC again proved more accurate than classic Boris, wherever spatial saturation was not yet reached. As a result, Boris-SDC reached the point of spatial error saturation at time-step sizes an order of magnitude larger than standard Boris.

Comparing the two sets of results, the cross-over in convergence between the two integrators occurred at around the same error magnitude for both the two-stream instability and Landau damping results ($\sim 10^{-3}$). This was expected to happen provided the particle integrators exhibited their expected convergence orders and error at the coarsest time-step was approximately the same for both problems. With this in mind, it was clear that for similar sets of initial errors exhibited by Boris and Boris-SDC, the cross-over points in performance must also be similar. If the error of the spatial solution is higher than this point, no performance gain will be found in using Boris-SDC. Conversely, increasing the accuracy of the spatial solution should eventually reveal a region in the space of error vs. computational effort where

Boris-SDC is superior.

6.3 Conclusions

Work-precision studies were conducted for two electrostatic problems using the custom project code KPPS. The two-stream instability and Landau damping were chosen for cold and hot plasma tests respectively. Both problems were studied for plasmas both weakly and strongly perturbed from equilibrium for a range of time-steps and mesh resolutions. In all cases, Boris-SDC demonstrated higher order convergence wherever the temporal error was above the saturation point of spatial error. For any of these time-step sizes Boris-SDC also produced lower errors than classic Boris and produced the same error even below the spatial saturation. The increased accuracy was generally not sufficient to offset the increased computational cost of SDC however and clear gains in computational efficiency were only seen in the linear growth regime of the two-stream instability. Note again, that the accuracy of Boris-SDC was limited by the spatial error saturation, which can be lowered by use of higher resolutions or higher order spatial methods.

This outcome was expected from the preliminary results in Chapter 5 which showed the same performance trade-off for the single particle oscillator case. From the above results, this same performance trade-off also appears for real ESPIC, with dynamic coupling of particles and field. Whether or not Boris-SDC is then useful in practice for ESPIC depends on the achievable spatial error, the accuracy requirements for the application and whether any increased accuracy beyond those requirements is desirable.

Chapter 7

Summary

The use of Boris-SDC as particle integrator in PIC schemes was investigated in this thesis through comparison with results produced by the classic Boris integrator. Relative performance of the two integrators was evaluated for electrostatic problems using work-precision studies of PIC codes implementing both integrators. The feasibility of replacing Boris in existing schemes was investigated in the implementation process of Boris-SDC and the effects of lower order spatial solutions on the particle integration were quantified. Finally, a relativistic generalisation of the integrator was formulated and implemented within the existing EMPIC code Runko followed by initial verification tests.

The following section contains a final discussion on the conclusions and contributions made during the project, across all tests and chapters. Recommendations are then given for future work, including both new investigations building on the sum of contributions here as well as unpursued opportunities identified along the way.

7.1 Conclusions

Development of a custom ESPIC code "KPPS", capable of using Boris-SDC, velocity-Verlet and leapfrog integration was detailed in Chapter 5. Key differences were identified in the operational flow for ESPIC depending on the need for time-synchronised vs time-staggered and single-stage vs. multi-stage particle integration. These were the order in which position and velocity are updated as well the interaction between particle and field solver operations. The project code KPPS was constructed with these needs in mind, but the work highlighted potential issues in implementing Boris-SDC in existing PIC codes, particularly those specifically designed for leapfrog Boris. If an existing PIC code is programmed with a specific order of operations in mind, it might not be feasible making the necessary modifications to allow for swapped position/velocity update order and multiple field solutions per time-step.

