Master's thesis

Master's Programme in Theoretical and Computational Methods

Relativistic Beam Instabilities with GPU-Accelerated Particle-in-Cell Method

Miro Palmu

December 18, 2025

Supervisor(s):	Assoc. Prof. Joonas Nättilä
Examiner(s):	Assoc. Prof. Joonas Nättilä Academy Fellow Tuomo Salmi

UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Degree programme Master's Programme in Theoretical and Computational Methods

Tekijä — Författare — Author

Miro Palmu

Työn nimi — Arbetets titel — Title

Relativistic Beam Instabilities with GPU-Accelerated Particle-in-Cell Method

Työn laji — Arbetets art — Level Master's thesis	Aika — Datum — Month and year December 18, 2025	Sivumäärä — Sidantal — Number of pages 34

Tiivistelmä — Referat — Abstract

Plasma dynamics are modeled with the Vlasov-Maxwell equations. Due to the multiscale nature of plasmas and their numerous degrees of freedom, the Vlasov-Maxwell equations are highly nonlinear. Therefore, plasma dynamics are often studied using numerical methods. One commonly used approach is the particle-in-cell (PIC) method, in which the electromagnetic fields are discretized on a so-called Yee mesh while the particles are free to move continuously in space.

The PIC method consists of several components. Field propagation evolves the fields in time according to Maxwell's equations. To compute the Lorentz force at the particle locations, the fields must be interpolated from the discrete Yee mesh to the particle positions. With the Lorentz force, the particle velocities are updated, and these updated velocities are then used to update the particle positions. The moving particles generate a current, which must be deposited back onto the Yee mesh.

This thesis is based on the work carried out by the author to port the PIC simulation framework Runko to GPUs. The thesis presents the components of the PIC method in detail and demonstrates the correctness of Runko's ability to resolve the full set of electromagnetic Vlasov-Maxwell equations. Correctness is demonstrated through simulations of the beam-plasma filamentation instability, whose growth rate is shown to match the analytical growth rate obtained from first-order perturbation theory in the cold-fluid limit.

Avainsanat — Nyckelord — Keywords

particle-in-cell, plasma, beam instabilities, GPU

Säilytyspaikka — Förvaringsställe — Where deposited

Kumpulan kampuskirjasto, Gustaf Hällströmin katu 2, 00560 Helsinki

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

Plasma dynamics has been studied in many contexts, including astrophysics, laser physics, and fusion (Uzdensky & Rightley, 2014; Umstadter, 2003; Kirkwood et al., 2013). The dynamics are highly non-linear, and although there is a wealth of theoretical analysis, the complexity eventually becomes infeasible to handle analytically. Consequently, numerical methods are required to study many complex plasma phenomena. Today, high-performance computing simulations are performed on supercomputers to model the underlying physical systems with such accuracy that they can reveal new physical insights. One commonly used approach is the particle-in-cell (PIC) method, in which the electromagnetic fields are discretized on a grid while the particles are free to move continuously in space.

This thesis is based on the work carried out by the author to port the PIC simulation framework Runko (Nättilä, 2022) to GPUs. The Git repository of Runko is located at github.com/hel-astro-lab/runko. At the time of writing this thesis, the commits related to the GPU port reside on the `dev-v5` branch. All later references to file locations inside the Runko repository specifically refer to the repository state at the Git commit with the abbreviated hash `92d840e`[†].

In this chapter, the required background theory is presented, and in the following chapter, different aspects of PIC simulation are discussed. In the third chapter, the GPU port of Runko is used to simulate the electron beam-plasma filamentation instability, which is then compared to the analytical result in order to confirm physical accuracy of the port. The final chapter contains conclusions and a discussion of further development avenues for the GPU port of Runko.

1.1 Plasma

Plasma is a quasi-neutral gas containing enough free charges that long-range collective electromagnetic interactions dominate over interactions between individual particles (Pert, 2021; Hazeltine, 2018). This can occur, for example, when a gas is heated

[†] The full hash is `92d840eb20cba166d394d3b256d50b7335e984e9`.

sufficiently for its atoms to become ionized. Quasi-neutrality means that the gas is approximately electrically neutral:

$$\rho = \sum_s n_s q_s \approx 0\,, \tag{1.1}$$

where the sum is taken over all particle species species s, n_s denotes the number density, and q_s is the charge.

If a charge is introduced into a plasma in thermal equilibrium, the free charges quickly screen the resulting long-range electric field. This phenomenon is known as Debye screening (or Debye shielding). It gives rise to a characteristic length scale over which short-range Coulomb interactions can occur. This scale is defined by the Debye length, λ_D which is given by:

$$\lambda_D^{-2} = \sum_s \frac{n_s q_s^2}{\epsilon_0 k_B T_s}\,, \tag{1.2}$$

where T_s, n_s and q_s are the temperature, number density and charge of particle species s, respectively; k_B is the Boltzmann constant; and ϵ_0 is the vacuum permittivity.

For a plasma in thermal equilibrium, the scale of number density deviations from quasi-neutrality over a length L due to thermal fluctuations is $(\lambda_D/L)^2$ (Wiesemann, 2014; Hazeltine, 2018). Thus, we can express the quasi-neutrality condition for a plasma over the length scale L as the requirement that the Debye length is much smaller than L:

$$\lambda_D \ll L\,. \tag{1.3}$$

However, for the Debye length to be statistically valid, there must be enough particles inside the "Debye sphere": $N_D = \frac{4\pi}{3} n \lambda_D^3 \gg 1$. This requirement is usually expressed in terms of the plasma parameter, which is defined (up to a constant factor) as $\Lambda = n\lambda_D^3$ and written as (Kilpua et al., 2017):

$$\Lambda \gg 1\,. \tag{1.4}$$

1.2 Fundamental frequencies in plasma physics

There are a few characteristic frequencies that appear frequently in plasma physics. In a plasma, a small displacement of charges oscillates around the neutral configuration at a frequency known as the plasma frequency (Wiesemann, 2014; Pert, 2021; Hazeltine, 2018). For a particle species s, the classical plasma frequency is given by:

$$\omega_{\mathrm{p}s} = \sqrt{\frac{n_s q_s^2}{\epsilon_0 m_s}}\,. \tag{1.5}$$

For electron-ion systems, the electron plasma frequency is much larger than the ion plasma frequency; therefore, ion plasma oscillations can usually be neglected.

If a magnetic field is present in a plasma, the charges gyrate around the field lines at the cyclotron frequency also known as Larmor frequency. For a particle species s, the cyclotron frequency is given by:

$$\omega_B = \frac{q_s B}{m_s c \gamma}\,, \tag{1.6}$$

where B is the magnitude of the magnetic field. This motion is also known as cyclotron motion.

1.3 Kinetic plasma description

As seen from the condition in Equation (1.4), the number of particles in plasma is large. This makes modeling a plasma using only particles and Coulomb forces between them computationally unfeasible. In the kinetic description of plasma, the problem is addressed by embracing the large number of particles and statistically describing their collective behavior.

For a particle species s, the distribution function $f_s(\mathbf{x}, \mathbf{p}, t)$ describes the number of particles in a phase space region spanned by the time interval $[t, t+dt]$, spatial region $[\mathbf{x}, \mathbf{x}+d\mathbf{x}]$ and three-momentum region $[\mathbf{p}, \mathbf{p}+d\mathbf{p}]$ (Klimontovich, 1960; De Groot, 1980; Hazeltine, 2018). The particles lie on a mass-shell defined by the Lorentz factor $\gamma^{-1} = \sqrt{1-\frac{v^2}{c^2}}$, where v is the coordinate velocity and c is the speed of light. We assume that the distribution function is normalized such that

$$\int f_s(\mathbf{x}, \mathbf{p}, t) d\mathbf{p} = n_s(\mathbf{x}, t)\,, \tag{1.7}$$

where $n_s(\mathbf{x}, t)$ is the number density of a particle species s. Summing over all particle species, we obtain the total charge density of the plasma:

$$\rho(\mathbf{x}, t) = \sum_s q_s n_s(\mathbf{x}, t)\,. \tag{1.8}$$

Current density of the plasma is given by:

$$\mathbf{J}(\mathbf{x}, t) = \int \sum_s q_s \mathbf{v} f_s(\mathbf{x}, \mathbf{p}, t) d\mathbf{p}\,. \tag{1.9}$$

With these definitions of charge and current densities, we can determine the self-consistent electromagnetic fields using Maxwell's equations (in Gaussian-CGS units):

$$\nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t}\,, \qquad \nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{J} + \frac{1}{c}\frac{\partial \mathbf{E}}{\partial t}\,, \qquad \nabla \cdot \mathbf{E} = 4\pi\rho\,, \qquad \nabla \cdot \mathbf{B} = 0 \tag{1.10}$$

The evolution of the distribution function is governed by the Boltzmann equation:

$$\partial_t f_s + \mathbf{v} \cdot \nabla f_s + \frac{d\mathbf{p}}{dt} \cdot \nabla_{\mathbf{p}} f_s = \left(\frac{\partial f_s}{\partial t} \right)_{\text{coll}} , \tag{1.11}$$

where $\left(\frac{\partial f_s}{\partial t}\right)_{\text{coll}}$ is the collision term, which accounts for interactions between individual particles. In the case of plasmas, $\frac{d\mathbf{p}}{dt}$ is given by the Lorentz force. Due to the condition in Equation (1.4), short-range collisions between particles can be neglected, and the collision term vanishes. When the collision term vanishes, the equation is called the Vlasov equation:

$$\partial_t f_s + \mathbf{v} \cdot \nabla f_s + q_s(\mathbf{E} + \boldsymbol{\beta} \times \mathbf{B}) \cdot \nabla_{\mathbf{p}} f_s = 0 \, , \tag{1.12}$$

where we have defined the vector $\boldsymbol{\beta} = \frac{\mathbf{v}}{c}$. Maxwell's equations together with the Vlasov equation form the Vlasov-Maxwell system that describes the dynamics of plasma.

1.4 Cold-fluid limit

Solving the Vlasov-Maxwell system is challenging due to the non-linear nature of plasma dynamics. Fortunately, in the case of beam instabilities, taking the cold-fluid limit of the system allows us to investigate these instabilities analytically. To do so, we calculate the moments of the Vlasov equation.

In general, the moment $M(\mathbf{p})$ of the Vlasov equation (see Equation 1.12) is:

$$\int M \left(\partial_t f_s + \mathbf{v} \cdot \nabla f_n + q_s(\mathbf{E} + \boldsymbol{\beta} \times \mathbf{B}) \cdot \nabla_{\mathbf{p}} f_s \right) d\mathbf{p} = 0 \, , \tag{1.13}$$

where the integration is carried out over momentum space. In the first two terms, the derivatives can be trivially taken outside the integrand, but the third term requires some manipulation.

For brevity, we define $a_s^i = q_s E^i + c^{-1} q_s \epsilon^i_{lk} v^l B^k$, where ϵ^i_{lk} is the totally antisymmetric Levi-Civita symbol and Einstein summation convention is followed. Using the identity:

$$\partial_{p_i} v^l = \frac{\delta^l_i}{m\gamma} - \frac{p^l p_i}{m^3 c^2 \gamma^3} \, , \tag{1.14}$$

where δ^i_j is the Kronecker delta, we find that $\partial_{p_i} a^i_s = 0$. After integration by parts and noting that physical distributions vanishes as $\mathbf{p}$ approaches infinity, we can write the third term as:

$$\int M \mathbf{a_s} \cdot \nabla_{\mathbf{p}} f_s d\mathbf{p} = - \int a^i \partial_{p_i} M f d\mathbf{p} \, . \tag{1.15}$$

We obtain:

$$\partial_t \int M f_s d\mathbf{p} + \partial_i \int M v^i f_s d\mathbf{p} = \int a^i \partial_{p_i} M f_s d\mathbf{p} \, . \tag{1.16}$$

A cold-fluid is characterized by a distribution function $f(\mathbf{x},\mathbf{p},t) = n_s(\mathbf{x},t)\delta(\mathbf{p} - \mathbf{P}_s(\mathbf{x},t))$, where $\mathbf{P}_s(\mathbf{x},t) = m_s\gamma_s\mathbf{V}_s(\mathbf{x},t)$ is calculated from the bulk velocity $\mathbf{V}_s(\mathbf{x},t)$ of the fluid, and $n_s(\mathbf{x},t)$ is its number density (Bret et al., 2010, 2004). Inserting this and $M = 1$ into Equation (1.16), we obtain the continuity equation (Hazeltine, 2018):

$$\partial_t n_s + \nabla \cdot (\mathbf{V}_s n_s) = 0\,. \tag{1.17}$$

Using the continuity equation for $M = \mathbf{p}$, we obtain:

$$\partial_t \mathbf{P}_s + (\mathbf{V}_s \cdot \nabla)\mathbf{P}_s = q_s(\mathbf{E} + c^{-1}\mathbf{V}_s \times \mathbf{B})\,. \tag{1.18}$$

1.5 Linearization of the cold-fluid limit

Even in the cold-fluid limit, solving for the general evolution of the system is challenging, so we resort to solving it perturbatively to a certain order. In this thesis, linear-order solutions to Equations (1.17) and (1.18) are sufficient. Furthermore, we assume that the initial electromagnetic fields vanish*.

