School of Computation, Information and Technology

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master’s Thesis
in
Computational Science and Engineering

Semi-Lagrangian Non Interpolating Schemes for Plasma Modelling

Michael Pio Basile

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis
in
Computational Science and Engineering

Semi-Lagrangian Non Interpolating Schemes for Plasma Modelling

Author:	Michael Pio Basile
Supervisor:	Prof. Dr. Eric Sonnendrücker, (TUM)
Advisor:	Dr. Michael Hardman, (Tokamak Energy)

I confirm that this master's thesis is my own work and I have documented all sources and material used.

13/01/2024

Abstract

This master's thesis investigates the application of Non-Interpolating Semi-Lagrangian (NISL) methods in plasma physics simulations. The study extends these methods to tackle partial differential equations specific to Plasma Physics Modeling.

The research begins with an in-depth review of existing literature, paving the way for a sequence of detailed experiments. An interesting result is that the use of Runge-Kutta (RK) methods as integrators in the Eulerian part of the scheme has proven to be highly effective. The study culminates with its most advanced simulation, applying these methods to the Plasma Sheath Formation model. This is achieved using RK integrators and WENO (Weighted Essentially Non-Oscillatory) discretization techniques. This method proves especially effective for simulating Sheath Formation in plasma, a task complicated by strong gradients due to diverging electric fields, which challenge many numerical solvers.

The methods are rigorously tested using analytical solutions and the Method of Manufactured Solutions, verifying their reliability and accuracy. The results show that NISL methods are a viable and computationally efficient alternative to standard semi-Lagrangian methods, offering more stable performance in high CFL (Courant-Friedrichs-Lewy) conditions.

An important conclusion of this thesis is the potential integration of this framework with common Eulerian codes, enabling simulations to surpass the CFL limitations for hyperbolic partial differential equations (PDEs).

In conclusion, the thesis demonstrates the effectiveness of NISL methods in advanced plasma physics simulations and sets the stage for future research by combining NISL with traditional Eulerian approaches. This integration promises to advance and streamline numerical methods in plasma physics.

Contents

1 Introduction

This thesis delves into a particular type of Semi-Lagrangian (SL) methods called Non Interpolating Semi-Lagrangian (NISL) methods and their application to plasma physics modeling. To introduce NISL methods, one must first acknowledge conventional SL methods. SL methods arise from the need in numerical weather prediction to obtain stable simulations that could predict meteorological events with good fidelity even days in advance. This practical need translates into the necessity of overcoming the limits of the CFL condition. To overcome this condition in the 60s-70s, the main options were to use implicit numerical integrators or to use a pure Lagrangian approach. However, the first option was computationally unfeasible due to the resolution of linear systems and the computational capabilities of the computers of that period. Whereas, the second option presented some important disadvantages. As a matter of fact, using a pure Lagrangian approach means solving the problem by following each individual "particle" within the system and resolving its trajectory. The first consequence is reaching a very high computational cost as the number of particles increases. Another issue in the Numerical Weather Prediction field of that period was that using uniformly distributed particles led to unreliable results in the long term, as the particles tended to accumulate in specific regions of the simulation. Therefore, starting from uniformly distributed points, solutions were obtained with points concentrated in certain areas, leading to a solution that was not reliable or accurate throughout the entire spatial region.

For these reasons, semi-implicit time integrators, which could use computationally efficient spatial discretization techniques like Finite-Difference (FD) without the need to solve linear systems, began to be investigated. Towards the end of the 20th century, among the first works introducing the fundamental concepts of semi-Lagrangian methods one can find those of Robert [1]. For the drafting of this thesis, another important source used is the review of Staniforth (1991) [2], which executed a very accurate classification of the types of SL methods, including the non-interpolating methods. This thesis follows the approach of Staniforth's review, starting from simple models i.g. advection, the numerical schemes will be applied to models of increasing difficulty.

To introduce SL methods coherently and solidly in this first chapter, the necessary theoretical information will be included to understand the conditions under which these methods can be applied and the fundamental ideas on which they are based. The Chapter 2 summarises the formulations of the models present in the literature that are used in this work . Chapter 3 replicates the experiments conducted by the authors cited in the previous chapter. Chapter 4 introduces a numerical method based on non-interpolating semi-Lagrangian techniques, applied on the Burger equation. Chapter 5 extends the methodologies developed in earlier chapters to the Vlasov-Poisson equation. Here, the escalating challenges of this model are met with a formulation of non-interpolating methods, integrating the renowned Runge-Kutta integrators. The final chapter delves into a more sophisticated and realistic model of Sheath Formation in plasma. This model, characterized by wall boundary conditions and the development of strong gradients, poses a significant challenge to most numerical solvers. To effectively address these robust gradients, a non-interpolating semi-Lagrangian solver, employing a WENO (finite difference) discretization, is presented.

1.1 Hyperbolic PDE and Conservation Laws

This section of the thesis strongly follows the theory from the early chapters of Leveque (1991) [2] and utilizes the expository method proposed in the lecture notes by Dr. W. Barsukow (2022) [3].

The term hyperbolic Partial Differential Equation (PDE) derives from a particular mathematical definition for second-order PDEs, specifically the algebraic structure of these PDEs. Intuitively, we can write a generic second-degree polynomial as:

$$P(X,Y) = \alpha_{20}X^2 + \alpha_{11}XY + \alpha_{02}Y^2 + \alpha_{10}X + \alpha_{01}Y + \alpha_{00}. \tag{1.1}$$

In which, if we plot $P(X,Y) = 0$, we can obtain three types of curves: an ellipse, a parabola, or a hyperbola. Based on the coefficients α, we can obtain one of these curves, so it makes sense to define the polynomial that traces a hyperbola as hyperbolic and so on.

Instead of X and Y, we can use as inputs of the polynomial differential operators $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ which are applied to a generic function f. We obtain:

$$\alpha_{20}\frac{d^2f}{dx^2} + \alpha_{11}\frac{d^2f}{dx\,dy} + \alpha_{02}\frac{d^2f}{dy^2} + \alpha_{10}\frac{df}{dx} + \alpha_{01}\frac{df}{dy} + \alpha_{00}f = 0. \tag{1.2}$$

Thus, using the same definition for polynomials based on the values of the coefficients, we can create a useful classification of PDEs. Using the definition for second-order PDEs, we find that the simplest is the wave equation, defined as a hyperbola:

$$\frac{d^2f}{dx^2} - \frac{d^2f}{dy^2} = 0. \tag{1.3}$$

It is important to specify that this classification is not exhaustive and is very limited: there are many types of PDEs that do not fall within this classification. A very simple example is the Schrödinger equation.

The physical characteristic that distinguishes hyperbolic PDEs is that the information from a certain perturbation of the initial state propagates with finite speed through space in a wave-like type. This characteristic has made these types of PDEs among the most studied in history.

1.1.1 Continuity equation

To properly introduce the basic principles of semi-Lagrangian methods, it's necessary to correctly define the meanings of conservation. Let's consider a density $\rho(t,\mathbf{x})$, for a simplified visualization we can think of it as a function that takes as input $\mathbb{R}_0^+ \times \mathbb{R}^d \to \mathbb{R}$ and returns a scalar value representing the density of a material at a specific point in time and space. Based on this, we can define the mass at a moment t of a volume Ω as:

$$m|_\Omega(t) = \int_\Omega \rho(t,\mathbf{x})d\mathbf{x}. \tag{1.4}$$

We can imagine a box with uniform density identified by the dashed line that moves from the initial position (box with a solid line) see Figure 1.1

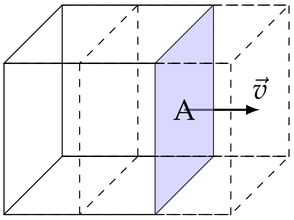


Figure 1.1: Example of ρ moving in v

Therefore, we find that the mass is leaving the initial volume according to a simple relation.

$$m(t) = m(0) - Avt\rho \tag{1.5}$$

Where A is the area of the surface of the face in lilac in Figure 1.1.

Once this simple idea is obtained, we can generalize with a velocity $\mathbf{v}$, which is a vector field over a generic volume $\Omega \subset R^d$, in which our face of the cube A becomes generally a smooth boundary $\partial\Omega$. Generally, instead of considering a finite Δt, we pass to its infinitesimal variation, obtaining the following equation.

$$\frac{d}{dt} m|_{\Omega}(t) = -\int_{\partial\Omega} (\mathbf{v}(t, \mathbf{y}) \cdot \mathbf{n}) \rho(t, \mathbf{y}) d\mathbf{y} \tag{1.6}$$

In which, by inserting the definition of mass starting from the integral on the density, we obtain the integral form of the continuity equation.

$$\frac{d}{dt} \int_{\Omega} \rho(t, \mathbf{x}) d\mathbf{x} = -\int_{\partial\Omega} (\mathbf{v}(t, \mathbf{y}) \cdot \mathbf{n}) \rho(t, \mathbf{y}) d\mathbf{y} \tag{1.7}$$

It is important to note the difference in the dimensions of x and y. In fact, dy is an infinitesimal quantity of the surface of the boundaries $\partial\Omega$, while dx is an infinitesimal unit of the volume Ω itself. Starting from equation1.7, we can define the flux of the quantity $q(t, x)$ through the surface $\partial\Omega$ along the normal $\mathbf{n}$.

$$\frac{d}{dt} \int_{\Omega} \rho(t, \mathbf{x}) d\mathbf{x} = -\int_{\partial\Omega} \mathbf{n} \cdot \mathbf{f}(q(t, y)) d\mathbf{y} \tag{1.8}$$

Where using the Gauss law it is possible to replace the integral over the boundary with an integral in the volume.

$$\frac{d}{dt} \int_{\Omega} \rho(t, \mathbf{x}) d\mathbf{x} = -\int_{\Omega} \nabla \cdot (\mathbf{v}(t, \mathbf{x}) \rho(t, \mathbf{x})) d\mathbf{x} \tag{1.9}$$

By validating this law for every Ω, we can generalize this integral equation into a PDE.

$$\partial_t \rho(t, \mathbf{x}) + \nabla_{\mathbf{x}} \cdot (\mathbf{v}(t, \mathbf{x}) \rho(t, \mathbf{x})) = 0 \tag{1.10}$$

We have obtained the form of a conservation law. By defining the flux $\mathbf{v}(t, \mathbf{x})\rho(t, \mathbf{x})$, its solutions can be studied. Intuitively, this form tells us that the variation of the quantity ρ over time is equal to the variation of the quantity in the spatial dimension. As the quantity varies over time simply by moving along the spatial dimensions, one can speaks of transport. In particular, transport occurs through the definition of flux. By defining simple fluxes, it is possible to find an analytical solution to the differential form; however, in most real-world applications, the form of the flux assumes strongly non-linear behaviors that prevents from finding strong solutions to the PDEs.

In general, a law is called a conservation law if it is written in a conservative form of the type.

$$\partial_t q + \nabla_{\mathbf{x}} \cdot f(x, t) = 0 \tag{1.11}$$

1.1.2 Advective derivative

So far, we have presented the mathematical concepts behind the idea of conservation laws. In the previous steps, we started from the physical concept of Figure 1.1, where, while keeping our volume fixed (solid line), we let the flux pass through and considered the variation of the quantity inside our volume. What has been done so far is called the Eulerian perspective, where, keeping the coordinates of the volume fixed, we considered the flow moving within our volume. Another approach is to consider the volume moving following the flow. This point of view is defined as Lagrangian. This approach is of particular interest in the field of particle simulations. In fact, in most Particle in Cell (PIC) solvers [4], the volume to follow in the flow is exactly a particle (or macro particle).