Using KPPS, the impact of combining lower order spatial solvers with higher order time-integration was shown for a simple oscillator case. The addition of increasing layers of approximation on the electric field value provided to the particle integrator was associated with an increasing level of spatial error saturation. The result is a new spatial error threshold, which time discretisation error converges towards instead of machine precision. As the saturation increases, the accuracy of the overall PIC simulation constrained to higher and higher levels of error which erodes work-adjusted performance of high-order time integrators is relative to lower order integrators. Nonetheless, the combination of spatial with temporal solution was not seen to inhibit the high-order convergence of Boris-SDC where temporal error was higher than spatial error. The same behaviour was seen for actual ESPIC applications in Chapter 6, where two well-known 1D problems were studied: The two-stream instability and Landau damping. The key difference between the ESPIC studies of Chapter 6 and the initial studies of Chapter 5 were in the inclusion of particle to field coupling and much larger numbers of particles. These more complicated test cases did exhibit increased spatial error at similar grid resolutions, but

high-order convergence was readily demonstrated for both applications, weakly or strongly perturbed.

In the linear growth/damping regime, spatial error saturation was close to the results from Chapter 5 and some computational efficiency gain was seen for Boris-SDC over classic Boris, most pronounced for the two-stream instability. In all cases tested, Boris-SDC produced lower error at any given time-step size before the saturation point, but the increased accuracy could not consistently offset the added computational cost. As a result, the performance gains were mostly lost as spatial saturation increased during the course of each simulation.

While performance gain was not consistently achieved, neither was Boris-SDC noticeably slower than classic Boris. Future performance improvements to SDC [22, 38, 91] could potentially make Boris-SDC faster than Boris more consistently in the critical range where spatial error saturation has not yet been reached. In terms of computational efficiency, Boris-SDC was found to achieve parity with standard Boris for all but the lowest spatial resolutions. If maximum possible accuracy is sought, a combination of Boris-SDC and high spatial accuracy should thus be an attractive proposition. If spatial accuracy is high enough, Boris-SDC should also outperform classic Boris, making it ideal for applications wherein temporal error dominates. Note that these results concern mainly the interplay of spatial and temporal error with similar investigation of the particle count and associated momentum error left for future work. Whether Boris-SDC is then useful for current ESPIC applications thus depends on the available computational resources, accuracy requirements and problem sensitivity to spatial error. Only two test cases were investigated in this study, and other applications may exhibit the right balance of spatial and temporal error to make Boris-SDC a performance competitive alternative to the standard leapfrog Boris.

Preparatory work to facilitate relativistic and EMPIC results were conducted in the latter half of Chapter 5. A novel generalisation of Boris-SDC to the relativistic

regime was derived which ensures convergence of the relativistic factor. High-order time convergence and gains in computational efficiency vs. classic Boris were then demonstrated using a single-particle Penning trap, tested at relativistic velocities up to 99.9975% lightspeed.

The relativistic Boris-SDC integrator was then implemented within the open-source EMPIC code Runko, using the same modifications to the operational flow discussed for ESPIC codes. To verify the working Runko/Boris-SDC code, a relativistic Penning trap setup was investigated using fixed mesh-defined fields interpolated to a single particle. The Runko implementation of Boris-SDC again demonstrated high-order convergence and superior computational efficiency even in the presence of round-off error from the Runko field interpolator. Due to limitations for mesh and domain size definitions in Runko, it was not possible to compare across multiple mesh resolutions and the spatial error of the interpolator could not be compared to previous results. Note that the open-source and modular nature of Runko is what made the necessary modifications to implement a time-synchronised, multi-stage particle integrator possible. For commercial or heavily interconnected codes, implementing Boris-SDC as a Boris alternative could prove very difficult depending on the code structure.

In summary, the three main original contributions of this research are:

1. Identifying the necessary modifications to implement Boris-SDC and similar integrators within traditional PIC schemes and demonstrating working implementations of Boris-SDC in ESPIC and Runko.

2. Demonstrating that Boris-SDC retains high-order time convergence and high accuracy in ESPIC, where spatial error does not dominate. Additionally, sufficiently low spatial error allows Boris-SDC to be more computationally efficient than classic Boris due to high-order temporal convergence, but this was generally not the case using first and second order spatial methods for non-linear

dynamics.

3. Extending Boris-SDC to the relativistic regime and demonstrating that the relativistic integrator retains high-order time convergence and favourable performance vs. classic Boris.