Denoting equilibrium solutions with the subscript 0 and small perturbations with the subscript 1, we can write (omitting particle species labels for brevity):

$$n = n_0 + n_1\,, \tag{1.19}$$

$$\mathbf{V} = \mathbf{V}_0 + \mathbf{V}_1\,, \tag{1.20}$$

$$\mathbf{E} = \mathbf{E}_1\,, \tag{1.21}$$

$$\mathbf{B} = \mathbf{B}_1\,, \tag{1.22}$$

$$\mathbf{P} = m\gamma[\mathbf{V}_0 + \mathbf{V}_1](\mathbf{V}_0 + \mathbf{V}_1)\,, \tag{1.23}$$

where $\gamma[\mathbf{V}]$ denotes the Lorentz factor corresponding to the coordinate velocity $\mathbf{V}$. We use the Fourier representation for the perturbations, i.e., $g_1 = \int \hat{g}_1 \exp(i(\mathbf{k}\cdot\mathbf{x} - \omega t))d\omega d^3k$. Keeping only the linear-order terms, the expression for the momentum simplifies to

$$\mathbf{P} = m\gamma\left(\mathbf{V}_0 + \frac{\gamma^2}{c^2}(\mathbf{V}_0 \cdot \hat{\mathbf{V}}_1)\mathbf{V}_0\right)\,, \tag{1.24}$$

where we defined $\gamma \equiv \gamma[\mathbf{V}_0]$.

Inserting these into Equations (1.17) and (1.18), we obtain the linearized continuity and the linearized momentum transport equations (Bret et al., 2004; Bret, 2012):

$$\hat{n}_1 = \frac{n_0\mathbf{k}\cdot\hat{\mathbf{V}}_1}{\omega - \mathbf{k}\cdot\mathbf{V}_0}\,, \tag{1.25}$$

$$im\gamma(\mathbf{k}\cdot\mathbf{V}_0 - \omega)\left(\hat{\mathbf{V}}_1 + \frac{\gamma^2}{c^2}(\mathbf{V}_0 \cdot \hat{\mathbf{V}}_1)\mathbf{V}_0\right) = q\left(\hat{\mathbf{E}}_1 + \frac{1}{c}\hat{\mathbf{V}}_1 \times \mathbf{B}_0 + \frac{1}{c}\mathbf{V}_0 \times \hat{\mathbf{B}}_1\right)\,. \tag{1.26}$$

*The analysis could be done with non-zero initial magnetic field for which $\mathbf{V}_0 \times \mathbf{B}_0 = 0$. However, this is unnecessary in context of this thesis.

Additionally, we utilize linearity of Maxwell's equations (see Equation 1.10) and combine them to obtain the linearized wave equation:

$$\mathbf{k} \times (\mathbf{k} \times \hat{\mathbf{E}}_1) + \frac{\omega^2}{c^2}\left(\hat{\mathbf{E}}_1 + \frac{4\pi i}{\omega}\hat{\mathbf{J}}_1\right) = 0\,, \tag{1.27}$$

where $\hat{\mathbf{J}}_1$ in the cold-fluid limit is:

$$\hat{\mathbf{J}}_1 = \sum_s q_s(n_{s0}\hat{\mathbf{V}}_{s1} + \hat{n}_{s1}\mathbf{V}_{s0})\,. \tag{1.28}$$

1.6 Finite-difference

The PIC scheme requires solving Maxwell's equations numerically on a discretized grid. The most common way to discretize the derivatives is the centered finite-difference method, which leads to finite-difference time-domain method which is discussed in the next section. Let the function $f(x)$ be discretized into uniform grid of n points with a separation h, where $n \in \mathbb{N}$ and h is the constant grid spacing. For brevity, we define the notation $f_n = f(nh)$ and $f'_n = \frac{df}{dx}\Big|_{x=nh}$.

The second-order accurate centered finite-difference discretizes the derivative as:

$$f'_n \approx \frac{f_{n+1} - f_{n-1}}{2h}\,. \tag{1.29}$$

The second-order accuracy of the algorithm is observed by Taylor expanding the right-hand side of Equation (1.29):

$$\frac{f_{n+1} - f_{n-1}}{2h} = \frac{1}{2h}\left(\sum_{k=0}^{\infty}\frac{h^k}{!k}f_n^{(k)} - \sum_{k=0}^{\infty}\frac{(-h)^k}{!k}f_n^{(k)}\right) = f'_n + O(h^2)\,. \tag{1.30}$$

A centered second-order derivative can be obtained by applying the centered first-order derivative twice:

$$f''_n \approx \frac{f'_{n+1} - f'_{n-1}}{2h} = \frac{f_{n+2} - 2f_n + f_{n-2}}{4h^2}\,. \tag{1.31}$$

Similarly to the first derivative, we can see the second-order accuracy from the Taylor expansion:

$$\frac{f_{n+2} - 2f_n + f_{n-2}}{4h^2} = f''_n + O(h^2)\,. \tag{1.32}$$

We note that the central finite-difference second-order derivative in Equation (1.31) does not require the function values f_{n+1} and f_{n-1}. Thus, it is usually defined on a finer grid with grid points at $\frac{nh}{2}$, where $n \in \mathbb{N}$. This gives us the standard centered finite-difference second-order derivative:

$$f''_n \approx \frac{f_{n+1} - 2f_n + f_{n-1}}{h^2}\,, \tag{1.33}$$

which is still second-order accurate because the change only modifies the constant coefficients in Equation (1.32).

There exist higher-order accurate schemes for calculating the centered finite-difference, but they involve a trade-off between improved numerical accuracy and increased computational cost.

1.6.1 Numerical dispersion relation

The stability of the centered finite-difference scheme for the wave equation determines its accuracy and stability properties. The one-dimensional wave equation for function $\psi(t, x)$ is given by:

$$\frac{\partial^2 \psi}{\partial t^2} = c^2 \frac{\partial^2 \psi}{\partial x^2}\,. \tag{1.34}$$

Inserting the plane wave solution $\psi = e^{i(kx-\omega t)}$ to Equation (1.34) gives the physical dispersion relation:

$$\omega = \pm ck\,. \tag{1.35}$$

We discretize the function $\psi(x, t)$ on a uniform grid $(n\Delta x, m\Delta t)$, where $n, m \in \mathbb{N}$, and denote $\psi_n^m = \psi(n\Delta x, m\Delta t)$. Applying the centered finite-difference for second derivatives (see Equation 1.33) to both the temporal and spatial derivatives in Equation (1.34), and solving for ψ_n^{m+1}, we obtain:

$$\psi_n^{m+1} = \left(c\frac{\Delta t}{\Delta x}\right)^2 \left(\psi_{n+1}^m - 2\psi_n^m + \psi_{n-1}^m\right) + 2\psi_n^m - \psi_n^{m-1}\,. \tag{1.36}$$

Inserting the discretized plane wave $\psi_n^m = e^{i(kn\Delta x - \omega m\Delta t)}$ into Equation (1.36) results in a numerical dispersion relation that is distinctly different from the physical dispersion relation (see Equation 1.35):

$$\frac{\omega}{c}\Delta x = \pm\frac{1}{\hat{c}}\cos^{-1}\left(\hat{c}^2(\cos(k\Delta x) - 1) + 1\right)\,, \tag{1.37}$$

where $\hat{c} \equiv c\frac{\Delta t}{\Delta x}$ is the Courant-Friedrichs-Lewy number, or the Courant number for short (Verboncoeur, 2005). The solutions are periodic in k with a period of $\frac{2\pi}{\Delta x}$, which means that the finite-difference wave equation in Equation (1.36) cannot resolve waves beyond a certain wavenumber.

The numerical dispersion relation in Equation (1.37) reduces to the physical dispersion relation from Equation (1.35) when $\hat{c} = 1$ or when both Δt and Δx approach zero, but not when either one alone approaches zero. These limits are summarized in Table 1.1.

The Courant number $\hat{c}$ is crucial for the stability of solutions to the finite-difference wave equation in Equation (1.36). For $\hat{c} > 1$, the numerical dispersion relation in Equation (1.37) shows that ω has a positive imaginary part for some values of k, which results

	Equation (1.37)
$\hat{c} = 1$	$\omega = \pm ck$
$\Delta t \to 0$	$\omega \Delta x = \pm c\sqrt{2(1 - \cos(k\Delta x))}$
$\Delta x \to 0$	$\cos(\omega \Delta t) = -c^2 k^2 \frac{(\Delta t)^2}{2} + 1$
$\Delta t \to 0$ and $\Delta x \to 0$	$\omega = \pm ck$

Table 1.1: Summary of different limits for the numerical dispersion relation from Equation (1.37) (Verboncoeur, 2005).

in an exponentially growing mode for $\psi(t, x)$. The condition for stability, $\hat{c} = c\frac{\Delta t}{\Delta x} \le 1$, is called the Courant-Friedrichs-Lewy condition. In multiple dimensions, the condition is given by (Verboncoeur, 2005):

$$\Delta t \le \frac{1}{c}\left(\sum_i \frac{1}{(\Delta x_i)^2}\right)^{-1/2} . \tag{1.38}$$

Figure 1.1 shows one period of the numerical dispersion relation given in Equation (1.37).

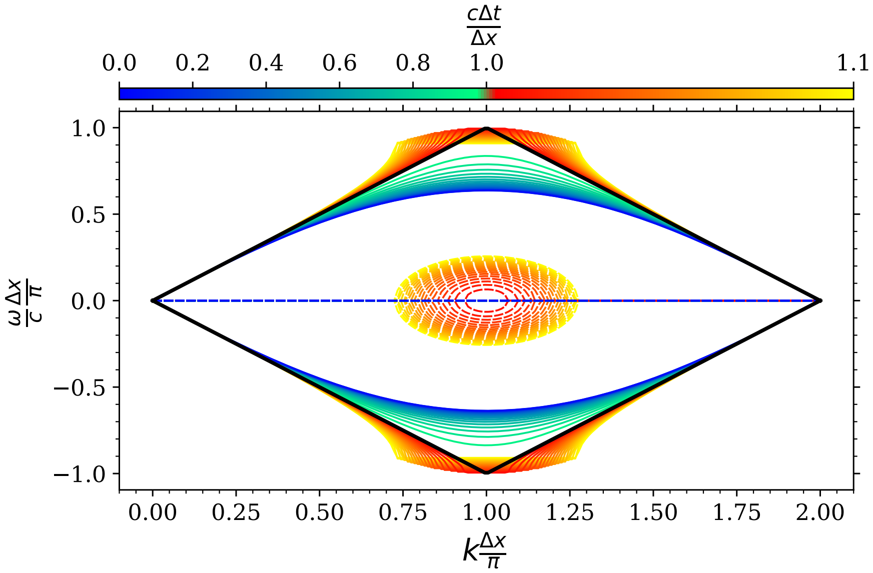

Figure 1.1: One period of the numerical dispersion relation given in Equation (1.37) for different values of the Courant number $\hat{c} = \frac{c\Delta t}{\Delta x}$ ranging from 0 to 1.1. The real parts of $\frac{\omega}{c}\frac{\Delta x}{\pi}$ are shown as continuous lines, while the imaginary parts are shown as dashed curves. Courant numbers that satisfy the Courant-Friedrichs-Lewy condition have colors ranging from blue to green, whereas those that do not satisfy the condition have colors ranging from red to yellow. The black curve represents the transition point at $\hat{c} = 1$. For every plane wave solutions, there exists an another plane wave solutions propagating in the opposite direction. This manifests as symmetry along $\omega = 0$ in the numerical dispersion relation. The symmetry along $k\frac{\Delta x}{\pi} = 1$ is a numerical aliasing artifact of the discrete grid, which cannot represent wavelengths $\lambda < 2\Delta x$.