When considering a volume that moves within space, we can define a trajectory that depends on the velocity field. Let's consider a scalar function that takes as input a spatial coordinate and a instant $q : \mathbb{R}_0^+ \times \mathbb{R}^d \to \mathbb{R}$, and a curve $\xi(t) : \mathbb{R}_0^+ \to \mathbb{R}^d$ that, given an instant t, returns the position of our volume in d-dimensional space. This trajectory is also defined as a pathline, and it is very convenient to use it to analyze how our quantity evolves as it moves along this trajectory. To evaluate the temporal evolution of our scalar quantity within the trajectory, we assess the total derivative using a simple chain rule.

$$\frac{d}{dt}q(t,\xi(t)) = \partial_t q(t,\xi(t)) + \xi'(t) \cdot \nabla q(t,\xi(t)) \tag{1.12}$$

Where the derivative of the trajectory over time is the definition of velocity.

$$\frac{d}{dt}q(t,\xi(t)) = (\partial_t + v(t,\xi(t)) \cdot \nabla)q(t,\xi(t)) \tag{1.13}$$

The previous equation is defined in this context as the material derivative (also known as "Lagrangian derivative" or "advective derivative") and will be of fundamental importance in the steps to obtain the schemes of SL methods. It is relevant to note that the pathline $\xi(t)$ is uniquely identified in space through an initial position, this can be explained considering that thetrajectory depends on the solution of an ODE (Ordinary Differential Equation) thus it requires initial conditions to be defined correctly. The formulation that will be adopted in the following chapters will use a more simple notation of the form:

$$\frac{D}{Dt} = \partial_t + \mathbf{v} \cdot \nabla \tag{1.14}$$

1.2 Method of Characteristichs

Using the form of a conservation law as in the 1-D equation 1.11, where we have our quantity q being transported, we can write the following PDE.

$$\partial_t q(t,x) + \partial_x f(q(t,x)) = 0 \quad \mathbb{R}_0^+ \times \mathbb{R} \to \mathbb{R} \tag{1.15}$$

Consider that our solution and the flux $q, f \in C^1$. Then we can define curves where our solution is constant over time.

$$\xi' = f'(q(t,\xi(t))), \quad \xi : \mathbb{R}_0^+ \to \mathbb{R} \tag{1.16}$$

In this way, we have defined an ODE whose solutions are curves where the initial condition is transported. These curves are called characteristics of the PDE. Hence, by defining an initial point x_0 where our quantity had value q_0 at the initial moment, in some particular cases we can obtain a strong solution.

1.2.1 Constant Advection and Method of Characteristics

We consider a constant velocity advection.

$$\partial_t q + \overline{v} \partial_x q = 0 \tag{1.17}$$

To solve this PDE, we can use the method of characteristics. The characteristic curves are defined by the ordinary differential equation (ODE):

$$\frac{d\xi(t;x_0)}{dt} = \overline{v}_x, \tag{1.18}$$

Where $\xi(t;x_0)$ is our characteristic curve with initial condition $\xi(0) = x_0$. A graphical explanation is reported in Figure1.2.

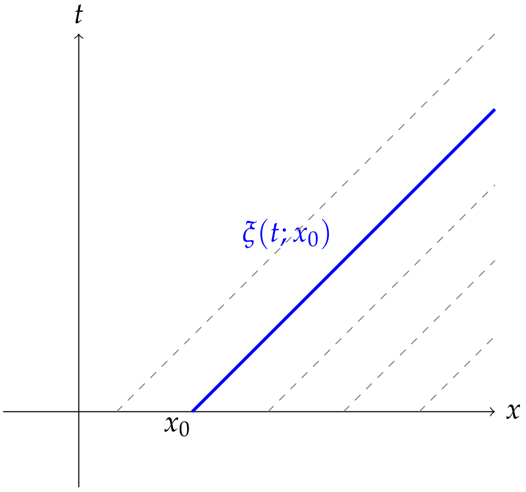


Figure 1.2: Example of a Characteristic $\xi(t;x_0)$

Since $\overline{v}$ is constant, the solution to this ODE is straightforward:

$$\xi(t) = \boldsymbol{x}_0 + \overline{\boldsymbol{v}}(t - t_0). \tag{1.19}$$

Along these characteristic curves, the solution $u(\boldsymbol{x}, t)$ remains constant, meaning $u(\xi(t), t) = u(\boldsymbol{x}_0, t_0)$. To find the solution at time t, we can express it in terms of the initial condition at t_0:

$$q(\boldsymbol{x}, t) = q(\boldsymbol{x}_0 + \overline{\boldsymbol{v}}(t - t_0), t_0). \tag{1.20}$$

This equation describes the solution to the advection equation with constant velocity and no sources. The solution at any time t can be found by tracing back the characteristic curve to the initial time t_0, and evaluating the initial condition at the corresponding spatial position.

1.2.2 Burger Equation and Method of Characteristics

If we consider the flux $f = q^2/2$ we can write the equation in conservative form.

$$\partial_t q + \partial_x \left(\frac{q^2}{2}\right) = 0 \tag{1.21}$$

This is called Burger equation[5] and it is the perfect example of non-linear PDE. In order to study with the method of characteristics this equation we define a simple initial condition

$q_0(x) = 1 - x$. This means that the quantinty $q_0(x_0)$ will move on the characteristic with velocity $1 - x_0$ which gives the position.

$$\xi(t) = x_0 + (1 - x_0)t. \tag{1.22}$$

We can write the solution for this simple initial condition.

$$q(x,t) = q_0(x_0) = q_0\left(\frac{x-t}{1-t}\right) = \frac{1-x}{1-t} \tag{1.23}$$

In this simple example proposed by Dr. W. Barsukow [3] it's easy to verify that for $t = 1$ the solution doesn't exist anymore in $\mathbb{C}^1$. In general this is the case also with initial conditions that are in $\mathbb{C}^\infty$.

An interesting property of the burger equation is that the characteristics are always straight lines. Following the demonstration by LeVeque [6].

$$\xi'(t) = q(\xi(t), t) \tag{1.24}$$

$$\frac{d}{dt}q(\xi(t), t) = \partial_t q + \xi' \partial_x q = \partial_t q + q(\xi(t), t)\partial_x q \tag{1.25}$$

Where we get the definition of our PDE.

$$\partial_t q + q(\xi(t), t)\partial_x q = 0 \tag{1.26}$$

Since we have verified that the solution is constant over the characteristic and since the slope of the characteristic is equal to the solution by definition we have verified that the slope is constant and is a straight line which depends only on the initial condition.

2 Semi-Lagrangian Methods

2.1 The Problem of the Starting Point

To introduce the general idea of semi-Lagrangian methods, let's consider a one-dimensional advection model with constant velocity as an example:

$$\frac{\partial u}{\partial t} + \overline{v}_x \frac{\partial u}{\partial x} = 0 \tag{2.1}$$

where $\overline{v}_x$ denotes the constant velocity in the x-direction. The solution to this equation manifests as a function of the form $u(x,t) = u(x - \overline{v}_x t, 0)$.

This PDE can be viewed as the total derivative of the function u with respect to time along the trajectory of a particle. Historically, this total derivative is also referred to as the material or Lagrangian derivative, a concept rooted in a Lagrangian approach to fluid description. Here, instead of approximating a continuous variable, the aim is to trace a "particle" or material point within the flow. This enables us to express the solution as:

$$\frac{du}{dt} = 0 \tag{2.2}$$

Through the chain rule, this equation can be rewritten as:

$$\frac{\partial u}{\partial t} + \frac{dx}{dt}\frac{\partial u}{\partial x} = 0 \tag{2.3}$$

By writing it in this manner, it becomes evident that the quantity is conserved along the trajectory, where $dx/dt = \overline{v}_x$ is the velocity of the particle. The solution to this ordinary differential equation (ODE) is also known as the characteristic curve of the PDE, which represents the trajectory of our particle.

Knowing the analytical solution of the characteristic allows us to understand the trajectory of any particle at any given moment in time. However, as we will see in the following chapters, the opportunity to analytically solve characteristics is relatively rare. For this reason, numerical methods are used to compute the particle trajectory, particularly semi-Lagrangian methods.

Interestingly, in this numerical formulation, it is possible to adopt analytical results for specific PDE models, which allow for an accurate approximation of certain characteristics.

Semi-Lagrangian approaches originate from the solution of the Lagrangian derivative (see eq. 2.2) through the approximation of the derivative using finite differences. A peculiar aspect of semi-Lagrangian methods is that we do not follow a particle as in pure Lagrangian methods. Instead, we start from a grid node and use the trajectory of a "pseudo particle" or characteristic, which we trace and follow backwards. At the end of each timestep, our pseudo particles end up in a fixed position.

This approach is used to address one of the main problems of pure Lagrangian methods, which results in non-uniform particle distributions unsuitable for solution representation after simulation [2].

To formulate our approximation of the Lagrangian derivative, let's introduce a graphical representation of the approximation we are considering.

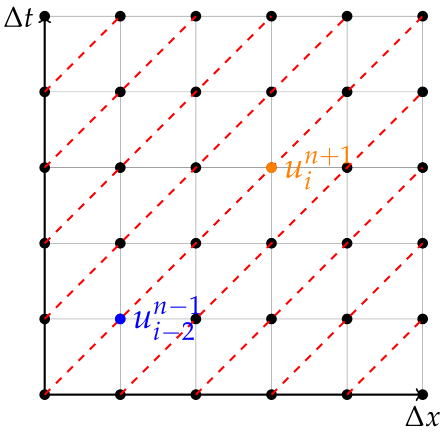


Figure 2.1: Example of a simple grid with the characteristic curve

In this figure, we have the simplest example of advection. We are looking at an advection scenario where the velocity is constant and equal to $\overline{v}_x * \Delta x / \Delta t$ (where $\overline{v}_x$ is an integer constant). In this elementary example, the solution of the characteristic curve (depicted as a red dashed line) is not only a straight line, but it also precisely intersects the nodes of our grid. This example serves as an excellent exercise to gain insight into semi-Lagrangian methods.

To approximate our solution at a point, for example u_i^{n+1}, with the SL approach, we employ the following scheme applied to the equation 2.2:

$$\frac{u_i^{n+1} - u_{i-2}^{n-1}}{2\Delta t} = 0 \tag{2.4}$$

In comparison, an Eulerian approach applied directly on the equation 2.1 would be:

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} + \overline{v}_x \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0 \tag{2.5}$$

Where in this last scheme 2.5 we have the CFL condition to be respected that bring a strong restriction over the Δt .

To better understand the significance of the equation 2.4, consider the following diagram:

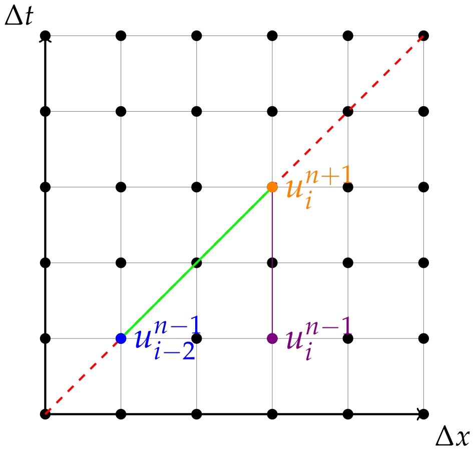


Figure 2.2: Example of a finite difference approach for the Lagrangian derivative. In green there is the semi-Lagrangian approach and in violet the standard Leap-Frog integrator

In this figure, we observe how the semi-Lagrangian approximation made in 2.4 traces the particle along its trajectory (characteristic curve). Hence, the general equation 2.4 translates into:

$$\frac{u_i^{n+1} - u_{i-2\alpha}^{n-1}}{2\Delta t} = 0 \tag{2.6}$$

where α is an important parameter of this scheme, providing information about the position of our particle $2\Delta t$ time steps in the past.

In our straightforward example, alpha is exactly equal to $2\Delta x$, as depicted in the following figure.