7.2 Further Work

During the course of the project, several open problems were discovered and left unexplored to focus on the main objectives. In terms of questions w.r.t. Boris-SDC itself, the relative performance of 2015 vs. 2019 formulations of Boris-SDC has yet to be studied. The gain in order of accuracy per SDC sweep has yet to be formally investigated, with the 2015 formulation used for the ESPIC studies generally observed to gain 2 orders per sweep vs. 1 for the 2019 formulation. The picture was further complicated by the ability to use either simple initialisation or a predictor for Boris-SDC. The predictor provides an initial approximation of node values via Boris, which was found to enhance the convergence rate of the 2019 formulation to 2 orders per sweep in the relativistic Penning trap in tests not documented in this thesis. The performance in terms of computational efficiency between the two formulations, with and without predictor initialisation should be studied further, to determine if either formulation outperforms the other. As an extension of this, no relativistic generalisation has been formulated for 2015 Boris-SDC, which would be needed for comparison in the relativistic EMPIC regime.

The options for further work on the topic of PIC/Boris-SDC are varied. Momentum error, stemming from the discrete particle sampling of the plasma in PIC as per [68], was neglected in favour of studying spatial and temporal errors. The momentum error is more visible close to thermal equilibrium of the particles, as seen in the weak Landau damping test, where particle noise drowned out the instability at low perturbation magnitudes ($\alpha \approx 0.01$). Similar studies to those conducted in this

thesis should be done on the balance of momentum to spatial and temporal error.

The effect of spatial error saturation on time integrator performance was demonstrated for three separate problems, but further investigation would serve to make clearer the picture and validate the results. Landau damping proved to be the worst performing case study for Boris-SDC and it might be worth exploring a larger range of perturbation magnitudes as well as spatial and particle resolutions. Combined with an investigation into momentum error, further study of Landau damping could yield better insight into the effect of the inherent randomness of thermal velocity distributions on the accuracy and performance of PIC/Boris-SDC

More electrostatic tests should be conducted, including 2D or 3D problems. The problem of spatial error saturation is likely to be more pronounced when the available computational effort must be divided between more than one dimension. Only relatively short time-scales were included in the work-precision studies due to the need for multiple simulations of varying resolutions. The spatial error was seen to rise with time in both problems studied but the comparative performance of Boris-SDC vs. Boris for longer time-scales should be investigated, including a comparison of long-term energy stability.

The work-precision of EMPIC/Boris-SDC should be investigated in depth to quantify the spatial/temporal error trade-off for electromagnetic problems. In particular, the problem of mismatched accuracy order for the time integration of the fields versus the particles needs to be investigated, as EMPIC requires both. Only the fundamental implementation of Boris-SDC functionality within EMPIC via Runko was realised in this thesis. The use of high-order particle integration as part of a second order EMPIC scheme presents issues not seen the ESPIC case, such as dual time advancement of field and particles. It is not known whether the high-order accuracy of Boris-SDC will contribute meaningfully to the overall PIC solution when the field integrator is only second order accurate in time. The Weibel instability presents an ideal test case for EMPIC/Boris-SDC, being the electromagnetic analogue to the

two-stream instability.

Performance gains for Boris-SDC were found to require a high degree of spatial accuracy. The simulations produced for this study included the highest spatial resolutions feasible on a contemporary 8-core workstation and further increases in spatial resolution would require week-long simulation times, significant code optimisation, high performance computing or all of the above. This level of spatial fidelity using second order methods might be too costly or just not useful for those interested in Boris-SDC. A better option would be combining PIC/Boris-SDC with similar high-order methods for field interpolation and solution, which would also serve to alleviate the potential problems of mismatching time convergence order for the field and particle integrators. The problem is mirrored in the results of Benedetti et al. [6] where potential performance gain for higher order field solution in ALaDyn is held back by second-order particle integrator. Combining Boris-SDC with a PIC code like ALaDyn would be a natural solution to performance constraints observed in isolated tests of both higher order field and particle integrators. If the higher order spatial schemes used have a similar performance outlook to those observed for Boris-SDC in this thesis, an overall higher order PIC scheme could be constructed which does not restrict the high-order time convergence of Boris-SDC.