2. Particle-in-Cell

To include kinetic effects in plasma simulations, the method must be able to describe the evolution of the full distribution function. With PIC, this is achieved by sampling the distribution function with macroparticles and evolving them according to Maxwell's equations. The electromagnetic fields are represented on a grid through which the particles move, giving rise to the name particle-in-cell.

The PIC scheme was formalised in 1970s but has its roots in work by Buneman (1959) and Dawson (1962). Later, two classic PIC text books were published: one by Hockney & Eastwood (1981) and another by Langdon (1991).

Basic PIC consists of many smaller tasks, for which numerous algorithms exist with varying trade-offs. At high-evel a single PIC time step contains four parts: field propagation, field interpolation, particle update/push, and current deposition. In this chapter, we describe these tasks in more detail and discuss the set of algorithms implemented in the GPU port of Runko.

2.1 Physical model

In relativistic PIC, the equations of motion for a particle with spatial component of a four-velocity $\mathbf{u}(\mathbf{v}) = \gamma\mathbf{v}$, coordinate velocity $\mathbf{v}$, mass m, and charge q are (Verboncoeur, 2005; Vay & Godfrey, 2014):

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{u}) = \frac{\mathbf{u}}{\gamma}\,, \tag{2.1}$$

$$m\frac{d\mathbf{u}}{dt} = \mathbf{F} = q\left(\mathbf{E} + \frac{\mathbf{v}(\mathbf{u})}{c} \times \mathbf{B}\right)\,, \tag{2.2}$$

where

$$\gamma = \sqrt{\frac{1}{1 - \frac{\mathbf{v}^2}{c^2}}} = \sqrt{1 + \frac{\mathbf{u}^2}{c^2}} \tag{2.3}$$

is the Lorentz factor, and the force $\mathbf{F}$ is given by the Lorentz force.

Electric and magnetic fields, $\mathbf{E}$ and $\mathbf{B}$, are evolved according to Maxwell's equations (see Equation 1.10):

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}\,, \tag{2.4}$$

$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{J}\,, \tag{2.5}$$

$$\nabla \cdot \mathbf{E} = 4\pi\rho\,, \tag{2.6}$$

$$\nabla \cdot \mathbf{B} = 0\,. \tag{2.7}$$

When charge conserving current deposition scheme is used and the initial conditions satisfy Equations (2.6) and (2.7), then they are automatically satisfied throughout the rest of the simulation as well (Eastwood, 1991).

2.2 Field propagator

The purpose of the field propagator is to evolve the electric field **E** and magnetic field **B** according to Equations (2.4) and (2.5). This is typically achieved using finite-difference time-domain (FDTD) discretizations on a Yee mesh (Yee, 1966). In this approach, the electromagnetic fields are staggered in both time and space to compute derivatives using centered finite-differences, as described in Section 1.6. In this Section, we derive the second-order accurate FDTD update equations for the electric and magnetic fields.

2.2.1 Yee mesh

In a Yee mesh, the vertices are located at:

$$\mathbf{x}_{m,n,l} = m\Delta x_0 \hat{\mathbf{e}}_0 + n\Delta x_1 \hat{\mathbf{e}}_1 + l\Delta x_2 \hat{\mathbf{e}}_2\,, \tag{2.8}$$

where Δx_i is the grid spacing in the direction of $\hat{\mathbf{e}}_i$, $\hat{\mathbf{e}}_i$ is the orthogonal unit vector, and $m, n, l \in \mathbb{N}$. Electric field components E_i and current density componentes J_i are stored at:

$$\mathbf{x}^{E_i}_{m,n,l} = \mathbf{x}_{m,n,l} + \frac{1}{2}\Delta x_i \hat{\mathbf{e}}_i = \begin{cases} \mathbf{x}_{m+\frac{1}{2},n,l} & i = 0 \\ \mathbf{x}_{m,n+\frac{1}{2},l} & i = 1 \\ \mathbf{x}_{m,n,l+\frac{1}{2}} & i = 2 \end{cases} \tag{2.9}$$

and magnetic field components B_i are stored at:

$$\mathbf{x}^{B_i}_{m,n,l} = \mathbf{x}_{m,n,l} + \frac{1}{2}\sum_{r\in\{1,2,3\}/i} \Delta x_r \hat{\mathbf{e}}_r = \begin{cases} \mathbf{x}_{m,n+\frac{1}{2},l+\frac{1}{2}} & i = 0 \\ \mathbf{x}_{m+\frac{1}{2},n,l+\frac{1}{2}} & i = 1 \\ \mathbf{x}_{m+\frac{1}{2},n+\frac{1}{2},l} & i = 2 \end{cases}\,. \tag{2.10}$$

See Figure 2.1 for a schematic illustration of the Yee mesh.

The most common way to solve Equations (2.4) and (2.5) numerically is the explicit leapfrog algorithm (Verboncoeur, 2005; Langdon, 1991). In the leapfrog algorithm, in addition to spatial staggering, the fields are also staggered in time. Electric fields and current densities are defined at times $s\Delta t$, while magnetic fields are defined at times $\left(s+\frac{1}{2}\right)\Delta t$, where $s \in \mathbb{N}$ and Δt is the length of one time step. This time staggering creates a challenge for particle pushing, which requires both fields at the same point in time. To address this, the magnetic field is usually updated in two half-steps of $\frac{1}{2}\Delta t$, and particle pushing is performed between these updates.

Because the components of the electromagnetic fields are stored at different locations, we can deduce the component solely from its position. Thus, we adopt the following shorthand notation for the components of $\mathbf{E}(t,\mathbf{x})$ and $\mathbf{B}(t,\mathbf{x})$:

$$\xi^T_{N,M,L} \equiv \xi\left(T\Delta t, \mathbf{x}^{\xi}_{N,M,L}\right) , \tag{2.11}$$

where ξ denotes either E or B. For example $E^s_{n,m,l+\frac{1}{2}} = E_2\left(s\Delta t, \mathbf{x}^{E_2}_{n,m,l}\right)$ and $B^{s+\frac{1}{2}}_{n+\frac{1}{2},m,l+\frac{1}{2}} = B_1\left(\left(s+\frac{1}{2}\right)\Delta t, \mathbf{x}^{B_1}_{n,m,l}\right)$. With this notation, we can generally write the second-order accurate centered finite-difference temporal and spatial derivatives as given by Equation (1.29). For the temporal derivative, $h = \frac{1}{2}\Delta t$, and for the spatial derivative, in the $\hat{\mathbf{e}}_i$ direction, $h = \frac{1}{2}\Delta x_i$. Thus, we obtain:

$$\Delta t \partial_t \xi^T_{N,M,L} \approx \xi^{T+\frac{1}{2}}_{N,M,L} - \xi^{T-\frac{1}{2}}_{N,M,L} , \tag{2.12}$$

$$\begin{bmatrix} \Delta x_0 \partial_{x_0} \\ \Delta x_1 \partial_{x_1} \\ \Delta x_2 \partial_{x_2} \end{bmatrix} \xi^T_{N,M,L} \approx \begin{bmatrix} \xi^T_{N+\frac{1}{2},M,L} - \xi^T_{N-\frac{1}{2},M,L} \\ \xi^T_{N,M+\frac{1}{2},L} - \xi^T_{N,M-\frac{1}{2},L} \\ \xi^T_{N,M,L+\frac{1}{2}} - \xi^T_{N,M,L-\frac{1}{2}} \end{bmatrix} . \tag{2.13}$$

Note that in the next subsection we will evaluate derivatives $\partial_\mu \xi$ at specific locations where, on the Yee mesh, the field ξ may not be directly defined. However, we will use centered finite-differences for these derivatives, and by construction the Yee mesh provides exactly the components we need.

2.2.2 Finite-difference time domain

We now apply the finite-difference derivatives from Equation (2.12) and (2.13) to the dynamical Maxwell Equations (2.4) and (2.5) in order to derive rules for updating the fields in time. Ignoring the current density for the moment*, both equations can be written

*We can always add the current to the electric field, as they are stored at the same location and time in the Yee mesh.

in a form that, at time $T\Delta t$ and position $\mathbf{x}_{N,M,L}$, becomes:

$$\partial_t \begin{bmatrix} \alpha^T_{N,M,L} \\ \beta^T_{N,M,L} \\ \gamma^T_{N,M,L} \end{bmatrix} = \pm c\nabla \times \begin{bmatrix} \mu^T_{N,M,L} \\ \nu^T_{N,M,L} \\ \rho^T_{N,M,L} \end{bmatrix} = \pm c \begin{bmatrix} \partial_{x_1}\rho^T_{N,M,L} - \partial_{x_2}\nu^T_{N,M,L} \\ \partial_{x_2}\mu^T_{N,M,L} - \partial_{x_0}\rho^T_{N,M,L} \\ \partial_{x_0}\nu^T_{N,M,L} - \partial_{x_1}\mu^T_{N,M,L} \end{bmatrix}, \tag{2.14}$$

where the symbols α, β, γ, μ, ν, and ρ are placeholders for electric or magnetic field components. Now we apply the second-order accurate centered finite-difference derivatives from Equations (2.12) and (2.13) on the Yee mesh to obtain the update equations:

$$\begin{aligned} \begin{bmatrix} \alpha^{T+\frac{1}{2}}_{N,M,L} \\ \beta^{T+\frac{1}{2}}_{N,M,L} \\ \gamma^{T+\frac{1}{2}}_{N,M,L} \end{bmatrix} \approx & \begin{bmatrix} \alpha^{T-\frac{1}{2}}_{N,M,L} \\ \beta^{T-\frac{1}{2}}_{N,M,L} \\ \gamma^{T-\frac{1}{2}}_{N,M,L} \end{bmatrix} \\ & \pm \begin{bmatrix} \hat{c}_1\left(\rho^T_{N,M+\frac{1}{2},L} - \rho^T_{N,M-\frac{1}{2},L}\right) - \hat{c}_2\left(\nu^T_{N,M,L+\frac{1}{2}} - \nu^T_{N,M,L-\frac{1}{2}}\right) \\ \hat{c}_2\left(\mu^T_{N,M,L+\frac{1}{2}} - \mu^T_{N,M,L-\frac{1}{2}}\right) - \hat{c}_0\left(\rho^T_{N+\frac{1}{2},M,L} - \rho^T_{N-\frac{1}{2},M,L}\right) \\ \hat{c}_0\left(\nu^T_{N+\frac{1}{2},M,L} - \nu^T_{N-\frac{1}{2},M,L}\right) - \hat{c}_1\left(\mu^T_{N,M+\frac{1}{2},L} - \mu^T_{N,M-\frac{1}{2},L}\right) \end{bmatrix}, \end{aligned} \tag{2.15}$$

where we have used the one-dimensional Courant numbers for different directions, defined as $\hat{c}_i \equiv \frac{c\Delta t}{\Delta x_i}$.