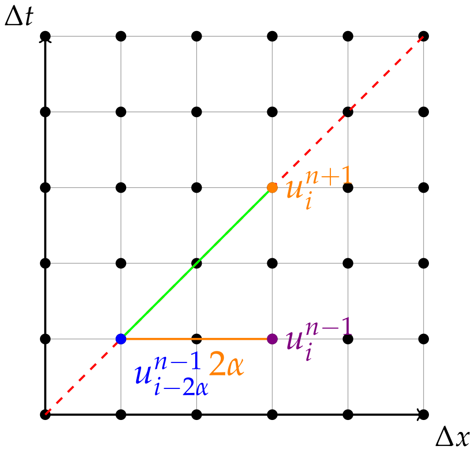


Figure 2.3: Representation of the displacement 2α

Finding α represents a critical step in semi-Lagrangian schemes. In the simplest cases where the characteristic curve is known analytically, a closed formula for α can be found. However, in almost all practical cases, this is not possible. For this reason, iterative methods are used to determine α. Generally, the challenge of finding α is often referred to in the literature as the "starting point" problem. One of the first iterative methods proposed for semi-Lagrangian schemes was introduced by Robert [1]. This iterative method seeks to find the slope at the point $u_{i-\alpha}^n$ by approximating the characteristic to a straight line, achieving an approximation order of $O(\Delta t^2)$.

$$\alpha^{k+1} = \Delta t v(x_i^n - \alpha^k, t^n) \tag{2.7}$$

where v represents the velocity field of the advection. There are two notable cases worth mentioning. In the simpler case, v has a closed-form expression of the form: $v(x,t)$. In this scenario, it is relatively straightforward to use the iterative method 2.7 to find α. The second case arises when v is a function of the form $v(x,t,u(x,t))$. In this situation, to evaluate the velocity, interpolation of the solution at previous steps may be required because the point $x - \alpha^k$ may not be a grid node for which we have a value.

2.2 The Interpolation Problem

In the previous paragraph, we introduced one of the two main steps of the semi-Lagrangian methods. In this section, we will analyze interpolation. We saw how even in the iterative method, the need for interpolation may arise. However, the most important interpolation is used to obtain the solution value at $u_{i-2\alpha}^{n-1}$ as shown in equation 2.6. In fact, if we consider a situation where we have a linear velocity but with a characteristic that does not perfectly align with our grid, we end up to the case described in the following figure.

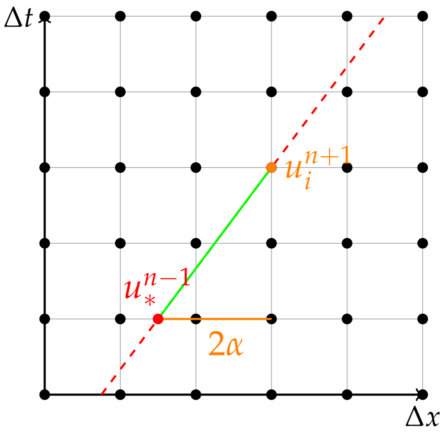


Figure 2.4: Example with a not alligned characteristic

As can be clearly seen from the figure, unlike equation 2.4, we are missing the u_*^{n-1} value.

$$\frac{u_i^{n+1} - u_*^{n-1}}{2\Delta t} = 0 \tag{2.8}$$

As in most cases, the point u_*^{n-1} does not belong to the grid, and a solution is therefore not readily available. Interpolation methods are used precisely for obtaining the value of u_*^{n-1}.

2.3 The Harold Ritchie Paper

Harold Ritchie in his work[7] is the first to introduce the concept of "Non-Interpolating Semi-Lagrangian" schemes. His paper represents one of the main inspirations for this thesis, and for this reason, it will be analyzed in detail. Specifically, in this section, the demonstrations, practical meanings, and their implementation in the code through examples will be analyzed.

2.3.1 Introduction

As seen in the previous chapter, semi-Lagrangian methods are methods based on the discretization of a particle's trajectory. In particular, in semi-Lagrangian methods, there are always two phases: a propagation phase and an interpolation phase. The propagation phase is responsible for calculating the particle's trajectory, while the interpolation phase is responsible for calculating the solution value at the starting point. This approach has primary application in advection problems, where the solution is transported by a velocity field. For clarity, the prototype of the hyperbolic PDE, the advection equation, will be used as an example in the following subsections.

The presence of interpolation within these schemes can lead to damping effects and generally involves one of the highest computational costs. Therefore, the possibility of eliminating interpolation within these schemes is very appealing.

2.3.2 The Non-Interpolating Semi-Lagrangian Scheme

Starting from equation 2.2 and adding the source $g(x,t)$, we get the starting point of the Harold Ritchie paper [7].

$$\frac{du}{dt} = g(x,t) \tag{2.9}$$

that can be written as seen in 2.3 as:

$$\frac{\partial u}{\partial t} + \frac{dx}{dt}\frac{\partial u}{\partial x} = g(x,t) \tag{2.10}$$

This paper introduces a novel approach with the NISL methods. The primary concept behind these methods involves decomposing the advection velocity into two components. The first component is designed to transport our particle from the point x_{i-p} to the point x_i. The second component represents the residual, which can be handled either through finite differences, spectral methods or more in general with an Eulerian approach.

We begin by decomposing the advection velocity $dx/dt = v(x,t)$ into two components:

$$v(x,t) = v(x,t) + \frac{p\Delta x}{2\Delta t} - \frac{p\Delta x}{2\Delta t} \tag{2.11}$$

Now, we can rewrite equation 2.10 as:

$$\frac{\partial u}{\partial t} + \frac{p\Delta x}{2\Delta t}\frac{\partial u}{\partial x} = (\frac{p\Delta x}{2\Delta t} - v(x,t))\frac{\partial u}{\partial x} + g(x,t) \tag{2.12}$$

In this new equation format, we can observe that the left-hand side Left-Hand Side (LHS) of our equation represents our Lagrangian derivative, while the Right-Hand Side (RHS) depicts a residual to be evaluated.

$$\underbrace{\frac{Du}{Dt}}_{\text{Lagrangian Derivative}} = (\frac{p\Delta x}{2\Delta t} - v(x,t))\frac{\partial u}{\partial x} + g(x,t) \tag{2.13}$$

Proceeding with the semi-Lagrangian approach and employing the position and time on the grid, we arrive at the following equation:

$$\frac{u_i^{n+1} - u_{i-p}^{n-1}}{2\Delta t} = (\frac{p\Delta x}{2\Delta t} - v^n_{i-\frac{p}{2}})\frac{\partial u^n_{i-\frac{p}{2}}}{\partial x} + g^n_{i-\frac{p}{2}} \tag{2.14}$$

To better understand this equation, let's consider the following diagram:

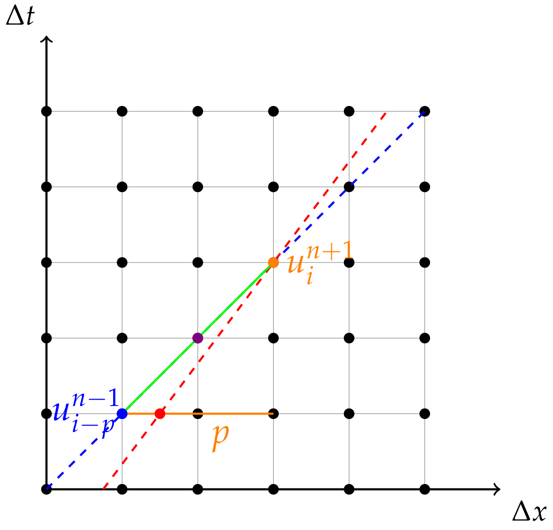


Figure 2.5: Example of NISL scheme with the new characteristic after the decomposition shown in blue, and the original in red.

In the diagram, unlike in 2.2, our scheme does not follow the characteristic of the original PDE, but rather a new characteristic defined by $\frac{2\Delta t}{p\Delta x}$ (shown in blue in the diagram 2.5). This allows us to derive an equation that does not require interpolation to be solved. Indeed, the characteristic feature of this equation is that p is computed as $[2\alpha]$ where the square brackets denote rounding to the nearest integer. The RHS, on the other hand, is an evaluation of the solution at the midpoint $u^n_{i-\frac{p}{2}}$ (shown in purple in the diagram 2.5); this part can be evaluated with any classical Eulerian scheme.

For instance, using a finite difference approximation with a central scheme for the RHS of equation 2.14, we the following two options.

In the case of even p:

$$\frac{u_i^{n+1} - u_{i-p}^{n-1}}{2\Delta t} = (\frac{p\Delta x}{2\Delta t} - v^n_{i-\frac{p}{2}})\frac{u^n_{i+1+\frac{p}{2}} - u^n_{i-1+\frac{p}{2}}}{2\Delta x} + g^n_{i-\frac{p}{2}} \tag{2.15}$$

In the case of odd p:

$$\frac{u_i^{n+1} - u_{i-p}^{n-1}}{2\Delta t} = (\frac{p\Delta x}{2\Delta t} - v^n_{i-\frac{p}{2}})\frac{u^n_{i-\frac{p-1}{2}} - u^n_{i-\frac{p+1}{2}}}{2\Delta x} + g^n_{i-\frac{p}{2}} \tag{2.16}$$

Here, $v^n_{i-\frac{p}{2}}$ can be evaluated explicitly as $v(x_i - \frac{p\Delta x}{2}, n\Delta t)$ or it can be evaluated with a finite difference approach in the case of an implicit form of the type $v(u(x,t),t)$.In general, in the case of p being odd, the scheme can easily become more complex; the example in 2.16 is a simplification of the problem. In fact, to identify the middle point accurately, it is important to respect the order of approximation of the spatial discretization, which may lead to the adoption of an interpolation approach. However, using explicit interpolation to evaluate the point may devalue the present work. Without better ideas, this approach, which increases the order of spatial approximation and avoids interpolation, would be very challenging using the Ritchie method.

2.3.3 The p displacement

One of the most interesting parts of this scheme, and one of the main differences from the standard SL scheme, is the definition of the displacement. While in the standard SL we define a real value of 2α as displacement, for the NISL scheme we introduce the integer value p, which is defined based on the 2α displacement as $p = \lfloor 2\alpha \rceil$. It is important to note that this is not a fixed formula; the definition of p is strictly related to the time integrator used. In the previous sub-chapters, we have used a Leap-Frog like time-integrator, which is a known 3-steps[1] method. However, in general, the displacement depends on the number of steps in time. This observation, which seems straightforward, is very important in the implementation of these schemes and may easily lead to errors in the code. In order to implement more advanced time integrators with a NISL scheme, the first step is to properly define the p displacement. Being smart in the definition of the p displacement can eliminate the necessity of using the formula 2.16 or, more generally, to obtain an odd p. Another important point is that, unlike the displacement α, being p approximated to the nearest integer, it is sufficient to have an approximation of α to the first decimal places. The leading error will be created by the Eulerian step.

[1] Where the 3-steps refers to the number of points used in the time integration in this case u^{i-1}, u^i, u^{i+1}

2.4 Rančić's Study

2.4.1 Introduction

The paper presented by Rančić and Sindjić [8] employs a novel approach to SL methods with the intent to eliminate the need for interpolation. The primary insight behind this approach is that rather than decomposing the advection speed into two components, the method introduces what is termed an 'Eulerian step'. The main idea of this method is that at certain instants in time, the characteristics intersect the grid points. Therefore, to obtain the solution u_i^{n+1}, we can advance with a time step Δt_* using a standard Eulerian scheme.