For current physical applications, PIC/Boris-SDC could be applied to problems where spatial error is low compared to temporal error. For instance, problems that require PIC simulation of linear growth regimes over long time frames would allow Boris-SDC to excel based on the performance observed in Chapter 6. One example of such is the study of TeV blazar beams impacting the intergalactic medium, which has been modelled as a long phase of linear growth in energy transfer between beam and plasma, culminating in a possible instability onset [70, 64, 69]. The enhanced accuracy and performance of Boris-SDC during PIC simulation of the long linear growth would be a benefit to these studies, as the existence and potential characteristics of the instability depend strongly on the growth history.

Bibliography

[1] Hirotade Abe, Natsuhiko Sakairi, Ryohei Itatani, and Hideo Okuda. High-order spline interpolations in the particle simulation. *Journal of computational physics*, 63(2):247–267, 1986.

[2] Milton Abramowitz and Irene A Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. National Bureau of Standards Applied Mathematics Series 55. Tenth Printing.* ERIC, 1972.

[3] Mariano Andrenucci, Leonardo Biagioni, and Andrea Passaro. *PIC/DSMC models for Hall effect thruster plumes: Present status and ways forward.* AIAA, 2002.

[4] T D Arber, K Bennett, C S Brady, A Lawrence-Douglas, M G Ramsay, N J Sircombe, P Gillies, R G Evans, H Schmitz, A R Bell, and C P Ridgers. Contemporary particle-in-cell approach to laser-plasma modelling. *Plasma Physics and Controlled Fusion*, 57(11):113001, 2015. URL http://stacks.iop.org/0741-3335/57/i=11/a=113001.

[5] Blanca Ayuso de Dios and Soheil Hajian. High order and energy preserving discontinuous galerkin methods for the vlasov-poisson system. *arXiv preprint arXiv:1209.4025*, 2012.

[6] Carlo Benedetti, Andrea Sgattoni, Giorgio Turchetti, and Pasquale Londrillo. Aladyn: A high-accuracy pic code for the maxwell-vlasov equations. *IEEE Transactions on plasma science*, 36(4):1790–1798, 2008.

[7] Kallol Bera, Shahid Rauf, and Ken Collins. Pic-mcc/fluid hybrid model for low pressure capacitively coupled o 2 plasma. In *AIP Conference Proceedings*, volume 1333, pages 1027–1032. American Institute of Physics, 2011.

[8] Charles K Birdsall and Bruce A Langdon. *Plasma Physics Via Computer Simulation*. McGraw-Hill, Inc., New York, NY, 1985.

[9] Joseph D Blahovec, Lester A Bowers, John W Luginsland, Gerald E Sasser, and John J Watrous. 3-d icepic simulations of the relativistic klystron oscillator. *IEEE transactions on plasma science*, 28(3):821–829, 2000.

[10] Roger D Blandford and Kip S Thorne. *Applications of classical physics*. 2008.

[11] Jay P Boris. Relativistic plasma simulation-optimization of a hybrid code. In *Proc. Fourth Conf. Num. Sim. Plasmas, Naval Res. Lab, Wash. DC*, pages 3–67, 1970.

[12] Rod W Boswell and Ian J Morey. Self-consistent simulation of a parallel-plate rf discharge. *Applied physics letters*, 52(1):21–23, 1988.

[13] K J Bowers, B J Albright, L Yin, W Daughton, V Roytershteyn, B Bergen, and T J T Kwan. Advances in petascale kinetic plasma simulation with vpic and roadrunner. *Journal of Physics: Conference Series*, 180(1):012055, 2009. URL http://stacks.iop.org/1742-6596/180/i=1/a=012055.

[14] Oscar Buneman. Dissipation of currents in ionized media. *Physical Review*, 115 (3):503, 1959.

[15] José Canosa, Jenö Gazdag, JE Fromm, and BH Armstrong. Electrostatic oscillations in plasmas with cutoff distributions. *The Physics of Fluids*, 15(12): 2299–2305, 1972.

[16] Francis F Chen. *Introduction to plasma physics*. Springer, New York, 1974.