Now we can extract the actual update equations for $\mathbf{E}$ and $\mathbf{B}$ from Equation (2.15) by evaluating their components at specific locations in time and space. First, we substitute:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \mathbf{B}, \quad \begin{bmatrix} \mu \\ \nu \\ \rho \end{bmatrix} = \mathbf{E}, \quad (T,N,M,L) = \begin{cases} (s,n,m+\frac{1}{2},l+\frac{1}{2}) & \text{on row 0} \\ (s,n+\frac{1}{2},m,l+\frac{1}{2}) & \text{on row 1} \\ (s,n+\frac{1}{2},m+\frac{1}{2},l) & \text{on row 2} \end{cases} \tag{2.16}$$

and obtain:

$$\begin{aligned} \begin{bmatrix} B^{s+\frac{1}{2}}_{n,m+\frac{1}{2},l+\frac{1}{2}} \\ B^{s+\frac{1}{2}}_{n+\frac{1}{2},m,l+\frac{1}{2}} \\ B^{s+\frac{1}{2}}_{n+\frac{1}{2},m+\frac{1}{2},l} \end{bmatrix} \approx & \begin{bmatrix} B^{s-\frac{1}{2}}_{n,m+\frac{1}{2},l+\frac{1}{2}} \\ B^{s-\frac{1}{2}}_{n+\frac{1}{2},m,l+\frac{1}{2}} \\ B^{s-\frac{1}{2}}_{n+\frac{1}{2},m+\frac{1}{2},l} \end{bmatrix} \\ & - \begin{bmatrix} \hat{c}_1\left(E^s_{n,m+1,l+\frac{1}{2}} - E^s_{n,m,l+\frac{1}{2}}\right) - \hat{c}_2\left(E^s_{n,m+\frac{1}{2},l+1} - E^s_{n,m+\frac{1}{2},l}\right) \\ \hat{c}_2\left(E^s_{n+\frac{1}{2},m,l+1} - E^s_{n+\frac{1}{2},m,l}\right) - \hat{c}_0\left(E^s_{n+1,m,l+\frac{1}{2}} - E^s_{n,m,l+\frac{1}{2}}\right) \\ \hat{c}_0\left(E^s_{n+1,m+\frac{1}{2},l} - E^s_{n,m+\frac{1}{2},l}\right) - \hat{c}_1\left(E^s_{n+\frac{1}{2},m+1,l} - E^s_{n+\frac{1}{2},m,l}\right) \end{bmatrix}, \end{aligned} \tag{2.17}$$

and second, we substitute:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \mathbf{E} \quad \begin{bmatrix} \mu \\ \nu \\ \rho \end{bmatrix} = \mathbf{B} \quad (T,N,M,L) = \begin{cases} (s+\frac{1}{2},n+\frac{1}{2},m,l) & \text{on row 0} \\ (s+\frac{1}{2},n,m+\frac{1}{2},l) & \text{on row 1} \\ (s+\frac{1}{2},n,m,l+\frac{1}{2}) & \text{on row 2} \end{cases} \tag{2.18}$$

and obtain (with the current density added back in):

$$
\begin{aligned}
\begin{bmatrix} E^{s+1}_{n+\frac{1}{2},m,l} \\ E^{s+1}_{n,m+\frac{1}{2},l} \\ E^{s+1}_{n,m,l+\frac{1}{2}} \end{bmatrix} \approx \begin{bmatrix} E^{s}_{n+\frac{1}{2},m,l} \\ E^{s}_{n,m+\frac{1}{2},l} \\ E^{s}_{n,m,l+\frac{1}{2}} \end{bmatrix} + 4\pi\Delta t \begin{bmatrix} J^{s}_{n+\frac{1}{2},m,l} \\ J^{s}_{n,m+\frac{1}{2},l} \\ J^{s}_{n,m,l+\frac{1}{2}} \end{bmatrix} \\
+ \begin{bmatrix} \hat{c}_1 \left(B^{s+\frac{1}{2}}_{n+\frac{1}{2},m+\frac{1}{2},l} - B^{s+\frac{1}{2}}_{n+\frac{1}{2},m-\frac{1}{2},l} \right) - \hat{c}_2 \left(B^{s+\frac{1}{2}}_{n+\frac{1}{2},m,l+\frac{1}{2}} - B^{s+\frac{1}{2}}_{n+\frac{1}{2},m,l-\frac{1}{2}} \right) \\ \hat{c}_2 \left(B^{s+\frac{1}{2}}_{n,m+\frac{1}{2},l+\frac{1}{2}} - B^{s+\frac{1}{2}}_{n,m+\frac{1}{2},l-\frac{1}{2}} \right) - \hat{c}_0 \left(B^{s+\frac{1}{2}}_{n+\frac{1}{2},m+\frac{1}{2},l} - B^{s+\frac{1}{2}}_{n-\frac{1}{2},m+\frac{1}{2},l} \right) \\ \hat{c}_0 \left(B^{s+\frac{1}{2}}_{n+\frac{1}{2},m,l+\frac{1}{2}} - B^{s+\frac{1}{2}}_{n-\frac{1}{2},m,l+\frac{1}{2}} \right) - \hat{c}_1 \left(B^{s+\frac{1}{2}}_{n,m+\frac{1}{2},l+\frac{1}{2}} - B^{s+\frac{1}{2}}_{n,m-\frac{1}{2},l+\frac{1}{2}} \right) \end{bmatrix} .
\end{aligned}
\tag{2.19}
$$

As noted at the end of Subsection 2.2.1 on the Yee mesh, the update Equations (2.17) and (2.19) do not require any values that are not already present on the Yee mesh.

In the Runko Git repository, implementation of the second-order accurate FDTD scheme is located at `core/emf/yee_lattice_FDTD2.c++`.

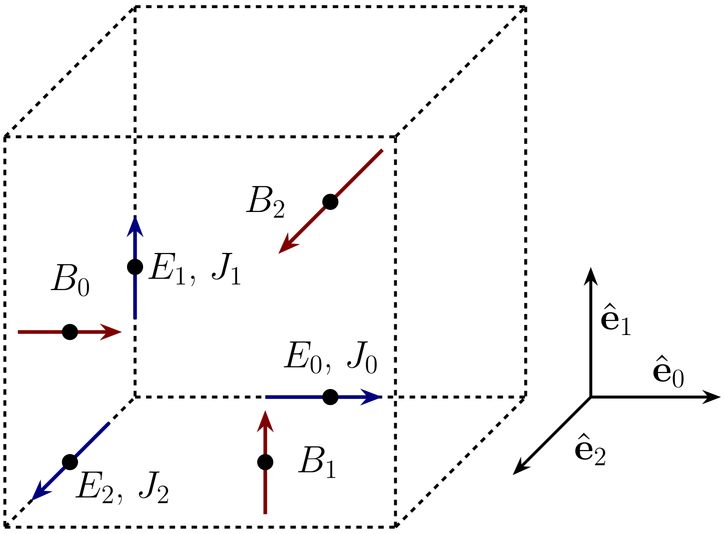

Figure 2.1: Fields on the Yee mesh. Component locations are shown with black circles. Electric field **E** and current density **J** components are shown with blue arrows and magnetic field **B** components are shown with red arrows. Black dots shows the point at which each component is defined.

2.3 Field interpolator

Before a particle can be pushed, the electromagnetic fields on the discrete Yee mesh must be interpolated to the continuous particle positions. Various interpolation schemes exists, offering different levels of accuracy and computational cost. Currently the GPU port of Runko uses linear interpolation because it is fast and simple to implement. In the Runko Git repository, implmementation is located at `core/emf/yee_lattice_interpolate_linear_1st.c++`. More interpolators can be implemented later when desired.

The interpolation is performed for the fields **E** and **B**, which are located on the Yee mesh according to Equations (2.9) and (2.10). While interpolation could be done directly from those locations to the particle locations, it is common practice to first compute **E** and **B** at the Yee mesh vertex locations (see Equation 2.8), for example via a simple spatial averaging. This approach simplifies the definition and implementation of interpolators and helps reduce short-wavelength noise (Langdon, 1991).

We can now express the interpolation of a generic function f, defined on the grid vertices (see Equation 2.8), to an arbitrary point $\mathbf{x}$:

$$f(\mathbf{x}) = \Delta V \sum_{n,m,l} f_{n,m,l} S' \left(\mathbf{x}_{n,m,l}, \mathbf{x}\right) \,, \tag{2.20}$$

where $\Delta V = \prod_i \Delta x_i$, $f_{n,m,l} \equiv f(\mathbf{x}_{n,m,l})$, and S' is the weighting function, normalized such that at every position*:

$$\Delta V \sum_{n,m,l} S' \left(\mathbf{x}_{n,m,l}, \mathbf{x}\right) = 1 \,. \tag{2.22}$$

The choice of the shape function S' determines the interpolation scheme. Typically, the interpolation is performed independently in each spatial direction. In such cases:

$$S'(\mathbf{X}, \mathbf{x}) = S(X_0 - x_0) S(X_1 - x_1) S(X_2 - x_2) \,. \tag{2.23}$$

B-splines are commonly used as the weighing functions $S(x)$. Starting from the zeroth-order B-spline, one can construct B-splines with arbitrarily high order, although typically only the lowest-order splines are used. The first four are shown in Figure 2.2. Choosing the zeroth order B-spline is equivalent to using the value from the nearest grid point, while choosing the first-order B-spline corresponds to linear interpolation.

2.4 Particle pusher

The positions and velocities of particles are updated according to Equations (2.1) and (2.2). This procedure is called particle pushing, and the numerical methods used for it are referred to as particle pushers. Similar to the field updates, particle pushers

*This normalization follows the convention used in Langdon (1991), ensuring that the grid contains the same total charge as the particles:

$$\int_V \rho dV \approx \Delta V \sum_{n,m,l} \rho_{n,m,l} = \Delta V \sum_{n,m,l} \sum_j q_j S'(\mathbf{x}_{n,m,l}, \mathbf{x}_j) = \sum_j q_j \,, \tag{2.21}$$

where the sum over j runs over all particles, and $\mathbf{x}_j$ denotes the particle positions.

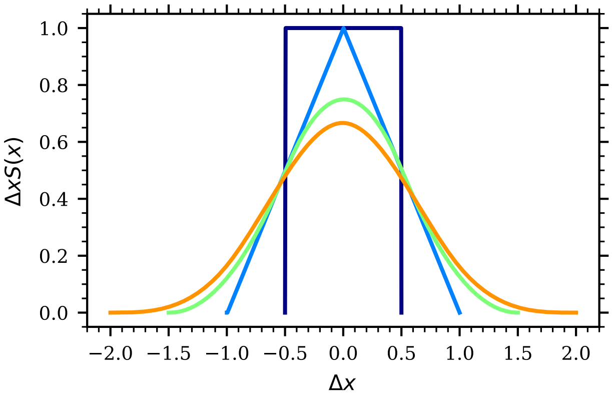

Figure 2.2: Zeroth- (violet curve), first- (blue), second- (green), and third-order (orange) B-spline weighing functions S. Higher-order B-splines produce smoother interpolation.

often employ the leapfrog scheme, so particle positions are stored at times $s\Delta t$ and velocities at times $\left(s+\frac{1}{2}\right)\Delta t$. Applying the second-order accurate centered finite-difference from Equation (1.29) to Equations (2.2) and (2.1) with $h = \frac{1}{2}\Delta t$, we can solve for the update equations:

$$\mathbf{x}^{s+1} = \mathbf{x}^{s} + \Delta t\, \mathbf{v}(\mathbf{u}^{s+\frac{1}{2}})\,, \tag{2.24}$$

$$\mathbf{u}^{s+\frac{1}{2}} = \mathbf{u}^{s-\frac{1}{2}} + \Delta t \frac{q}{m} \left(\mathbf{E}^{s} + \bar{\mathbf{v}}^{s} \times \mathbf{B}^{s}\right)\,, \tag{2.25}$$

where the superscripts denote the time step, similar to Equation (2.11), and $\bar{\mathbf{v}}^s$ represents some average for coordinate velocity at times $s\Delta t$. To close the system, this average must be expressed as a function of the other quantaties.

There are multiple different options for calculating $\bar{\mathbf{v}}^s$ corresponding to the different particle pushers. The common challenge is that the expressions often involve $\mathbf{u}^{s+\frac{1}{2}}$, which appears to make the update Equation (2.25) implicit (i.e., an equation that can not be solved directly but requires iterative or other more complex algorithms). However, various pushers have procedures that allow solving Equation (2.25) in steps, keeping the method explicit.

In this section, we compare properties of the relativistic Boris pusher (Boris, 1970), Vay's method (Vay, 2008), and Higuera-Cary method (Higuera & Cary, 2017). Since the GPU port of Runko uses the relativistic Boris pusher, we will examine it in more detail. In the Runko Git repository, implementation of the Boris pusher is located at `core/pic/particle_boris.c++`. From now on we will refer the relativistic Boris pusher as just the Boris pusher for short.