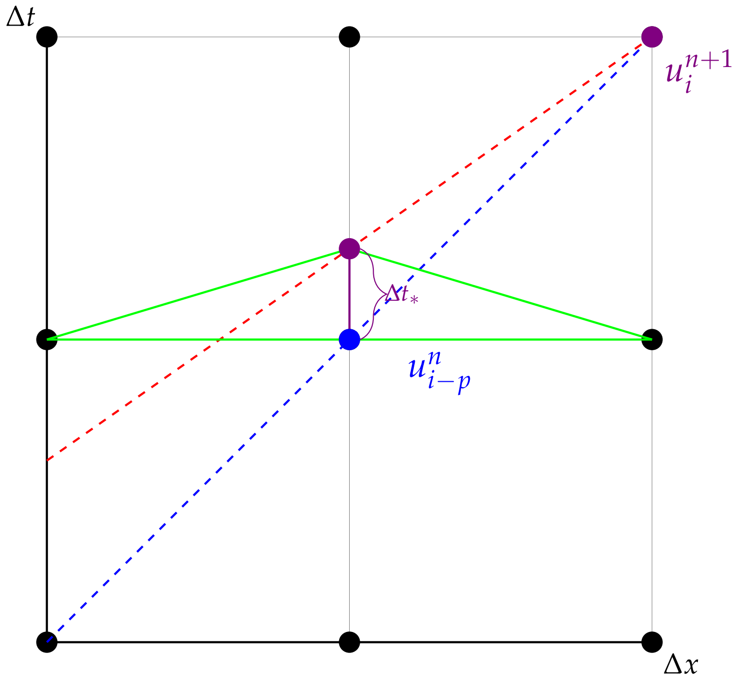


Figure 2.6: Rančić's NISL scheme incorporates the Eulerian step, where the violet dots represent the same values, as they lie on the same characteristics depicted in red.

Upon examining the diagram 2.6, it is clear that the construction of the method bears a resemblance to the approach proposed by Ritchie. The crucial difference, however, is that instead of integrating a residual on the RHS, we initiate an Eulerian step from the point u_{i-p}^n using a timestep of Δt_*. This approach facilitates the development of 2-step temporal methods.

For example, well-known methods like Lax-Wendroff can be effectively implemented using this technique:

$$u_i^{n+1} = u_{i-p}^n - \frac{\Delta t_*}{2\Delta x}(u_{i+1-p}^n - u_{i-1-p}^n) + \frac{\Delta t_*^2}{2\Delta x^2}(u_{i+1-p}^n - 2u_{i-p}^n + u_{i-1-p}^n) \tag{2.17}$$

To solve this equation, it is essential to ascertain the value of Δt_*. However, identifying Δt_* is straightforward once p is established. Through basic geometric constructs, it can be inferred that:

$$\Delta t_* = \Delta t(1 - \frac{p\Delta x}{v\Delta t}). \tag{2.18}$$

In the context of non-constant speeds, when applying the approximation used in Ritchie's iterative approach, as seen in the equation 2.7, the previously mentioned formula remains applicable. This iterative method simplifies the characteristic to a linear form within this interval, thereby enabling the derivation of Δt_* once α is determined.

For approximations of a higher order, the aforementioned formula becomes more complex. The main advantages of this formula include the opportunity to use a 2-step time integrator, which eliminates the need for approximations at the midpoint in the case of odd p. It also obviates the necessity of starting the simulation with a different solver.

2.4.2 Is an interpolating scheme?

Upon reviewing the literature, particularly starting from the reviews by Staniforth [2], we find that the scheme in 2.17 is algebraically equivalent to a standard SL scheme using Lagrangian interpolation, as demonstrated by Dietachmayer in the comment [9]. Even though it is possible to demonstrate that it is an interpolating scheme, it is important to note that Rančić's intuition is still very interesting and useful for this thesis. The possibility of writing a NISL method using a 2-step integrator is a very important feature for making these schemes usable in a broader context than just numerical experiments. In particular, it is important to note how the stencils of FD)methods are theoretically close to interpolating polynomials, and in fact, many of the FD methods are derived from specific interpolating polynomials. Therefore, it is possible and plausible that with some combinations of FD schemes and some simple time integrators, it can be demonstrated that the final schemes are algebraically equal to interpolating schemes. However, the most important characteristic that will be highlighted in the thesis is that this formulation opens many opportunities for combinations of well-known Eulerian schemes with semi-Lagrangian corrections, leading to very robust numerical methods with reduced computational cost.

2.5 The Relation between Rančić's and Ritchie Approach

To fully appreciate these two approaches and understand how they, despite differing in intuition, represent a very similar methodology, it is possible to perform some algebraic steps.

To do this, we can start by using a formulation by Ritchie with a 2-step integrator.

$$\frac{\partial u}{\partial t} + v(x,t)\frac{\partial u}{\partial x} = 0 \tag{2.19}$$

By defining :

$$v(x,t) = v(x,t) + \frac{p\Delta x}{\Delta t} - \frac{p\Delta x}{\Delta t} \tag{2.20}$$

One gets:

$$\frac{\partial u}{\partial t} + \frac{p\Delta x}{\Delta t}\frac{\partial u}{\partial x} = (\frac{p\Delta x}{\Delta t} - v(x,t))\frac{\partial u}{\partial x} \tag{2.21}$$

$$\underbrace{\frac{Du}{Dt}}_{\text{Lagrangian Derivative}} = (\frac{p\Delta x}{\Delta t} - v(x,t))\frac{\partial u}{\partial x} \tag{2.22}$$

$$\frac{Du}{Dt} = -v(x,t)(1 - \frac{p\Delta x}{v(x,t)\Delta t})\frac{\partial u}{\partial x} \tag{2.23}$$

By using the definition of Rančić's for Δt_* (see eq. 2.18) one gets:

$$\frac{Du}{Dt} = -v(x,t)\frac{\Delta t_*}{\Delta t}\frac{\partial u}{\partial x} \tag{2.24}$$

From this formulation, it is possible to use Lax-Wendroff scheme to obtain the same scheme proposed in equation 2.17. In this way, we have demonstrated that using Ritchie's intuition, it is possible to derive Rančić's scheme. However, it is important to specify that Ritchie's work is valid for 3-step integrators like Leap-Frog. Therefore, applying the same approach to a 2-step integrator goes beyond the validity of Ritchie's demonstration that the scheme is unconditionally stable under every CFL condition. This means that applying 2-step numerical schemes to the formulation 2.24 would require proving that the CFL condition is always respected. However, in our case, we know it is respected thanks to the proof proposed by Rančić in [8].

3 Advection Model

The first model investigated in the present semi-Lagrangian methods work is one of the simplest and most extensively studied models in the theory of PDEs: the Advection model. This model serves as a starting point for many numerical scheme studies, as its simplicity makes it well-suited for validating schemes, especially given the possibility of using analytical solutions. The Advection model represents the transport of a scalar quantity u within a fluid and can be expressed as:

$$\frac{\partial u}{\partial t} + \boldsymbol{v} \cdot \nabla u = g, \tag{3.1}$$

where

- $u(\boldsymbol{x}, t)$ is the scalar quantity dependent on the position vector $\boldsymbol{x}$ and time t
- $\boldsymbol{v}(\boldsymbol{x}, t)$ is the non-constant velocity field
- ∇u represents the gradient of u
- $g(\boldsymbol{x}, t)$ is a generic source term

$\boldsymbol{v} \cdot \nabla$ It is also called the advection operator and is expressed as:

$$\boldsymbol{v} \cdot \nabla = v_x \frac{\partial}{\partial x} + v_y \frac{\partial}{\partial y} + v_z \frac{\partial}{\partial z} \tag{3.2}$$

3.1 Advection with Constant Velocity and 0 Source Term

In this section, we will study the application of semi-Lagrangian methods for advection models with constant velocity and without sources. Starting from equation 3.1, we obtain:

$$\frac{\partial u}{\partial t} + \overline{\boldsymbol{v}} \cdot \nabla u = 0, \tag{3.3}$$

where $\overline{\boldsymbol{v}}$ is the constant velocity that does not depend on either position or time.

3.1.1 1D Model Discretization: Lax Wendroff

Starting from equation 3.3, we can begin to write the first model to test our numerical scheme. Specifically, in this subsection, we will focus on one dimension.

$$\frac{\partial u(x,t)}{\partial t} + \overline{v}_x \frac{\partial u(x,t)}{\partial x} = 0, \tag{3.4}$$

The process of discretizing the given PDE model is relatively straightforward and can be effectively achieved through the implementation of a second-order method such as the Lax-Wendroff scheme. In order to do so, we adopt the approach delineated in [10], which entails commencing with a second-order Taylor expansion concerning the time variable.

$$u(x, t + \Delta t) = u(x,t) + \Delta t \frac{\partial u}{\partial t} + \frac{1}{2} \Delta t^2 \frac{\partial^2 u}{\partial t^2} + O(\Delta t^3) \tag{3.5}$$

Using the definition of our PDE 3.4 one gets:

$$\frac{\partial u}{\partial t} = -\overline{v}_x \frac{\partial u(x,t)}{\partial x} \tag{3.6}$$

$$\frac{\partial}{\partial t}\frac{\partial u}{\partial t} = -\frac{\partial}{\partial t}\overline{v}_x \frac{\partial u(x,t)}{\partial x} \tag{3.7}$$

In this step we have that $\overline{v}_x$ is a constant advection velocity so we can move it outside and we can switch the order of the partial derivatives getting:

$$\frac{\partial^2 u}{\partial t^2} = -\overline{v}_x \frac{\partial}{\partial x}\frac{\partial u(x,t)}{\partial t} \tag{3.8}$$

Using again the definition of our PDE we can substitute the time derivative with the spatial derivative and obtain:

$$\frac{\partial^2 u}{\partial t^2} = -\overline{v}_x \frac{\partial}{\partial x}(-\overline{v}_x \frac{\partial u(x,t)}{\partial x}) \tag{3.9}$$

$$\frac{\partial^2 u}{\partial t^2} = +\overline{v}_x^2 \frac{\partial^2 u(x,t)}{\partial x^2} \tag{3.10}$$

Replacing 3.6 and 3.10 in 3.5 we get a new formulation with only derivative in the space.

$$u(x, t + \Delta t) = u(x,t) + \Delta t\ \overline{v}_x \frac{\partial u(x,t)}{\partial x} + \frac{1}{2}\Delta t^2\ \overline{v}_x^2 \frac{\partial^2 u(x,t)}{\partial x^2} + O(\Delta t^3) \tag{3.11}$$

To obtain a fully discrete model, we can replace the spatial partial derivatives with a simple central difference approximation. By applying Rančić's approach, we obtain the formula presented in 2.17 with the constant velocity $\overline{v}_x$.

$$u_i^{n+1} = u_{i-p}^n - \overline{v}_x \frac{\Delta t_*}{2\Delta x}(u_{i+1-p}^n - u_{i-1-p}^n) + \overline{v}_x^2 \frac{\Delta t_*^2}{2\Delta x^2}(u_{i+1-p}^n - 2u_{i-p}^n + u_{i-1-p}^n) \tag{3.12}$$

Where in this case $p = \lfloor \alpha \rceil$.

3.1.2 1D Model Discretization: Leapfrog

The Leapfrog discretization is straightforward and follows exactly the approach proposed by Ritchie.

$$\frac{\partial u(x,t)}{\partial t} + \overline{v}_x \frac{\partial u(x,t)}{\partial x} = 0 \tag{3.13}$$

$$\frac{Du}{Dt} = (\frac{p\Delta x}{2\Delta t} - \overline{v}_x)\frac{\partial u}{\partial x} \tag{3.14}$$

By using a central difference approximation:

$$u_i^{n+1} = u_{i-p}^{n-1} + \Delta t(\frac{p\Delta x}{2\Delta t} - \overline{v}_x))\frac{u_{i-\frac{p}{2}+1}^n - u_{i-\frac{p}{2}-1}^n}{\Delta x} \tag{3.15}$$

where p is defined in 2.3.3.

3.1.3 Initial Value Problem

We define the Initial Value Problem (IVP) as:

$$\frac{\partial u(x,t)}{\partial t} + \overline{v}_x \frac{\partial u(x,t)}{\partial x} = 0, \quad (x,t) \in \Omega = [0,1] \times [0,T]; \tag{3.16}$$

$$\text{Initial Condition:} \quad u(x,0) = sin(2\pi x) \tag{3.17}$$

$$\text{Boundary Conditions:} \quad u(0,t) = u(1,t). \tag{3.18}$$

We can use the solution obtained in 1.2.1 in order to evaluate the error.

3.1.4 Simulation: Respected CFL Condition

In this section the first simulation test, which is also one of the most relevant, is performed. Indeed, even though it is a very simple model, the presence of an analytical solution and ease of implementation allow us to conduct a more in-depth study, not only on the convergence of the error but also on performance. In this section, a third solver will be introduced, which is based on standard SL methods with quadratic spline interpolation.