[17] Chio-Zong Cheng and Georg Knorr. The integration of the vlasov equation in configuration space. *Journal of Computational Physics*, 22(3):330–351, 1976.

[18] John Dawson. One-dimensional plasma model. *The Physics of Fluids*, 5(4): 445–459, 1962.

[19] John M Dawson. Thermal relaxation in a one-species, one-dimensional plasma. *The Physics of Fluids*, 7(3):419–425, 1964.

[20] Beniamino Di Martino, Sergio Briguglio, Gregorio Vlad, and Piero Sguazzero. Parallel pic plasma simulation through particle decomposition techniques. *Parallel Computing*, 27(3):295–314, 2001.

[21] James W Eastwood. The virtual particle electromagnetic particle-mesh method. *Computer Physics Communications*, 64(2):252–266, 1991.

[22] Matthew Emmett and Michael L. Minion. Toward an efficient parallel in time method for partial differential equations. *Communications in Applied Mathematics and Computational Science*, 7:105–132, 2012. doi: 10.2140/camcos.2012.7.105.

[23] EPCC. Optimisation of the epoch laser-plasma simulation code, 2020. URL https://www.archer.ac.uk/community/eCSE/eCSE03-01/eCSE03-01.php. [Accessed: 9 October 2020].

[24] T Zh Esirkepov. Exact charge conservation scheme for particle-in-cell simulation with an arbitrary form-factor. *Computer Physics Communications*, 135 (2):144–153, 2001.

[25] Brendan B Godfrey and Jean-Luc Vay. Improved numerical cherenkov instability suppression in the generalized pstd pic algorithm. *Computer Physics Communications*, 196:221–225, 2015.

[26] Marcel Goossens. *An introduction to plasma astrophysics and magnetohydrodynamics*. Dordrect: Kluwer Academic Publishers, 2003.

[27] VP Gopinath, John Paul Verboncoeur, and Charles K Birdsall. Multipactor

electron discharge physics using an improved secondary emission model. *Physics of Plasmas*, 5(5):1535–1540, 1998.

[28] David J Griffiths. *Introduction to electrodynamics 4th Ed.* Cambridge University Press, 2017.

[29] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer Verlag, 1993.

[30] Ernst Hairer and Christian Lubich. Symmetric multistep methods for charged-particle dynamics. *SMAI J. Comput. Math., to appear*, 2017.

[31] Ernst Hairer and Christian Lubich. Energy behaviour of the boris method for charged-particle dynamics. *BIT Numerical Mathematics*, 58(4):969–979, 2018.

[32] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[33] Troels Haugbølle, Jacob Trier Frederiksen, and Åke Nordlund. photon-plasma: A modern high-order particle-in-cell code. *Physics of Plasmas*, 20(6):062904, 2013.

[34] Yang He, Yajuan Sun, Jian Liu, and Hong Qin. Higher order volume-preserving schemes for charged particle dynamics. *Journal of Computational Physics*, 305:172 – 184, 2016. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2015.10.032. URL http://www.sciencedirect.com/science/article/pii/S0021999115007081.

[35] Yang He, Yajuan Sun, Hong Qin, and Jian Liu. Hamiltonian particle-in-cell methods for vlasov-maxwell equations. *Physics of Plasmas*, 23(9):092108, 2016.

[36] Adam V Higuera and John R Cary. Structure-preserving second-order integration of relativistic charged particle trajectories in electromagnetic fields. *Physics of Plasmas*, 24(5):052104, 2017.

[37] Roger W Hockney and James W Eastwood. *Computer simulation using particles*. crc Press, 1988.

[38] Jingfang Huang, Jun Jia, and Michael Minion. Accelerating the convergence of spectral deferred correction methods. *Journal of Computational Physics*, 214 (2):633–656, 2006. doi: 10.1016/j.jcp.2005.10.004.

[39] GB Jacobs and Jan S Hesthaven. High-order nodal discontinuous galerkin particle-in-cell method on unstructured grids. *Journal of Computational Physics*, 214(1):96–121, 2006.

[40] GB Jacobs and Jan S Hesthaven. Implicit–explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning. *Computer Physics Communications*, 180(10):1760–1767, 2009.