2.4.1 Properties

We begin by listing the choises for $\bar{\mathbf{v}}^s$. The relativistic Boris pusher uses:

$$\bar{\mathbf{v}}_b^s \equiv \frac{\mathbf{v}\left(\mathbf{u}^{s+\frac{1}{2}} - \boldsymbol{\epsilon}^s\right) + \mathbf{v}\left(\mathbf{u}^{s-\frac{1}{2}} + \boldsymbol{\epsilon}^s\right)}{2\gamma^s}\,, \tag{2.26}$$

where $\boldsymbol{\epsilon}^s = \frac{q\Delta t\mathbf{E}^s}{2m}$ is the change of coordinate velocity due to the electric field in half a time step. It is assumed that the vectors $\mathbf{u}^{s\pm\frac{1}{2}} \mp \boldsymbol{\epsilon}^s$ share the same length. Vay's method uses:

$$\bar{\mathbf{v}}_v^s \equiv \frac{\mathbf{v}^{s-\frac{1}{2}} + \mathbf{v}^{s+\frac{1}{2}}}{2} \tag{2.27}$$

and Higuera-Cary method uses:

$$\bar{\mathbf{v}}_{hc}^s \equiv \mathbf{v}\left(\frac{\mathbf{u}^{s-\frac{1}{2}} + \mathbf{u}^{s+\frac{1}{2}}}{2}\right)\,. \tag{2.28}$$

The choices for $\bar{\mathbf{v}}^s$ appear very similar; however, the methods possess distinct properties that are crucial for physical accuracy. Since the underlying physical system is Hamiltonian, the solutions lie on a symplectic manifold in phase space (Goldstein et al., 2014). Symplectic integrators are those whose solutions remain on this symplectic manifold. The Kolmogorov-Arnold-Moser theorem guarantees that their long-term evolution preserves physical accuracy (Higuera & Cary, 2017)*. Unfortunately, no explicit symplectic integrator is currently known for charged-particle motion in arbitrary electromagnetic fields. Nevertheless, integrators can be designed to be phase-space volume-preserving, which prevents the emergence of attractors and repellers in the integrated system.

The Boris pusher is volume-preserving, whereas Vay's method is not. Conversely, Vay's method computes the $\mathbf{E} \times \mathbf{B}$ drift correctly, while the Boris pusher does not. The Higuera-Cary method, on the other hand, possesses both of these properties. A common feature among all three methods is that they conserve energy in the absence of an electric field and are second-order accurate in time. From a computational perspective, the Boris pusher is the fastest, whereas Vay's and Higuera-Cary methods exhibit similar computational cost.

2.4.2 Boris pusher

We begin by substituting $\bar{\mathbf{v}}_b^s$ from Equation (2.26) into Equation (2.25) and adopting a notation in which we drop the superscript s and replace superscripts $s + \frac{1}{2}$ with f and

*Defining what a symplectic manifold is and what constitutes a physically accurate long-term evolution of a Hamiltonian system lies outside the scope of this thesis.

$s - \frac{1}{2}$ with i:

$$\mathbf{u}^f - \mathbf{u}^i = 2\boldsymbol{\epsilon} + \eta(\mathbf{u}^f + \mathbf{u}^i) \times \hat{\mathbf{B}}\,, \tag{2.29}$$

where we define $\boldsymbol{\epsilon} = \frac{q\Delta t\mathbf{E}}{2m}$, $\eta = \frac{q\Delta t B}{2mv\gamma}$ and the unit vector $\hat{\mathbf{B}} = \mathbf{B}B^{-1}$. Note that the same Lorentz factor is shared between $\mathbf{u}^f$ and $\mathbf{u}^i$ because we assume they have the same length. We want to find a way to evaluate Equation (2.29) without using implicit methods; that is, we seek to evaluate the right-hand side without explicitly requiring $\mathbf{u}^f$.

We define the vectors $\mathbf{u}_+ = \mathbf{u}^f - \boldsymbol{\epsilon}$ and $\mathbf{u}_- = \mathbf{u}^i + \boldsymbol{\epsilon}$, thus Equation (2.29) can be rewritten as:

$$\mathbf{u}_+ - \mathbf{u}_- = \eta(\mathbf{u}_+ + \mathbf{u}_-) \times \hat{\mathbf{B}}\,. \tag{2.30}$$

We note that the above Equation (2.30) contains no components along $\hat{\mathbf{B}}$. Denoting vector components in the plane perpendicular to $\hat{\mathbf{B}}$ with a prime, we can write:

$$\mathbf{u}_+ - \mathbf{u}_- = \mathbf{u}'_+ - \mathbf{u}'_- = \eta(\mathbf{u}'_+ + \mathbf{u}'_-) \times \hat{\mathbf{B}}\,. \tag{2.31}$$

We can deduce that the vectors $\mathbf{u}'_+ - \mathbf{u}'_-$ and $\mathbf{u}'_+ + \mathbf{u}'_-$ are perpendicular to each other. By taking the dot product of Equation (2.31) with $\mathbf{u}'_+ + \mathbf{u}'_-$ we find that $u'_+ = u'_-$. Let θ denote the angle between the vectors $\mathbf{u}'_+$ and $\mathbf{u}'_-$. A diagram illustrating these vectors is shown in Figure 2.3, where we have added helper vectors $\mathbf{h}$, $\mathbf{L}_0$, $\mathbf{L}_1$ and $\mathbf{u}'_- \times \mathbf{b}$, such that $\mathbf{L}_0$ and $\mathbf{L}_1$ are perpendicular to each other, $\mathbf{b} = b\hat{\mathbf{B}}$, and that $\mathbf{L}_0 + \mathbf{L}_1 = \mathbf{u}'_- + \mathbf{u}'_- \times \mathbf{b}$. Using these definitions, we note that $b = \tan\frac{\theta}{2} = \frac{h}{L_1}$ and $L_0 = L_1 b^2$.

We now attempt to express $\mathbf{u}'_+ - \mathbf{u}'_-$ in a different form than in Equation (2.31). Specifically, we can write:

$$\begin{aligned}\mathbf{u}'_+ - \mathbf{u}'_- &= 2h\hat{\mathbf{L}}_1 \times \hat{\mathbf{B}} = \frac{2}{1+b^2}(\mathbf{L}_0 + \mathbf{L}_1) \times \mathbf{b} \\ &= \frac{2}{1+b^2}(\mathbf{u}'_- + \mathbf{u}'_- \times \mathbf{b}) \times \mathbf{b}\,,\end{aligned} \tag{2.32}$$

where $\hat{\mathbf{L}}_1$ is the unit vector along $\mathbf{L}_1$. We now return to $\mathbf{u}^f$ and $\mathbf{u}^i$ to obtain the Boris pusher update equation:

$$\begin{aligned}\mathbf{u}^f &= \mathbf{u}^i + 2\boldsymbol{\epsilon} + \frac{2}{1+b^2}(\mathbf{u}'_- + \mathbf{u}'_- \times \mathbf{b}) \times \mathbf{b} \\ &= \mathbf{u}^i + \frac{q\Delta t\mathbf{E}}{m} + \frac{2}{1+b^2}\left(\mathbf{u}^i + \frac{q\Delta t\mathbf{E}}{2m} + \left(\mathbf{u}^i + \frac{q\Delta t\mathbf{E}}{2m}\right) \times \mathbf{b}\right) \times \mathbf{b}\,.\end{aligned} \tag{2.33}$$

The only remaining step is to determine the angle θ between $\mathbf{u}_+$ and $\mathbf{u}_-$ in order to calculate $\mathbf{b}$. To do this, we use physical argument: the angle should correspond to the gyrotation induced by the magnetic field. Thus, using the cyclotron frequency from Equation (1.6):

$$\theta \approx \Delta t\omega_B = \frac{qB\Delta t}{mc\gamma}\,. \tag{2.34}$$

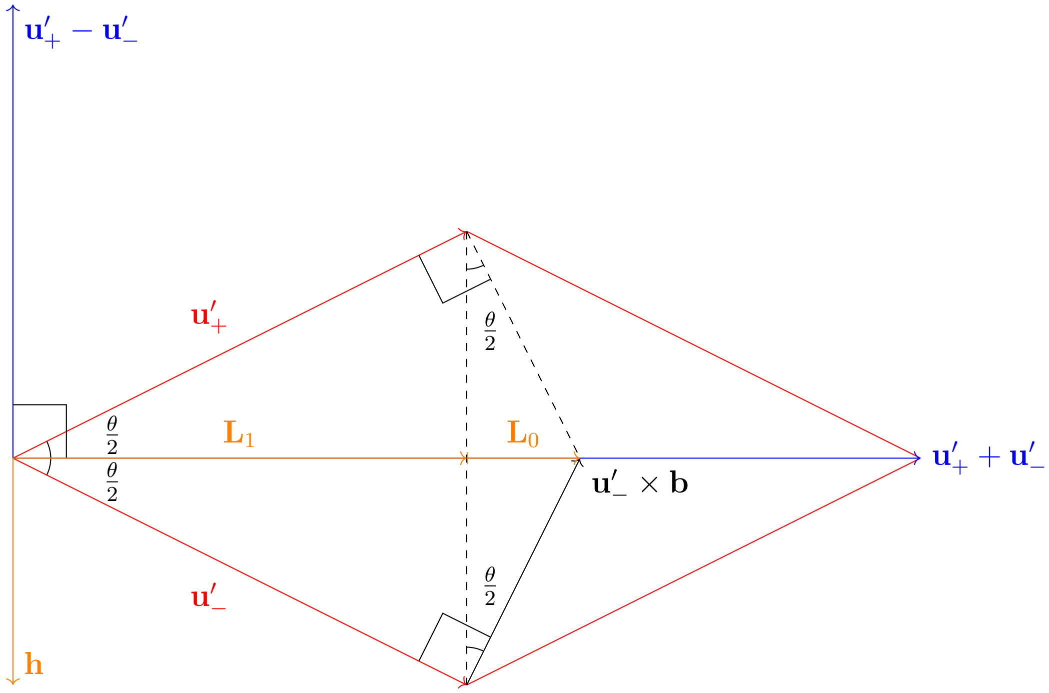

Figure 2.3: Vectors perpendicular to the magnetic field, which are used to derive the Boris pusher equations.

Under the assumption of a sufficiently small time step:

$$b = \tan\frac{\theta}{2} = \frac{\theta}{2} + O(\Delta t^3) = \frac{qB\Delta t}{2mc\gamma} + O(\Delta t^3)\,. \tag{2.35}$$

2.5 Current deposition

The moving particles generate a current according to Equation (1.9), which is deposited onto the Yee mesh. As stated in Section 2.1, if this deposition is charge-conserving and Equations (2.6) and (2.7) are satisfied initially, then they remain satisfied at all later times. For example, similar to the approach in Section 2.3, we could interpolate particle velocities to neighboring grid locations. However, this scheme is not charge conserving (Hockney & Eastwood, 1981; Langdon, 1991; Umeda et al., 2003). For a current deposition method to be charge-conserving, it must satisfy the continuity equation for charge density, which is derived by taking the time derivative of Gauss's law (see Equation 2.6):

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\mathbf{J} = 0\,. \tag{2.36}$$

Several charge-conserving current deposition schemes exist. The GPU port of Runko implements the so-called ZigZag method, as described in Umeda et al. (2003), so this section focuses on that method and the Villasenor-Buneman method (Villasenor & Buneman, 1992), since the ZigZag method can be considered an optimized version of the latter. In the Runko Git repository, implementation of the ZigZag method is located at `core/pic/particle_current_zigzag_1st.c++`.

Another notable scheme is the density decomposition method proposed by Esirkepov (2001). A general treatment of charge-conserving methods using finite-element techniques is provided by Eastwood (1991), with an extension to curvilinear coordinates in Eastwood et al. (1995). Both the Villasenor-Buneman method and Esirkepov's density decomposition can be regarded as special cases of Eastwood's scheme.

The way Villasenor-Buneman method calculates the current from a particle trajectory can only be applied if the trajectory remains within the same Yee mesh cell (see Figure 2.1). If it does not, the movement is split into multiple shorter segments, each confined to a single cell. The difference between the Villasenor-Buneman method and the ZigZag method lies in how this splitting is performed. In the Villasenor-Buneman method, the particle's motion within one time step is assumed to be straight and is split at the Yee mesh cell faces. In contrast, the ZigZag method assumes the motion occurs via a so-called relay point. The advantage of the relay point is that, regardless of particle position and velocity, the current deposition is calculated in a uniform manner, whereas the Villasenor-Buneman method requires conditional checks to determine whether the particle crosses zero, one, or two cell faces. This makes it possible to implement the ZigZag method without conditionals, which is important for good performance on parallel hardware (e.g. GPUs).