This first simulation in table 3.1 starts with a respected CFL condition to verify the accuracy of the solver.

Parameter	Value	Description
Model	$\partial_t u + v\partial_x u = 0$	1D Advection Model with constant velocity
Boundary condition	$u(0,t) = u(1,t)$	Periodic
Initial conditions	$u(x,0) = sin(2\pi x)$	Sinusoidal wave
Δt	0.0008	Starting timestep for time discretization
Δx	1/512	Starting spatial discretization
CFL	0.4096	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	$[0,1]$	Spatial Domain
t	$[0,0.8]$	Time Domain
Solver 1	SLSQ	Semi Lagrangian with quadratic splines interpolation
Solver 2	LW-NISL	Lax-Wendroff non interpolating semi lagrangian
Solver 3	LF-NISL	Leap-Frog non interpolating semi lagrangian

Table 3.1: Table with simulation parameters for Advection model with respected CFL

It is important to note how the NISL solvers in the low CFL regime, that is when $p = 0$, behave exactly like the associated Eulerian solvers without the need for any modifications. When $p \neq 0$, the semi-Lagrangian correction on the trajectories begins to take effect. This characteristic is very important because, during implementation, it allows for the separation of the Eulerian solver and the semi-Lagrangian correction. This separation facilitates more straightforward testing of both the code and the results.

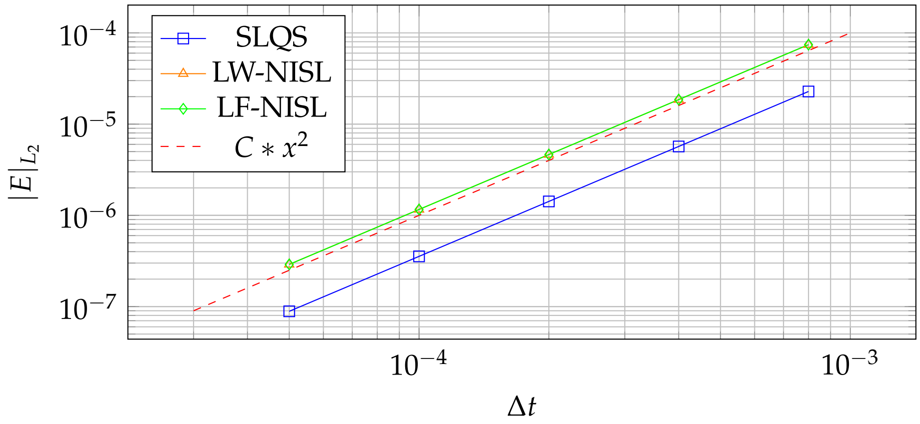


Figure 3.1: Convergence rate comparison with respected CFL condition

The results in Fig-3.1 confirm that the solvers adhere to the theoretical convergence, thus following the asymptotic behavior of x^2. Particularly in this case, the classic solver has a lower error, but it is still comparable.

3.1.5 Simulation: Not Respected CFL Condition

In the second simulation, we can observe the elimination of the CFL restriction (table 5.1). In this simple model, as there are very straightforward characteristic curves and no non-linearities in the model, we can push the CFL condition to very high values. This experiment allows us to verify how the solvers behave at high CFL values.

Parameter Name	Value	Description
Model	$\partial_t u + \partial_x u = 0$	1D Advection model with constant velocity
Boundary Conditions	$u(0,t) = u(1,t)$	Periodic boundary conditions
Initial Conditions	$u(x,0) = \sin(2\pi x)$	Sinusoidal wave
Δt	0.08	Starting timestep for time discretization
Δx	1/512	Starting spatial discretization
CFL	40.96	Courant–Friedrichs–Lewy (CFL) $CFL = \frac{\Delta t}{\Delta x}$
Ω	[0, 1]	Spatial domain
t	[0, 8]	Time Domain
Solver 1	SLSQ	Semi-Lagrangian with Quadratic Splines interpolation
Solver 2	LW-NISL	Lax-Wendroff non-interpolating semi-Lagrangian
Solver 3	LF-NISL	Leap-Frog non-interpolating semi-Lagrangian

Table 3.2: Table with simulation parameters for Advection model with not respected CFL

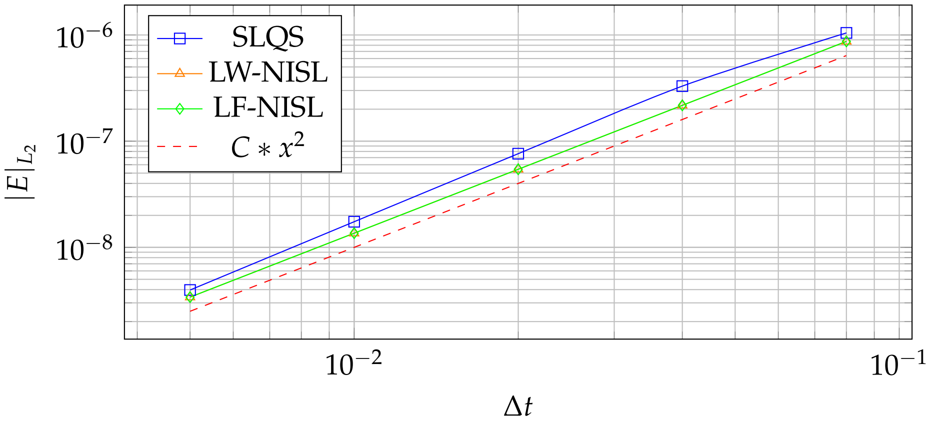


Figure 3.2: Convergence rate comparison with not respected CFL condition

The convergence results in Fig-3.2 for this simple experiment are quite interesting. Indeed, all solvers asymptotically follow their theoretical convergence, but standard methods seem to have more difficulty converging in cases of larger Δt. On the other hand, methods generally perform better at larger Δt. This tendency has been noted in other simulations and is therefore brought to the reader's attention.

It is generally important to consider that having high Δt values can be problematic as, in physical experiments, they can invalidate the physics of the simulation. This inability to capture phenomena occurring at higher frequencies than what our simulation can detect will be explored more thoroughly in the case of plasma physics models.

3.1.6 Simulation: Performance Analysis

As anticipated, these simulations also included an analysis of the solvers' performance. Performance analyses of solvers are generally very complex and depend on many external factors, not only on the numerical schemes. In particular, they can depend on the specific implementation chosen, and the results may vary based on the hardware and active processes in the operating system. In general, all experiments were conducted with high priority and by removing other processes on the same hardware.

Parameter Name	Value	Description
Model	$\partial_t u + \partial_x u = 0$	1D Advection Model with constant velocity
Boundary conditions	$u(0,t) = u(1,t)$	Periodic Boundary Conditions
Initial conditions	$u(x,0) = sin(2\pi x)$	Sinusoidal wave
Δt	0.1	Fixed Timestep for time discretization
N_x	512	Starting Spatial Discretization
CFL	51.2	Starting Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	[0, 1]	Spatial Domain
t	[0, 500]	Time Domain
Solver 1	SLSQ	Semi Lagrangian with Quadratic Splines interpolation
Solver 2	LW-NISL	Lax-Wendroff non interpolating semi lagrangian
Solver 3	LF-NISL	Leap-Frog non interpolating semi lagrangian

Table 3.3: Table with simulation parameters for Advection model with not respected CFL Performance Analysis

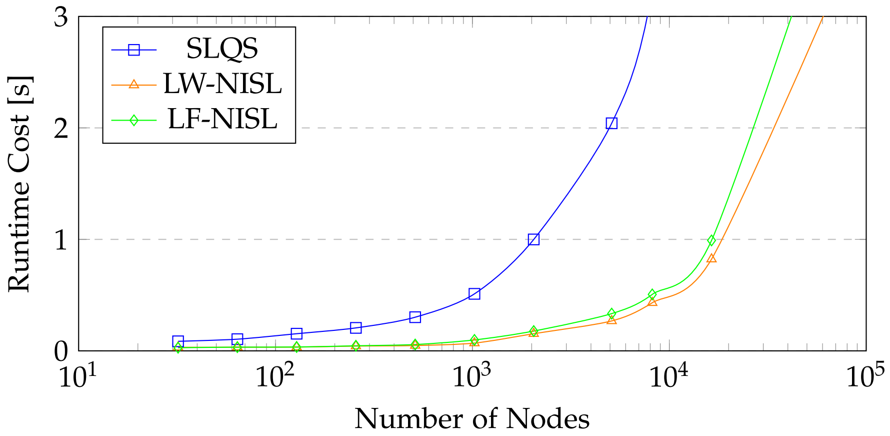


Figure 3.3: Performance Comparison over increasing number of nodes

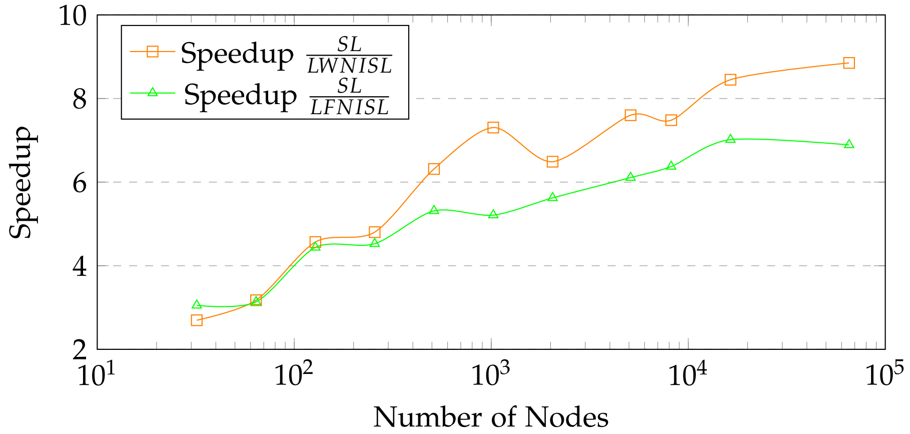


Figure 3.4: Runtime speedup

The methods are found to be more efficient as the number of nodes increases compared to semi-Lagrangian with Quadratic Splines (SLQS) method Fig-3.3, Fig-3.4. An important observation to make is that the SLQS numerical method is based on spline interpolation, utilizing a global interpolation matrix, whereas the NISL methods employ FD schemes that have a computational cost comparable to local interpolations, such as those with Lagrange polynomials. Therefore, one of the findings of this experiment is that NISL methods are more efficient than interpolative methods with global interpolations and are on the same level in performance with methods using local interpolations.This result aligns with the theory, as finite difference stencils are often based on local interpolating polynomials.

3.2 Advection with Constant Velocity and Source Term

Starting from equation 3.1, we can define a new model with a constant $\overline{v}$ and a source term g.

$$\frac{\partial u}{\partial t} + \overline{v} \cdot \nabla u = g, \tag{3.19}$$

3.2.1 1D Model and Analytical Solution

Beginning with equation 3.19, we define the 1D model incorporating a source term $g(x,t)$, which depends on x and t. It is important to note that this term does not depend on the solution u.

$$\frac{\partial u(x,t)}{\partial t} + \overline{v}_x \frac{\partial u(x,t)}{\partial x} = g(x,t), \tag{3.20}$$

The presence of the source implies that the solution over the characteristic curves will not be constant, as it will also include a term given by the source that evolves over time and space. Following the same approach presented in section 1.2, we can write the ODE associated with the characteristics of this PDE.

$$\frac{d\xi(t;x_0)}{dt} = \overline{v}_x, \tag{3.21}$$

with the same solution as reported in equation 1.22. However in this case the solution u over the characteristic curve is not constant and is given by the following equation:

$$\frac{du(\xi(t))}{dt} = g(\xi(t)) \tag{3.22}$$

In this case this ODE is solvable by direct integration and the general solution is given by:

$$u(\xi(t;x_0),t) = u(\xi(t;x_0),0) + \int_0^t g(\xi(\tau;x_0),\tau)d\tau \tag{3.23}$$

Where we have used the initial condition $u(x,0) = u(x_0,0)$.