[41] Boris B Kadomtsev. Disruptive instability in tokamaks. *FizPl*, 1:710–715, 1975.

[42] Michael Kraus, Katharina Kormann, Philip J Morrison, and Eric Sonnendrücker. Gempic: Geometric electromagnetic particle-in-cell methods. *Journal of Plasma Physics*, 83(4), 2017.

[43] FM Laggner, A Diallo, M Cavedon, and Egemen Kolemen. Inter-elm pedestal localized fluctuations in tokamaks: Summary of multi-machine observations. *Nuclear Materials and Energy*, 19:479–486, 2019.

[44] L. D. Landau. On the vibrations of the electronic plasma. *J. Phys.(USSR)*, 10: 25–34, 1946. [Zh. Eksp. Teor. Fiz.16,574(1946)].

[45] A. B. Langdon. Evolution of particle-in-cell plasma simulation. *IEEE Transactions on Plasma Science*, 42(5):1317–1320, May 2014. ISSN 0093-3813. doi: 10.1109/TPS.2014.2314615.

[46] Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*, volume 14. Cambridge university press, 2004.

[47] Ting Li and Bin Wang. Arbitrary-order energy-preserving methods for charged-particle dynamics. *Applied Mathematics Letters*, 100:106050, 2020.

[48] Paulett C Liewer and Viktor K Decyk. A general concurrent algorithm for plasma particle-in-cell simulation codes. *Journal of Computational Physics*, 85 (2):302–322, 1989.

[49] P Londrillo, C Benedetti, A Sgattoni, and G Turchetti. Charge preserving high order pic schemes. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 620(1):28–35, 2010.

[50] JW Luginsland, TA Antonsen Jr, JP Verboncoeur, RW Lemke, L Ludeking, PJ Mardahl, AT Lin, YY Lau, and JD Blahovec Jr. Computational techniques. *High-Power Microwave Sources and Technologies*, pages 376–437, 2001. ed RJ Barker and E Schamiloglu.

[51] Ingrid Mann, Asta Pellinen-Wannberg, Edmond Murad, Olga Popova, Nicole Meyer-Vernet, Marlene Rosenberg, Tadashi Mukai, Andrzej Czechowski, Sonoyo Mukai, Jana Safrankova, et al. Dusty plasma effects in near earth space and interplanetary medium. *Space science reviews*, 161(1-4):1–47, 2011.

[52] Akinobu Matsuyama and Masaru Furukawa. High-order integration scheme for relativistic charged particle motion in magnetized plasmas with volume preserving properties. *Computer Physics Communications*, 220:285–296, 2017.

[53] RL Morse and CW Nielson. Numerical simulation of the weibel instability in one and two dimensions. *The Physics of Fluids*, 14(4):830–840, 1971.

[54] Takashi Nakamura and Takashi Yabe. Cubic interpolated propagation scheme for solving the hyper-dimensional vlasov-poisson equation in phase space. *Computer Physics Communications*, 120(2-3):122–154, 1999.

[55] Valery M Nakariakov. Mhd oscillations in solar and stellar coronae: Current results and perspectives. *Advances in Space Research*, 39(12):1804–1813, 2007.

[56] K Nanbu. Theory of cumulative small-angle collisions in plasmas. *Physical Review E*, 55(4):4642, 1997.

[57] J. Nättilä. Runko: Modern multi-physics toolbox for simulating plasma. *arXiv e-prints*, art. arXiv:1906.06306, Jun 2019.

[58] L. Patacchini and I.H. Hutchinson. Explicit time-reversible orbit integration in particle in cell codes with static homogeneous magnetic field. *Journal of Computational Physics*, 228(7):2604–2615, 2009. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2008.12.021. URL https://www.sciencedirect.com/science/article/pii/S0021999108006608.

[59] Alexander Pukhov. Particle-in-cell codes for plasma-based particle acceleration. *arXiv preprint arXiv:1510.01071*, 2015.

[60] Ji Qiang. A fast numerical integrator for relativistic charged particle tracking. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 867:15 – 19, 2017. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2017.04.015. URL http://www.sciencedirect.com/science/article/pii/S0168900217304655.