We note that the 3D ZigZag method described in the original paper is defective and not charge-conserving. For particle motion confined to a single cell, the method adds current in the same way as the Villasenor-Buneman method (equations (35-37) in (Villasenor & Buneman, 1992)), but it omits the term proportional to $\delta x_0 \delta x_1 \delta x_2$, where δx_i represents the change in the particle's position during one time step. As a result, current deposition from diagonal trajectories is not charge-conserving. This observation is consistent with the findings reported in Steiniger et al. (2023). Preliminary tests indicate that the 3D ZigZag method can be corrected by adding the missing terms, but this subject requires further studies.

2.5.1 Villasenor-Buneman method

The Villasenor-Buneman method assumes that particles contribute current to the Yee mesh as an axis-aligned box with a uniform charge distribution, equal in size to one Yee mesh cell. Therefore, the particle's effective charge density is $\rho_q = \frac{q}{\Delta x_0 \Delta x_1 \Delta x_2}$. Charge is conserved on each cell of a discrete grid that has the same size as the Yee mesh but is staggered by half a grid spacing in each direction. When a particle moves inside a single Yee mesh cell, it generates current through exactly twelve faces of this staggered grid.

These twelve faces, along with the Yee mesh cell, are shown in Figure 2.4. The currents through each face can be integrated so that the total current into a single cell during one time step equals the change in charge within that cell. Thus, the continuity equation for charge density (see Equation 2.36) is satisfied when integrated over a single cell.

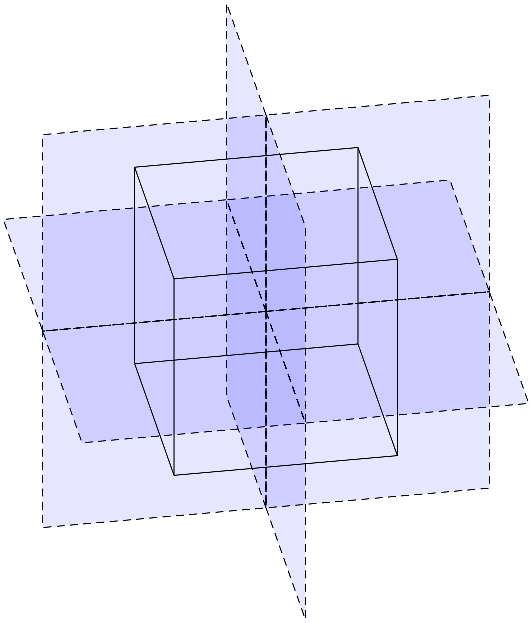

Figure 2.4: The twelve faces through which a moving particle generates current fluxes in the Villasenor-Buneman method. The Yee mesh cell in which the particle trajectory is confined is shown with solid lines.

Let the particle positions $\mathbf{x}^s$ and $\mathbf{x}^{s+1}$ lie within the same cell, which is spanned between $\mathbf{x}_{m,n,l}$ and $\mathbf{x}_{m+1,n+1,l+1}$. Define:

$$\delta\mathbf{x} \equiv \mathbf{x}^{s+1} - \mathbf{x}^s \,, \tag{2.37}$$

$$r_i^s \equiv \left(\mathbf{x}^s - \mathbf{x}_{m,n,l}\right)_i \tag{2.38}$$

and let:

$$\bar{\mathbf{r}} \equiv \frac{\mathbf{r}^{s+1} + \mathbf{r}^s}{2} \tag{2.39}$$

denote the midpoint in local coordinates inside the cell. Assuming that the particle with charge q moves at constant velocity during one time step, its trajectory can be written as

$$\mathbf{r}(t) = \bar{\mathbf{r}} + t\delta\mathbf{x} \,, \quad t \in \left[-\frac{1}{2}, \frac{1}{2}\right] \,. \tag{2.40}$$

Then, the integrals for the total currents in $\hat{\mathbf{e}}_0$ direction at $\mathbf{x}^{E_0}_{m,n+1,l+1}$ can be expressed as:

$$\begin{aligned}
J_{m+\frac{1}{2},n+1,l+1}\Delta x_1 \Delta x_2 &= \rho_q \int_{r_0^s}^{r_0^{s+1}} r_1(t) r_2(t) dr_0 \\
&= \rho_q \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\bar{r}_1 + t\delta x_1\right)\left(\bar{r}_2 + t\delta x_2\right)\delta x_0 dt \\
&= \rho_q \delta x_0 \bar{r}_1 \bar{r}_2 + \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, ,
\end{aligned} \tag{2.41}$$

where we use the notation from the Yee mesh Subsection 2.2.1 for the current density. Similarly, for the three other faces:

$$\begin{aligned}
J_{m+\frac{1}{2},n,l}\Delta x_1 \Delta x_2 &= \rho_q \int_{r_0^s}^{r_0^{s+1}} (1 - r_1(t))(1 - r_2(t)) dr_0 \\
&= \rho_q \delta x_0 (1 - \bar{r}_1)(1 - \bar{r}_2) + \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m+\frac{1}{2},n,l+1}\Delta x_1 \Delta x_2 &= \rho_q \int_{r_0^s}^{r_0^{s+1}} (1 - r_1(t)) r_2(t) dr_0 \\
&= \rho_q \delta x_0 (1 - \bar{r}_1)\bar{r}_2 - \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m+\frac{1}{2},n+1,l}\Delta x_1 \Delta x_2 &= \rho_q \int_{r_0^s}^{r_0^{s+1}} r_1(t)(1 - r_2(t)) dr_0 \\
&= \rho_q \delta x_0 \bar{r}_1 (1 - \bar{r}_2) - \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, .
\end{aligned} \tag{2.42}$$

Contributions to the other components can be obtained from Equations (2.41) and (2.42) by cyclic permutation of the indices. For the components in the $\hat{\mathbf{e}}_1$:

$$\begin{aligned}
J_{m+1,n+\frac{1}{2},l+1}\Delta x_0 \Delta x_2 &= \rho_q \delta x_1 \bar{r}_0 \bar{r}_2 + \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m,n+\frac{1}{2},l}\Delta x_0 \Delta x_2 &= \rho_q \delta x_1 (1 - \bar{r}_0)(1 - \bar{r}_2) + \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m+1,n+\frac{1}{2},l}\Delta x_0 \Delta x_2 &= \rho_q \delta x_1 \bar{r}_0 (1 - \bar{r}_2) - \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m,n+\frac{1}{2},l+1}\Delta x_0 \Delta x_2 &= \rho_q \delta x_1 (1 - \bar{r}_0)\bar{r}_2 - \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12}
\end{aligned} \tag{2.43}$$

and for the components in the $\hat{\mathbf{e}}_2$ direction:

$$\begin{aligned}
J_{m+1,n+1,l+\frac{1}{2}}\Delta x_0 \Delta x_1 &= \rho_q \delta x_2 \bar{r}_0 \bar{r}_1 + \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m,n,l+\frac{1}{2}}\Delta x_0 \Delta x_1 &= \rho_q \delta x_2 (1 - \bar{r}_0)(1 - \bar{r}_1) + \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m,n+1,l+\frac{1}{2}}\Delta x_0 \Delta x_1 &= \rho_q \delta x_2 (1 - \bar{r}_0)\bar{r}_1 - \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, , \\
J_{m+1,n,l+\frac{1}{2}}\Delta x_0 \Delta x_1 &= \rho_q \delta x_2 \bar{r}_0 (1 - \bar{r}_1) - \rho_q \frac{\delta x_0 \delta x_1 \delta x_2}{12} \, .
\end{aligned} \tag{2.44}$$

A scheme of what is integrated is shown in Figure 2.5.

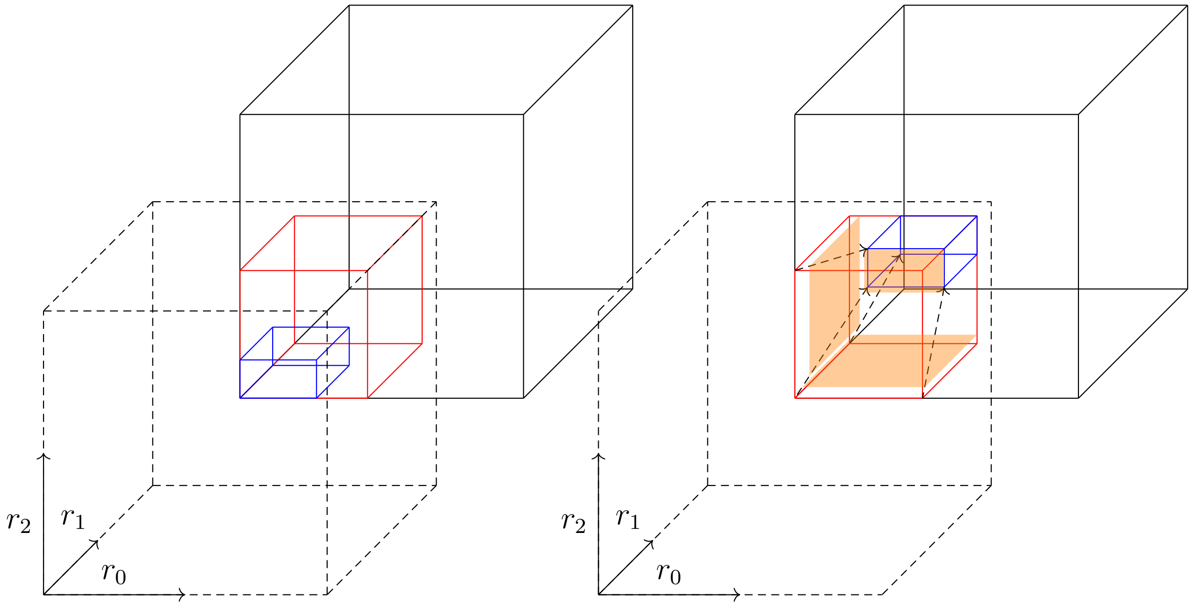

Figure 2.5: Visualization of the current integrals calculated in the Villasenor-Buneman method. The dashed cell represents the Yee mesh cell and the solid cell represents the staggered grid on which charge is conserved. Left: The blue and red boxes denote the particle's charge contribution arising from its overlap with the solid cell before and after its movement. Right: The initial contribution is shown at its new position after the particle has moved. The orange surfaces depict the integrated areas used in the derivation of the Villasenor-Buneman method.

To verify that the scheme is charge-conserving, we calculate the total change in charge within the volume spanned from $\mathbf{x}_{m+1,n+1,l+1}$ to $\mathbf{x}_{m+2,n+2,l+2}$ during one time step:

$$\begin{aligned}\Delta q &= J_{m+\frac{1}{2},n+1,l+1}\Delta x_1\Delta x_2 + J_{m+1,n+\frac{1}{2},l+1}\Delta x_0\Delta x_2 + J_{m+1,n+1,l+\frac{1}{2}}\Delta x_0\Delta x_1 \\ &= \rho_q\left(r_0^{s+1}r_1^{s+1}r_2^{s+1} - r_0^s r_1^s r_2^s\right) .\end{aligned} \tag{2.45}$$

This is precicesly the difference between the particle's charge contribution into the cell before and after the move. A similar check can be performed for the other cells as well. We note that, due to the linearity of the current continuity equation, this check for a single particle is sufficient to confirm that the entire method is charge conserving.

2.5.2 ZigZag method

As stated earlier, in order to apply the Villasenor-Buneman method, the particle trajectory must be split into sub-trajectories, each confined to a single Yee mesh cell. In the ZigZag method, these sub-trajectories pass through a relay point. Assuming that the particle position $\mathbf{x}^s$ lies in the cell spanned between $\mathbf{x}_{a_0,a_1,a_2}$ and $\mathbf{x}_{a_0+1,a_1+1,a_2+1}$, and the position

$\mathbf{x}^{s+1}$ lies in the cell spanned between $\mathbf{x}_{b_0,b_1,b_2}$ and $\mathbf{x}_{b_0+1,b_1+1,b_2+1}$, then the relay point $\mathbf{R}$ is defined as:

$$R_i = \begin{cases} \frac{x_i^s + x_i^{s+1}}{2} & \text{for } a_i = b_i \\ \Delta x_i \max(a_i, b_i) & \text{for } a_i \neq b_i \,. \end{cases} \tag{2.46}$$

Figure 2.6 illustrates the four different types of movements involving the relay point.