If we now use the solution of 3.21 to substitute $\xi(t, x\_0)$ we obtain our final solution:

$$u(\boldsymbol{x},t) = u(\boldsymbol{x}_0 + \overline{\boldsymbol{v}}_x(t-t_0),t_0) + \int_0^t g(\boldsymbol{x}_0 + \overline{\boldsymbol{v}}_x(\tau - t_0),\tau)d\tau \tag{3.24}$$

3.2.2 1D Model Discretization: Lax-Wendroff

In order to discretize the new model proposed in 3.20 we start as in the previous chapter 3.1.1 with a simple Taylor expansion.

$$u(x,t+\Delta t) = u(x,t) + \Delta t \frac{\partial u}{\partial t} + \frac{1}{2}\Delta t^2 \frac{\partial^2 u}{\partial t^2} + O(\Delta t^3) \tag{3.25}$$

We define the time derivatives with the definition of our PDE 3.20:

$$\frac{\partial u}{\partial t} = -\overline{v}_x \frac{\partial u(x,t)}{\partial x} + g(x,t) \tag{3.26}$$

$$\frac{\partial}{\partial t}\frac{\partial u}{\partial t} = -\frac{\partial}{\partial t}\overline{v}_x \frac{\partial u(x,t)}{\partial x} + \frac{\partial}{\partial t} g(x,t) \tag{3.27}$$

In this step we have that $\overline{v}_x$ is a constant advection velocity so we can move it outside and we can switch the order of the partial derivatives getting:

$$\frac{\partial^2 u}{\partial t^2} = -\overline{v}_x \frac{\partial}{\partial x}\frac{\partial u(x,t)}{\partial t} + \frac{\partial}{\partial t} g(x,t) \tag{3.28}$$

Using again the definition of our pde we can substitute the time derivative with the spatial derivative and obtain:

$$\frac{\partial^2 u}{\partial t^2} = -\overline{v}_x \frac{\partial}{\partial x}(-\overline{v}_x \frac{\partial u(x,t)}{\partial x} + g(x,t)) + \frac{\partial}{\partial t} g(x,t) \tag{3.29}$$

$$\frac{\partial^2 u}{\partial t^2} = +\overline{v}_x^2 \frac{\partial^2 u(x,t)}{\partial x^2} - \overline{v}_x \frac{\partial}{\partial x} g(x,t) + \frac{\partial}{\partial t} g(x,t) \tag{3.30}$$

If we use these new definition inside our taylor expansion 3.25 we obtain the following formulation:

$$\begin{aligned} u(x,t+\Delta t) = u(x,t) - \Delta t\, \overline{v}_x \frac{\partial u(x,t)}{\partial x} + \Delta t g(x,t) \\ + \frac{1}{2}\Delta t^2\, \overline{v}_x^2 \frac{\partial^2 u(x,t)}{\partial x^2} - \frac{1}{2}\Delta t^2 \overline{v}_x \frac{\partial}{\partial x} g(x,t) \\ + \frac{1}{2}\Delta t^2 \frac{\partial}{\partial t} g(x,t) + O(\Delta t^3) \end{aligned} \tag{3.31}$$

Where we can use a central difference to discretize in the space domain and apply the NISL approach inserting the p displacement and the Δt_* using the Rančić's approach.

$$\begin{aligned} u_i^{n+1} = u_{i-p}^n - \overline{v}_x \frac{\Delta t_*}{2\Delta x}(u_{i+1-p}^n - u_{i-1-p}^n) + \overline{v}_x^2 \frac{\Delta t_*^2}{2\Delta x^2}(u_{i+1-p}^n - 2u_{i-p}^n + u_{i-1-p}^n) \\ + \Delta t_* g(x - p\Delta x, t) - \frac{1}{2}\Delta t_*{}^2 \overline{v}_x \frac{\partial}{\partial x} g(x - p\Delta x, t) + \frac{1}{2}\Delta t_*{}^2 \frac{\partial}{\partial t} g(x - p\Delta x, t) \end{aligned} \tag{3.32}$$

Where, in blue, we denote the new terms introduced by the source. It is important to note that if $g(x,t)$ is explicitly given, we can compute its derivative explicitly. However, in cases where the source term depends on the solution, a FD discretization must be employed to obtain the derivative.

3.2.3 1D Model Discretization: Leapfrog

Unlike the Lax-Wendroff scheme, the discretization with Leapfrog is significantly simpler, and we will have only one new term created by the source.

$$\frac{\partial u(x,t)}{\partial t} + \overline{v}_x \frac{\partial u(x,t)}{\partial x} = g(x,t) \tag{3.33}$$

$$\frac{Du}{Dt} = (\frac{p\Delta x}{2\Delta t} - \overline{v}_x)\frac{\partial u}{\partial x} + g(x,t) \tag{3.34}$$

By using a central difference approximation:

$$u_i^{n+1} = u_{i-p}^{n-1} + \Delta t(\frac{p\Delta x}{2\Delta t} - \overline{v}_x))\frac{u_{i-\frac{p}{2}+1}^n - u_{i-\frac{p}{2}-1}^n}{\Delta x} + 2\Delta t g(x - \frac{p}{2}\Delta x, t) \tag{3.35}$$

where p is defined in 2.3.3.

3.2.4 Initial Value Problem

We define the IVP as:

$$\frac{\partial u(x,t)}{\partial t} + \overline{v}_x \frac{\partial u(x,t)}{\partial x} = cos(2\pi x + t), \quad (x,t) \in \Omega = [0,1] \times [0,T]; \tag{3.36}$$

$$\text{Initial Condition:} \quad u(x,0) = sin(2\pi x) \tag{3.37}$$

$$\text{Boundary Conditions:} \quad u(0,t) = u(1,t). \tag{3.38}$$

We can use the solution obtained in 3.24 in order to evaluate the error.

3.2.5 Model Error Analysis

For this model, only one experiment will be presented to prove the convergence of the numerical methods table 3.4.

Parameter Name	Value	Description
Model	$\partial_t u + \partial_x u = cos(2\pi x + t)$	1D Advection model with source term
Boundary Conditions	$u(0,t) = u(1,t)$	Periodic boundary conditions
Initial Conditions	$u(x,0) = \sin(2\pi x)$	Sinusoidal wave
Δt	0.04	Starting timestep for time discretization
Δx	0.02	Starting spatial discretization
CFL	2.0	Courant–Friedrichs–Lewy (CFL) $CFL = \frac{\Delta t}{\Delta x}$
Ω	[0, 1]	Spatial domain
t	[0, 8]	Time Domain
Solver 1	LW-NISL	Lax-Wendroff non-interpolating semi-Lagrangian
Solver 2	LF-NISL	Leap-Frog non-interpolating semi-Lagrangian

Table 3.4: Table with simulation parameters for Advection model with not respected CFL

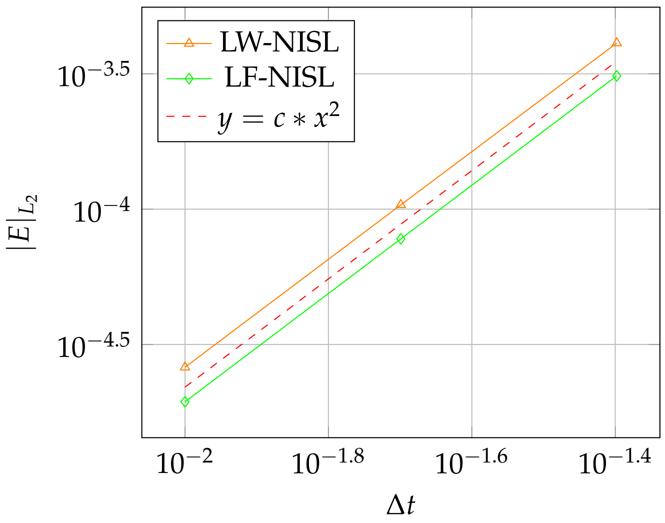


Figure 3.5: Convergence rate comparison with not respected CFL condition

Overall, both solvers converge properly with this simple model see Fig-3.5. However, it is important to note the effort required to obtain and implement the Lax-Wendroff scheme in the code, which makes the Leapfrog method more convenient and effective for this problem.

3.3 Advanced solvers

3.3.1 Runge-Kutta Time Integrators

In this section, we introduce an intriguing alternative for time integration compared to the methods previously discussed: the use of Runge-Kutta methods within the NISL framework.

The central concept is that, following the decomposition of velocity as per Ritchie's approach (see eq. 2.13), this scheme can generally be approached as a purely Eulerian scheme. This is done by applying Rančić's intuition, which involves simply executing an Eulerian step. To illustrate this, let's begin by writing the Butcher tableau for a basic RK2 integrator.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

We can considere the advection model for instance.

$$\frac{\partial f}{\partial t} = -v\frac{\partial f}{\partial x} \tag{3.39}$$

In this scenario, we will execute a single step of length Δt in time. As a result, our decomposition of the advection velocity, in accordance with Ritchie's method, is delineated as follows:

$$\frac{Df}{Dt} = -(v - \frac{p\Delta v}{\Delta t})\frac{\partial f}{\partial x} \tag{3.40}$$

With our equation 3.40 now in SL form, we proceed by applying the trajectory correction using the parameter p. It is crucial to implement the p correction exclusively in the first stage. This approach enables us to evolve points from the base of the characteristic curve, obviating the need to evaluate intermediate points, as was necessary in 2.16. Consequently, the RK2 algorithm is modified to:

1. $f^* = f^n_{(i-p)} - \frac{\Delta t}{2}\left(v^n_i - \frac{p\Delta x}{\Delta t}\right)\frac{\partial f^n_{(i-p)}}{\partial x}$
2. $f^{n+1} = f^n_{(i-p^*)} - \frac{\Delta t}{2}\left(v^* - \frac{p^*\Delta x}{\Delta t}\right)\frac{\partial f^*}{\partial x}$

In this context, v^* represents the electric field updated with information at the midpoint Δt^*. It is important to note that the p correction is also dependent on the velocity, necessitating its recalculation in response to changes. A notably simple yet effective enhancement of this scheme is provided by the TVD (Total Variation Diminishing) method, as proposed in [11]. This adaptation leads to the following form for the second stage:

$$f^{n+1} = \frac{1}{2}f^n_{(i-p^*)} + \frac{1}{2}f^* - \frac{\Delta t}{2}\left(v^* - \frac{p^*\Delta x}{\Delta t}\right)\frac{\partial f^*}{\partial v} \tag{3.41}$$

.

This RK formulation is especially beneficial in the context of splitting operators. The Leapfrog methods are known to be affected by time splitting, as confirmed in [12]. Implementing multistep methods with time-splitting operators is not always straightforward, as it is important to synchronize the timesteps between these two operators. The Lax-Wendroff method, when applied to complex models, increases the complexity of the solver significantly.

In the forthcoming simulation results, the NISL-RK4 method will be utilized, guided by the following Butcher Tableau:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

3.3.2 Weighted Essentially Non-Oscillatory (WENO)

So far, we have only explored very simple discretizations over the spatial domain, using solely second-order central differences. In this section, a brief overview of Weighted Essentially Non-Oscillatory (WENO) schemes is presented, along with their application in the NISL framework. WENO schemes were first introduced by Liu, Osher, and Chan in 1994 [13] in a solver based on Finite Volume. Specifically, for the implementation of WENO in this thesis, the approach proposed by Jiang and Shu in [14] will be used, which utilized a FD framework, resulting in a highly efficient solver even for multi-dimensional problems. A comprehensive review is offered by Shu himself in [15], where he explains the relationship between WENO solvers based on Finite Volume and those based on finite differences.