[61] Ji Qiang. High order numerical integrators for relativistic charged particle tracking. *arXiv preprint arXiv:1702.04486*, 2017.

[62] Hong Qin, Shuangxi Zhang, Jianyuan Xiao, Jian Liu, Yajuan Sun, and William M Tang. Why is boris algorithm so good? *Physics of Plasmas*, 20(8): 084503, 2013.

[63] M. Quandt, C. Munz, and R. Schneider. A high order relativistic particle push method for pic simulations. *The 30th International Electric Propulsion Conference*, pages 1–9, 2007. cited By 1.

[64] Iman Rafighi, Sergei Vafin, M Pohl, and J Niemiec. Plasma effects on relativistic pair beams from tev blazars: Pic simulations and analytical predictions. *Astronomy & Astrophysics*, 607:A112, 2017.

[65] Bart Ripperda, Fabio Bacchini, Jannis Teunissen, Chun Xia, Oliver Porth, Lorenzo Sironi, Giovanni Lapenta, and Rony Keppens. A comprehensive comparison of relativistic particle integrators. *The Astrophysical Journal Supplement Series*, 235(1):21, 2018.

[66] James A Rossmanith and David C Seal. A positivity-preserving high-order semi-lagrangian discontinuous galerkin scheme for the vlasov-poisson equations. *Journal of Computational Physics*, 230(16):6203–6232, 2011.

[67] Daniel Ruprecht and Robert Speck. Spectral deferred corrections with fast-wave slow-wave splitting. *SIAM Journal on Scientific Computing*, 38(4):A2535–A2557, 2016.

[68] Mohamad Shalaby, Avery E Broderick, Philip Chang, Christoph Pfrommer, Astrid Lamberts, and Ewald Puchwein. Sharp: A spatially higher-order, relativistic particle-in-cell code. *The Astrophysical Journal*, 841(1):52, 2017.

[69] Mohamad Shalaby, Avery E Broderick, Philip Chang, Christoph Pfrommer, Ewald Puchwein, and Astrid Lamberts. The growth of the longitudinal beam–plasma instability in the presence of an inhomogeneous background. *Journal of Plasma Physics*, 86(2), 2020.

[70] Lorenzo Sironi and Dimitrios Giannios. Relativistic pair beams from tev blazars: a source of reprocessed gev emission rather than intergalactic heating. *The Astrophysical Journal*, 787(1):49, 2014.

[71] T Stindl, J Neudorfer, A Stock, M Auweter-Kurtz, CD Munz, S Roller, and Rainer Schneider. Comparison of coupling techniques in a high-order discontinuous galerkin-based particle-in-cell solver. *Journal of Physics D: Applied Physics*, 44(19):194004, 2011.

[72] P. H. Stoltz, J. R. Cary, G. Penn, and J. Wurtele. Efficiency of a boris-like integration scheme with spatial stepping. *Phys. Rev. ST Accel. Beams*,

5:094001, Sep 2002. doi: 10.1103/PhysRevSTAB.5.094001. URL https://link.aps.org/doi/10.1103/PhysRevSTAB.5.094001.

[73] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Physics Letters A*, 146(6):319–323, 1990.

[74] William Taitano, Dana Knoll, and Anil Prinja. Vlasov-fokker-planck simulation of a collisional ion-electron shockwave. *APS*, 54:CP8–091, 2012.

[75] Tomonor Takizuka and Hirotada Abe. A binary collision model for plasma simulation with a particle code. *Journal of Computational Physics*, 25(3):205–219, 1977.

[76] Molei Tao. Explicit high-order symplectic integrators for charged particles in general electromagnetic fields. *Journal of Computational Physics*, 327:245–251, 2016.

[77] Tech-X Corporation. Plasma discharge solutions, 2020. URL https://www.txcorp.com/success-stories/vsim/plasma-discharges. [Accessed: 9 October 2020].