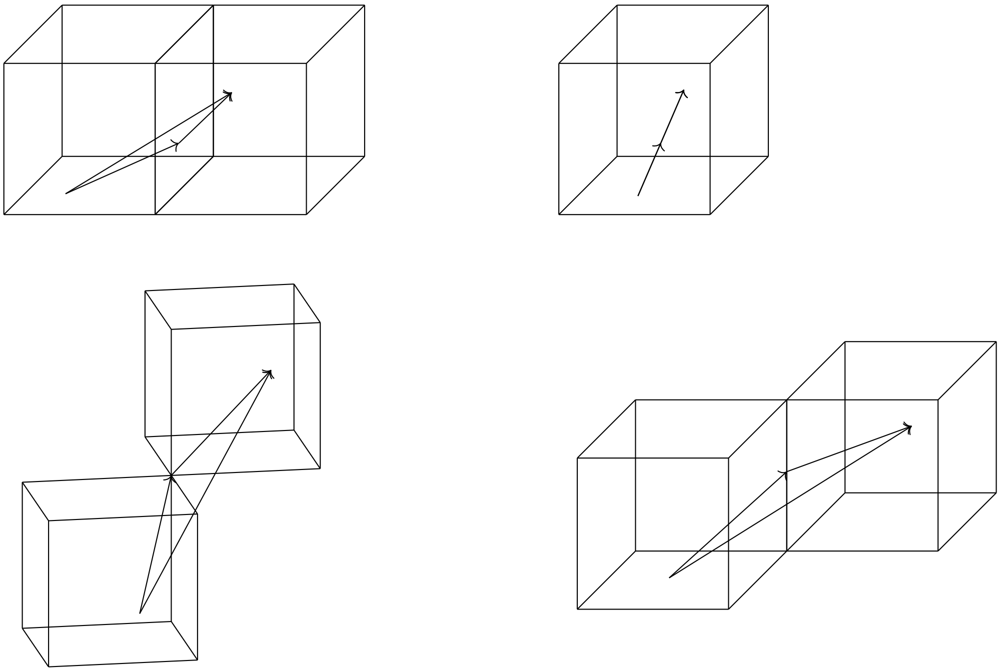

Figure 2.6: The four types of movements in ZigZag current depositer.

The benefit of splitting the trajectory in this manner is that the relay point coordinates can be computed without conditionals by using the algorithm described in Umeda et al. (2003):

$$R_i = \Delta x_i \min\left[\min(a_i, b_i) + 1, \max\left\{\max(a_i, b_i), \frac{x_i^s + x_i^{s+1}}{2}\right\}\right] \,. \tag{2.47}$$

After the relay point has been determined, the Villasenor-Buneman method can be applied to both subtrajectories. Compared to the original scheme described in Villasenor & Buneman (1992), the conditionless ZigZag achieves superior computational performance.

3. Electron Beam-Plasma Instabilities

The beam-plasma system consists of two homogeneous, counter-streaming pair-plasma populations. In this chapter, we provide a brief overview of the instabilities that arise in such a system, derive the maximum growth rates in the cold-fluid limit, and show that these agree with the measured growth rates from symmetric beam-plasma simulations performed with GPU-accelerated Runko. In the cold-fluid limit and for symmetric beams, the fastest-growing instability is the so-called filamentation instability, whose unstable modes are fully electromagnetic (Bret et al., 2004). Matching these growth rates therefore demonstrates the correctness of Runko's ability to resolve the full set of electromagnetic Vlasov-Maxwell equations.

3.1 Overview

Beam-plasma systems have been studied since Langmuir (1925) first observed the existence of oscillations in such systems (Briggs, 1964). Later, Pierce (1948) explained Langmuir's observations by demonstrating that unstable oscillations can arise in such systems. This prompted Bohm and Gross to develop the kinetic theory of ”two-stream instability”, which contains unstable waves propagating along the beam direction (Bohm, 1949). Fried (1959) showed that a second kind of instability exists, in which electromagnetic perturbations perpendicular to the flow become unstable. These modes tend to break up an initially homogeneous beam profile into small-scale current filaments, giving rise to the name ”filamentation instability”. Fried's article also mentioned closely related work by Weibel (1959), which demonstrated the instability of an anisotropic, two-temperature Maxwellian plasma. In literature, these two instabilities have become almost interchangeable, even though they are technically not equivalent (Bret et al., 2010). In this thesis, we use the term ”filamentation instability” for clarity.

In addition to the parallel two-stream modes and perpendicular filamentation modes, the stability of oblique modes can also be investigated. This was promptly done in the cold-fluid limit, where it was found that the unstable spectrum is two-dimensional, as the oblique perturbations are likely to be unstable (Watson et al., 1960; Bludman et al., 1960; Watson et al., 1970). In 21st century, Bret et al. carried out comprehensive kinetic treatments that provided a unified description of the entire unstable spectrum, including temperature effects (Bret et al., 2004; Bret, 2005a,b), supported by numerous PIC simulations (Gremillet et al., 2007; Dieckmann et al., 2006; Frederiksen & Dieckmann, 2008; Kong et al., 2009; Karmakar, 2009).

3.2 Growth rate in the cold-fluid limit

As stated at the beginning of this chapter, our system consists of two homogeneous, counter-streaming pair-plasma populations. Ions are treated as a fixed, neutralizing background. We refer to the two populations as the beam and the background plasma, and denote their number densities as n_b and n_p, respectively. Similarly, we denote their bulk velocities by $\mathbf{V}_b$ and $\mathbf{V}_p$. Initially, the system is current- and charge-neutral:

$$n_{b0}\mathbf{V}_{b0} = n_{p0}\mathbf{V}_{p0} \tag{3.1}$$

and without electromagnetic fields. We define the beam to propagate along the x-axis. Without loss of generality, we assume in the following discussion that the wave vector lies in the xy-plane.

Following the symbolic-calculator technique from Bret (2007), we solve for $\mathbf{V}_{b1}$ and $\mathbf{V}_{p1}$ from the linearized momentum transport equation (see Equation 1.26) and insert them to the linearized continuity equation (see Equation 1.25) to obtain n_{b1} and n_{p1}. These are then inserted into the linearized wave equation (see Equation 1.27). The resulting expression can be rearranged into a matrix form (Bret et al., 2004; Bret, 2012):

$$\mathbf{T}(\mathbf{k},\omega)\hat{\mathbf{E}}_1 = 0\,, \tag{3.2}$$

where $\mathbf{T}(\mathbf{k},\omega)\hat{\mathbf{E}}_1$ is a matrix-vector product between the matrix $\mathbf{T}(\mathbf{k},\omega)$ and the electric field perturbation $\hat{\mathbf{E}}_1$. The matrix has the following form:

$$\mathbf{T}(\mathbf{k},\omega) = \frac{\omega^2}{c^2}\boldsymbol{\epsilon}(\mathbf{k},\omega) + \mathbf{k}\otimes\mathbf{k} - k^2\mathbb{I}\,, \tag{3.3}$$

where $\mathbb{I}$ is the unit matrix and $\boldsymbol{\epsilon}(\mathbf{k},\omega)$ is the dielectric tensor:

$$\mathbf{T}(\mathbf{k},\omega) = \begin{bmatrix} \frac{\omega^2}{c^2}\epsilon_{xx} - k_z^2 & 0 & \frac{\omega^2}{c^2}\epsilon_{xz} + k_z k_x \\ 0 & \frac{\omega^2}{c^2}\epsilon_{yy} - k^2 & 0 \\ \frac{\omega^2}{c^2}\epsilon_{xz} + k_x k_z & 0 & \frac{\omega^2}{c^2}\epsilon_{zz} - k_x^2 \end{bmatrix} \tag{3.4}$$

Equation (3.2) admits solutions only when the determinant of $\mathbf{T}(\mathbf{k},\omega)$ vanishes, which results in two branches:

$$\epsilon_{yy} = \frac{c^2k^2}{\omega^2} \tag{3.5}$$

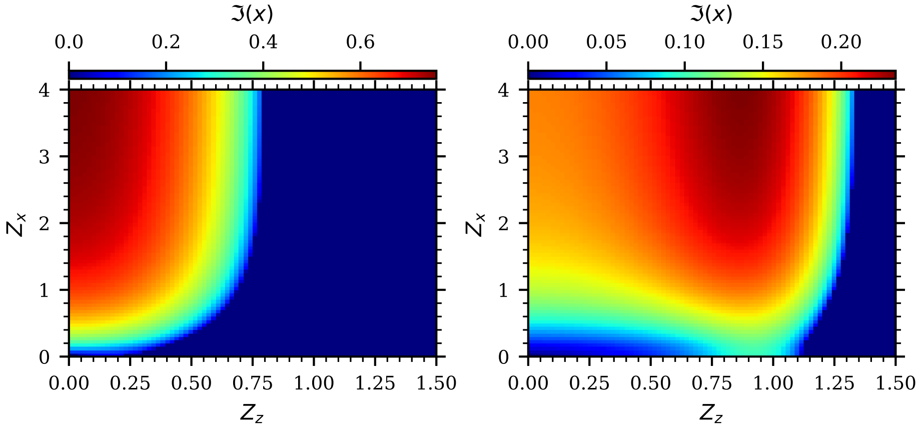

Figure 3.1: Growth rates from dispersion relation seen in Equation (3.6) for $\gamma_b = 3$. Left is the symmetric case $\alpha = 1$ and right is the diluted case $\alpha = 0.1$.

and

$$\left(\frac{\omega^2}{c^2}\epsilon_{xx} - k_z^2\right)\left(\frac{\omega^2}{c^2}\epsilon_{zz} - k_x^2\right) = \left(\frac{\omega^2}{c^2}\epsilon_{xz} + k_x k_z\right)^2 \tag{3.6}$$

For the instabilities, the first branch is not interesting, as the second holds solutions corresponding to the two-stream, the oblique and the filamentation instabilities. To investigate the nature of the solutions, we first define standard dimensionless variables:

$$x = \frac{\omega}{\omega_{pp}}, \qquad \mathbf{Z} = \frac{\mathbf{k}V_{b0}}{\omega_{pp}}, \qquad \alpha = \frac{n_{b0}}{n_{p0}} \tag{3.7}$$

Now x can be solved numerically from dispersion relation shown in Equation (3.6). Figure 3.1 presents the normalized growth rate (i.e. imaginary part of x) as a function of $\mathbf{Z}$ for two different cases. For the diluted case ($\alpha \ll 1$), we see that the largest growth rate occurs in the oblique-angle regime, while for the symmetric case ($\alpha = 1$) the largest growth is at $Z_z = 0$ and $Z_x \to \infty$, corresponding to the filamentation instability.

For the transverse case $k_z = Z_z = 0$, the components of dielectric tensor $\boldsymbol{\epsilon}(\mathbf{k}, \omega)$

simplify to (Bret, 2012)*:

$$\epsilon_{xx} = 1 - \frac{\alpha}{\gamma_b x^2} - \frac{1}{\gamma_p x^2}, \tag{3.8}$$

$$\epsilon_{yy} = 1 - \frac{\alpha}{\gamma_b x^2} - \frac{1}{\gamma_p x^2}, \tag{3.9}$$

$$\epsilon_{zz} = 1 + \frac{Z_x x^2 \alpha^2}{\gamma_p x^4} - \frac{Z_x x^2 \alpha}{\gamma_b x^4} - \frac{\alpha}{\gamma_b^3 x^2} - \frac{1}{\gamma_p^3 x^2}, \tag{3.10}$$

$$\epsilon_{xz} = \frac{Z_x \alpha}{x^3} \left(\frac{1}{\gamma_p} - \frac{1}{\gamma_b} \right) \tag{3.11}$$

Inserting these into the dispersion relation (see Equation 3.6) and taking the limit $Z_x \to \infty$, the growth rates for symmetric and diluted cases of filamentation instability can be solved exactly:

$$\Im(x) = \frac{V_b}{c} \sqrt{\frac{\alpha}{\gamma_b}}, \quad \alpha \ll 1 \tag{3.12}$$

$$\Im(x) = \frac{V_b}{c} \sqrt{\frac{2}{\gamma_b}}, \quad \alpha = 1 \tag{3.13}$$

3.3 Numerical validation

GPU-accelerated Runko was used to simulate filamentation instability of symmetric beams ($\alpha = 1$) following the setup from (Nättilä, 2022). The two counter-streaming pair-plasma populations were initialized to flow along $\hat{\mathbf{e}}_0$ with spatial components of a four-velocity $\mathbf{u} = \pm\gamma_b \hat{\mathbf{e}}_0$, where γ_b is the Lorentz factor shared between the symmetric beams. The plasmas were modeled with 32 particles per cell per species, which initial positions were sampled from a unifrom distribution and velocities were sampled from Jüttner-Synge distribution (Zenitani, 2015) with thermal spread of $k_b T/(mc^2) = 10^{-5}$, where k_b is the Boltzmann constant, T is the temperature, m is the particle mass, and c is the speed of light. Initially, the electromagnetic fields were set to zero. The periodic simulation domain consisted of a grid of $320 \times 80 \times 6$ uniform cells and the skin depth was resolved with 10 cells. The time step was related to the cell width as: $\Delta t = 0.45 \Delta x$.