The Weighted Essentially Non-Oscillatory (WENO) schemes are highly effective for shock-capturing simulations compared to standard FD. These schemes adeptly address the challenges posed by very steep slopes Fig-3.6 . Fundamentally, they operate by blending three stencils in a convex combination, guided by smoothness indicators. This method's effectiveness in handling steep gradients makes it an invaluable tool in the arsenal of numerical methods for solving complex PDEs.

$$S_1 = \{f_{i-2}, f_{i-1}, f_i\}, \quad S_2 = \{f_{i-1}, f_i, f_{i+1}\}, \quad S_3 = \{f_i, f_{i+1}, f_{i+2}\} \tag{3.42}$$

For each stencil, a polynomial approximation is formulated. For example, in approximating the flux at $f_{i+\frac{1}{2}}$, we consider the following three polynomial options:

$$f^{(1)}_{i+\frac{1}{2}} = \frac{1}{3}f_{i-2} - \frac{7}{6}f_{i-1} + \frac{11}{6}f_i \tag{3.43}$$

$$f^{(2)}_{i+\frac{1}{2}} = -\frac{1}{6}f_{i-1} + \frac{5}{6}f_i + \frac{1}{3}f_{i+1} \tag{3.44}$$

$$f^{(3)}_{i+\frac{1}{2}} = \frac{1}{3}f_i + \frac{5}{6}f_{i+1} - \frac{1}{6}f_{i+2} \tag{3.45}$$

Once these polynomials are obtained, the WENO scheme performs a convex combination of them, based on the smoothness coefficients ω_i.

$$f_{i+\frac{1}{2}} = \omega_1 f^{(1)}_{i+\frac{1}{2}} + \omega_2 f^{(2)}_{i+\frac{1}{2}} + \omega_3 f^{(3)}_{i+\frac{1}{2}} \tag{3.46}$$

In the WENO scheme, the ω coefficients play a crucial role. In scenarios where the solution is completely smooth, the referenced equation evolves into a 5th-order finite difference approximation. However, in regions with shocks, these coefficients adapt the stencil configuration to mitigate oscillations.

Once this flow reconstruction is utilized, we can solve the PDE.

$$\frac{du_i}{dt} + \frac{1}{\Delta x}\left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right) = 0 \tag{3.47}$$

To ensure the stability of the stencil, a flux-splitting is performed to obtain $f(u) = f^+ + f^-$. For the implementation in this thesis, the Lax-Friedrichs method was used:

$$f^{\pm} = f(u) \pm \alpha u \tag{3.48}$$

Implementing WENO methods within the SL framework is quite straightforward. In fact, starting from the velocity splitting as in 3.14, the process can be effectively executed.

$$\partial_t u + v\partial_x u = 0 \tag{3.49}$$

$$\frac{Du}{Dt} = (\frac{p\Delta x}{2\Delta t} - v)\frac{\partial u}{\partial x} \tag{3.50}$$

In cases where v does not explicitly depend on v, we can write it in the conservative form:

$$\frac{Du}{Dt} = f^*(u)_x \tag{3.51}$$

In this way, we have determined that the flux f^* is the residual of advection in the x direction. By discretizing this, we obtain:

$$\frac{Du}{Dt} + \frac{1}{\Delta x}\left(f^*_{i+\frac{1}{2}-p} - f^*_{i-\frac{1}{2}-p}\right) = 0 \tag{3.52}$$

After this point, we can utilize the standard WENO procedure with the p correction and the new flux f^*. It is important to note that the residual flux f^* does not always have the same direction as the original f. This is because the residual depends on the next integer approximation of the particle's position, and the residual advection can be either a positive or a negative quantity. All this implies that applying a fixed upwinding based on the direction of the original flux f will render the NISL scheme unstable. However, for WENO schemes, this corner case is resolved automatically. Simply by using the residual flux f^*, the scheme adaptively changes the stencils and the upwinding thanks to flux splitting and smoothness coefficients.

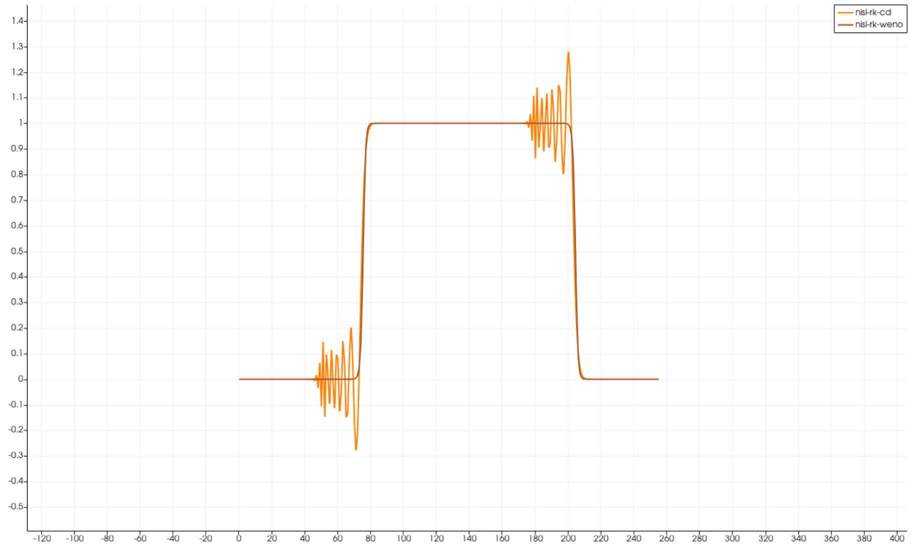


Figure 3.6: Comparison WENO and central difference oscillations with rect initial condition

3.3.3 Simulation results

This simulation presents a perfect scenario to compare an Eulerian code WENO5-RK3 with two NISL solvers (table 3.5). In this situation, the CFL condition is respected for the Eulerian solver, but for the semi-Lagrangian ones, the trajectory correction is already active. Indeed, with a $CFL = 0.512$ in this simulation, $p = 1$ is achieved.

Parameter	Value	Description
Model	$\partial_t u + v\partial_x u = 0$	1D Advection Model with constant velocity
Boundary condition	$u(0,t) = u(1,t)$	Periodic
Initial conditions	$u(x,0) = sin(x)$	sinusoidal
Δt	0.064	Starting timestep for time discretization
Δx	1/16	Starting spatial discretization
CFL	0.512	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	$[0,1]$	Spatial Domain
Solver 1	NISL-RK-WENO	RK2 WENO non interpolating semi lagrangian
Solver 2	RK3-WENO	RK3 with weno
Solver 3	RK4-WENO	RK4 WENO non interpolating semi lagrangian

Table 3.5: Table with simulation parameters for Advection model with NISL-WENO-RK

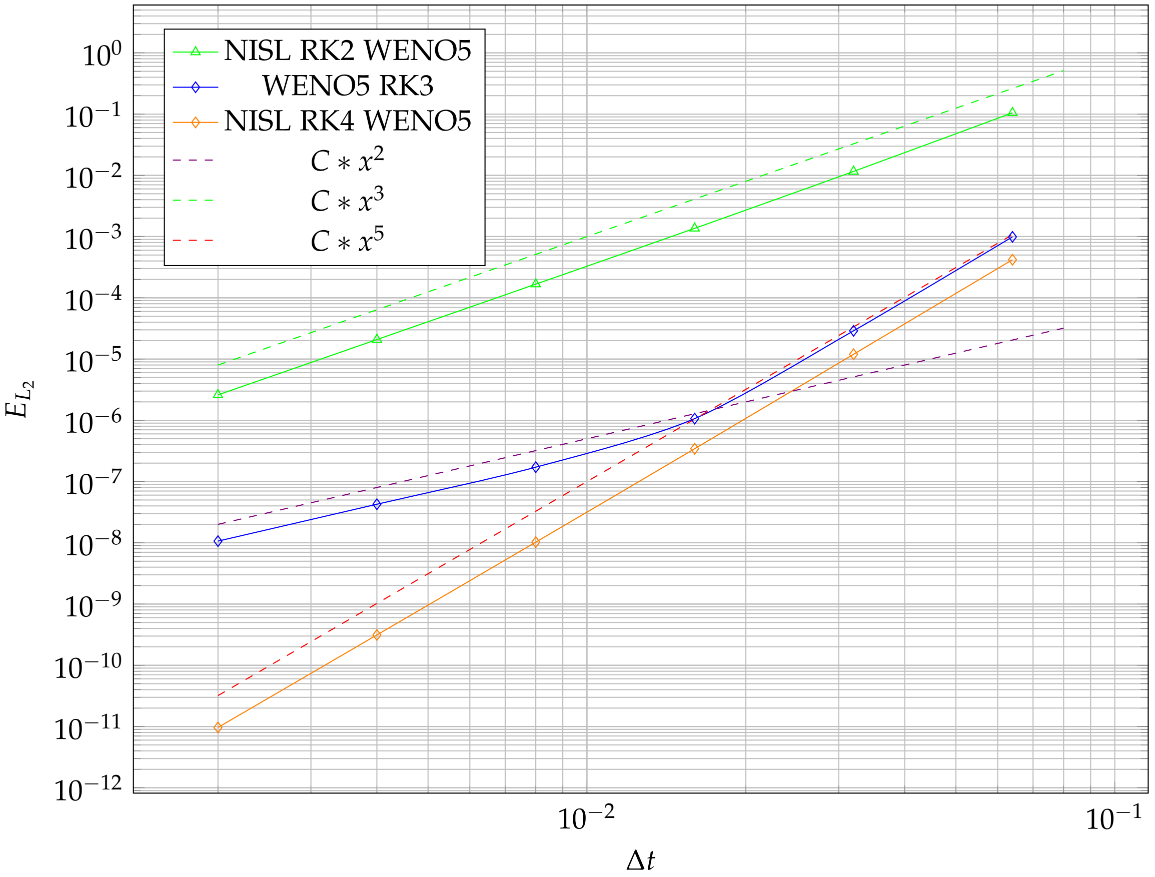


Figure 3.7: Advection RK-WENO Convergence rate comparison respected CFL condition but p=1

The results presented in the present subsection are very interesting and promising see Fig-3.7. Specifically, for WENO5-RK3, we can observe a third-order convergence as predicted by theory. For the NISL solvers, the semi-Lagrangian correction has significantly reduced the error over time, making the spatial error the dominant factor in NISL-RK2-WENO until reaching an error order of 10^{-6}. Beyond this point, the timestep becomes the dominant error. On the other hand, for NISL-RK4-WENO, the RK4 method has reduced the time error so substantially that a convergence of the 5th order is achieved, almost up to machine precision. In general, the results obtained for this type of solver are indeed very promising.

4 Burger Equation

Before testing the NISL methods for plasma physics models, we will verify the compatibility of the NISL methods with simulations of nonlinear PDE. In this chapter, we will perform a discretization of the Burgers' equation and test the convergence of this category of SL solvers in this context.

The Burgers' equation is the simplest non-linear PDE capable of producing shocks, a property that makes it very interesting from a numerical perspective. These shocks are particularly meaningful in the context of characteristic analysis. This chapter proposes a study of this model, focusing on its numerical solution through the use of non-interpolating semi-Lagrangian solvers. The Burgers' equation can be expressed as follows:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{4.1}$$

In this equation, $u(x,t)$ represents the unknown function of the variables x and t. The parameter ν is a positive constant representing the viscosity of the fluid or the diffusivity in the medium.

The Burgers' equation is nonlinear due to the term $u\partial_x u$, representing the advection or convection of the quantity u in space. This advection term can lead to the formation of shocks or discontinuities in the solution. This equation has been extensively studied in contexts such as shock wave theory, traffic flow modeling, and turbulence. Its solutions display a wide range of behaviors, from smooth profiles to the formation of shocks and rarefaction waves.

In this section, we will solve the Burgers' equation using the NISL scheme with a Leapfrog Eulerian solver.

For the purposes of this thesis, our focus will be exclusively on the 1D inviscid Burgers' equation, which omits the viscosity term. The main reason for this, as stated in the introduction, is that we are analyzing pure hyperbolic PDE. Introducing a viscosity term would complicate the treatment with the SL method, as it would transform this PDE into a parabolic one. It is, however, possible to extend the SL framework to parabolic PDE. For instance, Bonaventura has proposed the following research [16].