[78] Tech-X Corporation. High power hall thruster channel plasma simulations and lifetime predictions, 2020. URL https://txcorp.com/success-stories/vsim/plasma-discharges/high-power-hall-thruster-channel-plasma-simulations-and-lifetime-predictio [Accessed: 9 October 2020].

[79] Krasymyr Tretiak and Daniel Ruprecht. An arbitrary order time-stepping algorithm for tracking particles in inhomogeneous magnetic fields. *Journal of Computational Physics: X*, 4:100036, 2019.

[80] Vahid Vahedi and G DiPeso. Simultaneous potential and circuit solution for two-dimensional bounded plasma simulation codes. *Journal of Computational Physics*, 131(1):149–163, 1997.

[81] Vahid Vahedi and Maheswaran Surendra. A monte carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges. *Computer Physics Communications*, 87(1-2):179–198, 1995.

[82] J-L Vay. Simulation of beams or plasmas crossing at relativistic velocity. *Physics of Plasmas*, 15(5):056701, 2008.

[83] John P Verboncoeur. Particle simulation of plasmas: review and advances. *Plasma Physics and Controlled Fusion*, 47(5A):A231, 2005.

[84] John P Verboncoeur, A Bruce Langdon, and NT Gladd. An object-oriented electromagnetic pic code. *Computer Physics Communications*, 87(1-2):199–211, 1995.

[85] John Paul Verboncoeur, Maria Virginia Alves, Vahid Vahedi, and Charles Kennedy Birdsall. Simultaneous potential and circuit solution for 1d bounded plasma particle simulation codes. *Journal of Computational Physics*, 104(2):321–328, 1993.

[86] J.P. Verboncoeur. Symmetric spline weighting for charge and current density in particle simulation. *Journal of Computational Physics*, 174(1):421 – 427, 2001. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.2001.6923. URL http://www.sciencedirect.com/science/article/pii/S0021999101969232.

[87] P Verboncoeur. A digital filtering scheme for particle codes in curvilinear coordinates. In *Pulsed Power Plasma Science, 2001. IEEE Conference Record-Abstracts*, page 270. IEEE, 2001.

[88] John Villasenor and Oscar Buneman. Rigorous charge conservation for local electromagnetic field solvers. *Computer Physics Communications*, 69(2-3):306–316, 1992.

[89] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren

Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020. doi: https://doi.org/10.1038/s41592-019-0686-2.

[90] A Vlasov. The vibrational properties of an electron gas. *Soviet Physics Uspekhi*, 10(6):721, 1968. URL http://stacks.iop.org/0038-5670/10/i=6/a=R01.

[91] Martin Weiser. Faster SDC convergence on non-equidistant grids by DIRK sweeps. *BIT Numerical Mathematics*, 55(4):1219–1241, 2014. doi: 10.1007/s10543-014-0540-y.

[92] Mathias Winkel, Robert Speck, and Daniel Ruprecht. A high-order boris integrator. *Journal of Computational Physics*, 295:456 – 474, 2015. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2015.04.022. URL http://www.sciencedirect.com/science/article/pii/S0021999115002685.

[93] Yinhua Xia, Yan Xu, and Chi-Wang Shu. Efficient time discretization for local discontinuous galerkin methods. *Discrete and Continuous Dynamical Systems Series B*, 8(3):677, 2007.

[94] Jianyuan Xiao, Hong Qin, Jian Liu, Yang He, Ruili Zhang, and Yajuan Sun. Explicit high-order non-canonical symplectic particle-in-cell algorithms for vlasov-maxwell systems. *Physics of Plasmas*, 22(11):112504, 2015. doi: 10.1063/1.4935904. URL https://doi.org/10.1063/1.4935904.

[95] Jianyuan Xiao, Hong Qin, and Jian Liu. Structure-preserving geometric particle-in-cell methods for vlasov-maxwell systems. *Plasma Science and*

Technology, 20(11):110501, sep 2018. doi: 10.1088/2058-6272/aac3d1. URL https://doi.org/10.1088%2F2058-6272%2Faac3d1.

[96] Haruo Yoshida. Construction of higher order symplectic integrators. *Physics letters A*, 150(5-7):262–268, 1990.