For a comparison of the simulation results with the analytical growth rate of Equation (3.13), Figure 3.2 shows the average magnetic-field energy density, normalized by the relativistic plasma enthalpy density, as a function of time. We note that the analytical growth rate matches the numerical growth rates perfectly during the linear growth phase of the instability. However, the linear phase does not begin immediately, as the instability must first grow from numerical noise, and it takes some time before it begins to

*Note that in (Bret, 2012) powers of -3 are missing from Equation (3.10) and there is extra γ_p^{-1} in Equation (3.11).

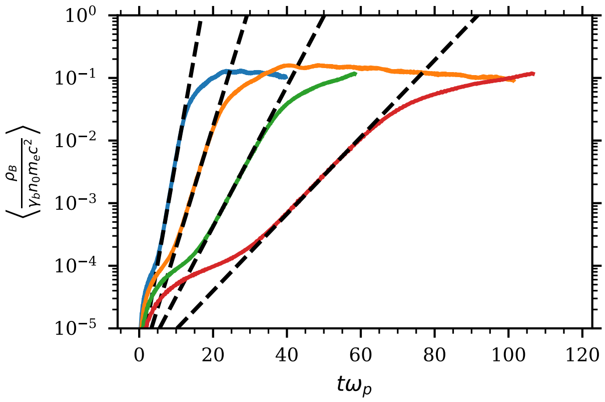

Figure 3.2: Simulation of the filamentation instability using the GPU accelerated Runko for different values of γ_b. Solid curves represent the averagemagnetic-field energy density $\rho_B = \frac{B^2}{8\pi}$, normalized by the relativistic plasma enthalpy density, $\gamma_b n_0 m_e c^2$, as a function of time $t\omega_p$. From left to right, the values for γ_b are 3 (blue), 10 (orange), 30 (green), and 100 (red). The dashed lines show the analytic growth rates from the linearized cold-fluid limit.

dominate. At a later stage, nonlinear effects terminate the linear growth phase, and the growth saturates. Qualitatively, the formed filaments begin to collide with each other, disturbing the system sufficiently to halt further growth.

4. Conclusions

In this thesis, we introduced the necessary background theory of plasma physics and finite-difference methods, presented the key components of the PIC method, derived the growth rate of the filamentation instability in the cold-fluid limit, and demonstrated the ability of the GPU-accelerated Runko to simulate the filamentation instability correctly.

This thesis concludes the first stage of porting Runko to GPUs, in which only the minimal viable set of features was ported, without yet focusing on performance. Due to the architectural differences between CPUs and GPUs, significant parts of the code did not translate well and had to be rewritten entirely. Before investing resources in performance optimization, the goal was to show that the GPU port is feasible and that it produces correct physical results.

In the future, the focus will shift more toward performance and how to effectively utilize modern supercomputers. Efficient use of supercomputers would enable Runko to perform simulations that push the boundaries of computational astrophysics. Currently, the main performance bottleneck lies in particle communication. Addressing this will require redesigning how particles are handled. One potential option is to adopt tag-based particle handling, as demonstrated in (Hakobyan et al., 2025). Further work is also required to compare the performance of the GPU-accelerated Runko with other PIC codes.

In addition to performance optimizations, new features will be added to the GPU-accelerated Runko. Currently, its feature set is limited compared to the original version. For example, dynamically injecting particles into a running simulation and using user-defined boundary conditions are not yet supported by the GPU implementation. Furthermore, alternative field-propagation schemes, field-interpolation methods, particle pushers, and current-deposition algorithms can be implemented when needed.

Beyond the standard PIC framework, additional physics modules can also be incorporated into the GPU version. One possible direction is to implement the capability to run PIC simulations in curved spacetimes (e.g., Schwarzschild or Kerr geometries). Another avenue is to port existing quantum-electrodynamics effects–already available in the CPU version of Runko–to GPUs (Nättilä, 2024).

Bibliography

Bludman, S. A., Watson, K. M., & Rosenbluth, M. N. 1960, The Physics of Fluids, 3, 747, doi: 10.1063/1.1706121

Bohm, D. 1949, Physical Review, 75, 1851, doi: 10.1103/PhysRev.75.1851

Boris, J. 1970, in Proceedings of the Fourth Conference on the Numerical Simulation of Plasmas (RESEARCH LABORATORY, WASHINGTON, D.C.: Naval Research Laboratory Washington, D.C.), 3–67

Bret, A. 2005a, Physical Review E, 72, doi: 10.1103/PhysRevE.72.016403

—. 2005b, Physical Review Letters, 94, doi: 10.1103/PhysRevLett.94.115002

—. 2007, Beam-plasma dielectric tensor with Mathematica, arXiv, doi: 10.48550/arXiv.physics/0701257

—. 2012, CfA Plasma Talks, arXiv, doi: 10.48550/arXiv.1205.6259

Bret, A., Firpo, M.-C., & Deutsch, C. 2004, Physical Review E, 70, 046401, doi: 10.1103/PhysRevE.70.046401

Bret, A., Gremillet, L., & Dieckmann, M. E. 2010, Physics of Plasmas, 17, 120501, doi: 10.1063/1.3514586

Briggs, R. J. 1964, Electron-Stream Interaction with Plasmas (The MIT Press), doi: 10.7551/mitpress/2675.001.0001

Buneman, O. 1959, Physical Review, 115, 503, doi: 10.1103/PhysRev.115.503

Dawson, J. 1962, The Physics of Fluids, 5, 445, doi: 10.1063/1.1706638

De Groot, S. R. 1980, Relativistic Kinetic Theory. Principles and Applications, ed. W. A. Van Leeuwen & C. G. Van Weert

Dieckmann, M. E., Frederiksen, J. T., Bret, A., & Shukla, P. K. 2006, Physics of Plasmas, 13, 112110, doi: 10.1063/1.2390687

Eastwood, J., Arter, W., Brealey, N., & Hockney, R. 1995, Computer Physics Communications, 87, 155, doi: https://doi.org/10.1016/0010-4655(94)00165-X

Eastwood, J. W. 1991, Computer Physics Communications, 64, 252, doi: https://doi.org/10.1016/0010-4655(91)90036-K

Esirkepov, T. 2001, Computer Physics Communications, 135, 144, doi: https://doi.org/10.1016/S0010-4655(00)00228-9

Frederiksen, J. T., & Dieckmann, M. E. 2008, Physics of Plasmas, 15, 094503, doi: 10.1063/1.2985776

Fried, B. D. 1959, The Physics of Fluids, 2, 337, doi: 10.1063/1.1705933

Goldstein, H., Safko, J., & Poole, C. 2014, Classical Mechanics: Pearson New International Edition (Pearson Education). https://books.google.fi/books?id=Xr-pBwAAQBAJ

Gremillet, L., Bénisti, D., Lefebvre, E., & Bret, A. 2007, Physics of Plasmas, 14, 040704, doi: 10.1063/1.2714509

Hakobyan, H., Böss, L. M., Cai, Y., et al. 2025, Entity – Hardware-agnostic Particle-in-Cell Code for Plasma Astrophysics. I: Curvilinear Special Relativistic Module, arXiv, doi: 10.48550/arXiv.2511.17710

Hazeltine, R. D. 2018, The Framework Of Plasma Physics (Boca Raton: CRC Press), doi: 10.1201/9780429502804

Higuera, A. V., & Cary, J. R. 2017, Physics of Plasmas, 24, 052104, doi: 10.1063/1.4979989

Hockney, R. W., & Eastwood, J. W. 1981, Computer Simulation Using Particles

Karmakar, A. 2009, Physical Review E, 80, doi: 10.1103/PhysRevE.80.016401

Kilpua, E., Koskinen, H., & Helsingin yliopisto, j. 2017, Introduction to plasma physics / Emilia Kilpua and Hannu Koskinen., first edition. edn. (Helsinki: Helsingin yliopisto). https://doi.org/10.31885/9789515183095

Kirkwood, R. K., Moody, J. D., Kline, J., et al. 2013, Plasma Physics and Controlled Fusion, 55, 103001, doi: 10.1088/0741-3335/55/10/103001

Klimontovich, Y. L. 1960, Soviet Physics JETP, 37(10), 524. http://jetp.ras.ru/cgi-bin/dn/e_010_03_0524.pdf

Kong, X., Park, J., Ren, C., Sheng, Z. M., & Tonge, J. W. 2009, Physics of Plasmas, 16, 032107, doi: 10.1063/1.3088056

Langdon, A. B., C. K. B. 1991, Plasma Physics via Computer Simulation (Boca Raton: CRC Press), doi: 10.1201/9781315275048

Langmuir, I. 1925, Physical Review, 26, 585, doi: 10.1103/PhysRev.26.585

Nättilä, J. 2022, Astronomy & Astrophysics, 664, A68, doi: 10.1051/0004-6361/201937402

—. 2024, Nature Communications, 15, 7026, doi: 10.1038/s41467-024-51257-1

Pert, G. J. 2021, Foundations of Plasma Physics for Physicists and Mathematicians, 1st edn. (Wiley), doi: 10.1002/9781119774297

Pierce, J. R. 1948, Journal of Applied Physics, 19, 231, doi: 10.1063/1.1715050

Steiniger, K., Widera, R., Bastrakov, S., et al. 2023, Computer Physics Communications, 291, 108849, doi: 10.1016/j.cpc.2023.108849

Umeda, T., Omura, Y., Tominaga, T., & Matsumoto, H. 2003, Computer Physics Communications, 156, 73, doi: https://doi.org/10.1016/S0010-4655(03)00437-5

Umstadter, D. 2003, Journal of Physics D: Applied Physics, 36, R151, doi: 10.1088/0022-3727/36/8/202

Uzdensky, D. A., & Rightley, S. 2014, Reports on Progress in Physics, 77, 036902, doi: 10.1088/0034-4885/77/3/036902

Vay, J.-L. 2008, Physics of Plasmas, 15, 056701, doi: 10.1063/1.2837054

Vay, J.-L., & Godfrey, B. B. 2014, Comptes Rendus. Mécanique, 342, 610, doi: 10.1016/j.crme.2014.07.006

Verboncoeur, J. P. 2005, Plasma Physics and Controlled Fusion, 47, A231, doi: 10.1088/0741-3335/47/5A/017

Villasenor, J., & Buneman, O. 1992, Computer Physics Communications, 69, 306, doi: https://doi.org/10.1016/0010-4655(92)90169-Y

Watson, K. M., Bludman, S. A., & Rosenbluth, M. N. 1960, The Physics of Fluids, 3, 741, doi: 10.1063/1.1706120

—. 1970, Soviet Physics JETP, 30, 528. http://jetp.ras.ru/cgi-bin/dn/e_030_03_0528.pdf

Weibel, E. S. 1959, Physical Review Letters, 2, 83, doi: 10.1103/PhysRevLett.2.83

Wiesemann, K. 2014, A Short Introduction to Plasma Physics, doi: 10.5170/CERN--2013--007.85

Yee, K. 1966, IEEE Transactions on Antennas and Propagation, 14, 302, doi: 10.1109/TAP.1966.1138693

Zenitani, S. 2015, Physics of Plasmas, 22, 042116, doi: 10.1063/1.4919383