The inviscid Burgers' equation can be expressed as follows:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{4.2}$$

4.1 Model Discretization: Leapfrog

Starting from the simple formulation of the Burgers' equation with a source term $S(x,t)$.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = S(x,t) \tag{4.3}$$

We can proceed using the approach proposed by Ritchie. This leads to an initial splitting of the velocity field.

$$\frac{\partial u}{\partial t} + p\frac{\Delta x}{\Delta t}\frac{\partial u}{\partial x} = -(u - p\frac{\Delta x}{2\Delta t})\frac{\partial u}{\partial x} + S(x,t) \tag{4.4}$$

On the left side of the equation, we encounter an expression of the Lagrangian derivative, which can be discretized using the Leapfrog scheme, as done previously:

$$\frac{u_i^{n+1} - u_{i-p}^{n-1}}{2\Delta t} = -(u_{i-\frac{p}{2}} - p\frac{\Delta x}{2\Delta t})\frac{u^n_{i-\frac{p}{2}+1} - u^n_{i-\frac{p}{2}-1}}{2\Delta x} + S(x - \Delta x\frac{p}{2}, n\Delta t) \tag{4.5}$$

Where we remind that p can be computed by:

$$p = \lfloor 2\frac{\Delta x}{\Delta t} u^n_{i-p} \rceil. \tag{4.6}$$

The approximation of the trajectory using 4.6 in this case proves to be very accurate. Indeed, as seen in section 1.2.2, the characteristic curves are straight lines, so only one iteration is required to correctly obtain the starting point of our characteristic.

4.2 Method of Manufactured solutions

As seen in subsection 1.2.2, even starting from initial conditions that belong to $\mathbb{C}^\infty$, shocks can occur. However, the primary aim of this chapter is to demonstrate that NISL methods also work with this type of PDE. Studying the capability of these solvers in resolving shock waves is a complex task but could yield interesting results. In fact, it would theoretically be possible to use these schemes with an Eulerian component capable of resolving shock waves with high precision.

Nevertheless, for the purposes of this thesis, to verify the convergence of the solver, we will use the method of manufactured solutions [17] .

Let's start by defining the solution we wish to obtain.

$$u(x,t) = 1 + \sin(2\pi x + t) \tag{4.7}$$

By inserting this solution into the definition of the PDE 4.3, we obtain a new source term. This term will ensure that our final solution is the one we have chosen.

$$S(x,t) = (1 + 2\pi)\cos(2\pi x + t) + 2\pi \sin(2\pi x + t)\cos(2\pi x + t) \tag{4.8}$$

4.3 Simulation: Not Respected CFL

For this model, only one experiment will be presented to prove the convergence of the numerical methods (table 4.1).

Parameter Name	Value	Description
Model	$\partial_t u + u\partial_x u = S(x,t)$	1D Burger equation with Method of Manufactured Solution (MMS)
Boundary Conditions	$u(0,t) = u(1,t)$	Periodic boundary conditions
Initial Conditions	$u(x,0) = 1 + \sin(2\pi x)$	Sinusoidal wave
Δt	0.04	Starting timestep for time discretization
Δx	$1/64$	Starting spatial discretization
CFL	5.12	Courant–Friedrichs–Lewy (CFL) $CFL = \frac{\Delta t}{\Delta x}$
Ω	[0, 1]	Spatial domain
t	[0, 8]	Time Domain
Solver 2	LF-NISL	Leap-Frog non-interpolating semi-Lagrangian

Table 4.1: Table with simulation parameters for Burger equation with not respected CFL

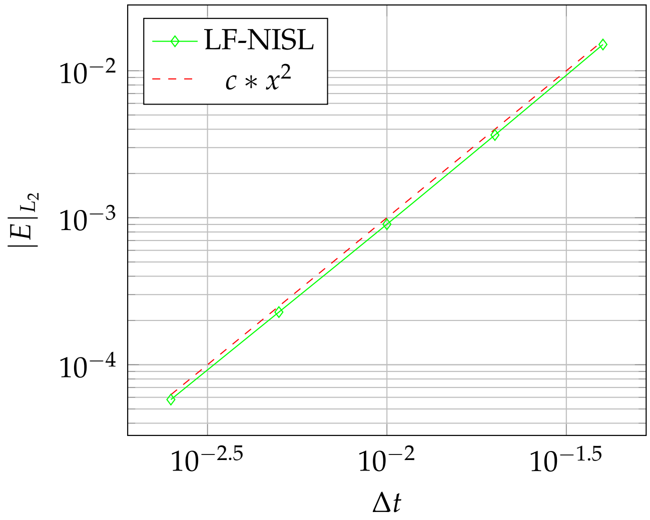


Figure 4.1: Burger equation Convergence rate comparison with not respected CFL condition

The convergence is verified in Fig-4.1. In this simulation, to test the stability of the scheme with nonlinear PDE, the Leapfrog scheme was used. For this simple simulation with a smooth solution, we can also push with higher CFL conditions. The main problem occurs when the Δt is very high, near to 0.1, at which point the simulation breaks down. Further studies could be conducted on this simple experiment using more challenging initial conditions to observe the solver's behavior in situations with increasing nonlinear oscillations and to understand its stability under such conditions.

5 Vlasov-Poisson Equation

Thanks to the increased computational power of processors and the emergence of HPC architectures with GPUs, in recent decades computational sciences have had a significant impact in the field of plasma physics. The ability of increasingly advanced numerical methods to verify and solve complex physical models is accelerating scientific research in plasma physics. Particularly, one of the main reasons has been that, in recent years, kinetic models have become computationally feasible, greatly enriching the physics of simulations. In this chapter, we will begin discussing NISL methods applied to kinetic models of plasma.

To start introducing models, let's begin by defining our quantities of interest, which in the case of kinetic simulations are the distribution functions $f(\mathbf{r}, \mathbf{v}, t)$. Distribution functions describe the position and velocity of particles in phase space $(\mathbf{r}, \mathbf{v}) : \mathbb{R}^d \times \mathbb{R}^d$ and their evolution over time $t : \mathbb{R}_{\geq 0}$.

The idea behind these distribution functions originates from the definition of the distribution for a single particle defined by the Klimontovich distribution:

$$f_k(\mathbf{r}, \mathbf{v}, t) = \sum_{i=1}^{N} \delta[\mathbf{r} - \mathbf{r}_i(t)]\delta[\mathbf{v} - \mathbf{v}_i(t)]. \tag{5.1}$$

This function is characterized by the fact that if a particle exists at a certain position in time t and at a point in phase space $(\mathbf{r}, \mathbf{v})$, then it assumes an infinite value due to the Dirac delta functions. Starting from a function that can precisely describe whether a particle exists or not, we can move to a description of the particles by integrating the Klimontovich function over a volume $\Delta\mathbf{r}\Delta\mathbf{v}$ that contains a large number of particles $N_p(\mathbf{r}, \mathbf{v}, t) \gg 1$.

$$f(\mathbf{r}, \mathbf{v}, t) = \frac{1}{\Delta\mathbf{r}\Delta\mathbf{v}} \int_{\Delta\mathbf{r}} d\mathbf{r} \int_{\Delta\mathbf{v}} d\mathbf{v} f_k = \frac{N_p(\mathbf{r}, \mathbf{v})}{\Delta\mathbf{r}\Delta\mathbf{v}} \tag{5.2}$$

In this, we have defined the quantity of interest that we will analyze and study in the upcoming chapters. This quantity tells us the density of particles in a region of space and their velocity.

5.1 The model

In the introduction to this chapter, we have seen the quantity of interest $f(\mathbf{r}, \mathbf{v}, t)$. Considering that the particles are neither destroyed nor created, this quantity obeys the continuity equation explained in section 1.1.1. Therefore, we can write that:

$$\partial_t f + \nabla_{\mathbf{r}} \cdot (f\mathbf{v}) + \nabla_{\mathbf{v}} \cdot (f\mathbf{a}) = 0 \tag{5.3}$$

In this equation, considering $\mathbf{r}$ and $\mathbf{v}$ as independent variables and assuming that $\nabla_{\mathbf{v}} \cdot \mathbf{a} = 0$, we can write:

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{r}} f + \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{v}} f = 0 \tag{5.4}$$

Where we have substituted the acceleration with the force term. In this way, we have obtained the kinetic equation.

In the Vlasov-Poisson model, charged particles are accelerated only by a self-consistent Electric Field. 'Self-consistent' means that it evolves in the simulation by moving the particles, creating a chain reaction where the electric field will modify the acceleration of the particles, which in turn will modify the electric field. Thus, the electric field is not an external force but evolves together with the system. By inserting the contribution of the electric field into the kinetic equation 5.4, we obtain the Vlasov-Poisson model in d dimensions:

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{r}} f + \frac{q}{m} \mathbf{E} \cdot \nabla_{\mathbf{v}} f = 0 \tag{5.5}$$

Here, q is the charge of the particle under consideration, and m is its mass.

To close the model, we need an equation that governs the evolution of the electric field. We know that the electric field depends on the distribution of electric charge in space. Therefore, it suffices to integrate the charge distribution over the velocity dimension to obtain the charge distribution in the position space.

$$\rho(\mathbf{r}, t) = \int_{\mathbb{R}^d} f(\mathbf{r}, \mathbf{v}, t) d\mathbf{v} \tag{5.6}$$

In this way, we can use the Poisson equation to obtain the corresponding electric potential.

$$\Delta\phi(\mathbf{r}, t) = \rho(\mathbf{r}, t) \tag{5.7}$$

Once the potential is obtained, the electric field is derived by taking its gradient.

$$\mathbf{E}(\mathbf{r}, t) = -\nabla\phi \tag{5.8}$$

This close the model, and the formulation can be applied to multiple particle species.

5.2 The characteristic equations

To define our NISL numerical schemes, it is essential to study the characteristic curves of the hyperbolic PDE introduced in 5.5. For simplicity, from this point forward, we will use Cartesian coordinates, so the position vector will be $\mathbf{r} \to \mathbf{x}$.

In this case, our Ordinary Differential Equation (ODE) system that resolves the characteristics is as follows:

$$\frac{d\boldsymbol{\xi}}{dt} = \gamma(t) \tag{5.9}$$

$$\frac{d\gamma}{dt} = \frac{q}{m} \left(\mathbf{E}(\boldsymbol{\xi}(t), t) \right) \tag{5.10}$$

Where $\boldsymbol{\xi} : \mathbb{R}_{\geq 0} \to: \mathbb{R}^d$ is the pathline associated with the spatial coordinates, while $\gamma : \mathbb{R}_{\geq 0} \to: \mathbb{R}^d$ is the curve associated with the velocity coordinates. This ODE system takes the same form as the classical equations of motion of a particle in an electric field.

The particular structure of these ODEs, thanks to the pioneering work by Horst [18], has allowed for analytical results and assumptions about the initial value problem to obtain local solutions. Moreover, thanks to the work of Pfaffelmoser [19] to expand Horst's assumptions and obtain an existence theorem for 3d models with generic initial conditions. Generally, with sufficiently smooth initial conditions and simple boundary conditions, it is possible to find an analytical solution to the characteristics and, therefore, to the PDE. However, despite these analytical results, the Vlasov-Poisson model in real cases with advanced boundary conditions remains a subject of ongoing investigation and scientific interest.

5.3 Simplified model

One of the objectives of this thesis is the development of our numerical methods on a reduced 1D-1V model of the equation 5.5. The choice of this reduced model is primarily due to the literature used. By using this model, it will be possible to compare the results obtained from the NISL solver with the main solvers studied in the drafting of this thesis ([20], [21], [22]). Furthermore, we will normalize the quantities $m_e = 1$ and $q_e = 1$, obtaining:

$$\frac{\partial f(t,x,v)}{\partial t} + v\frac{\partial f(t,x,v)}{\partial x} + E(t,x)\frac{\partial f(t,x,v)}{\partial v} = 0 \tag{5.11}$$

and

$$E(t,x) = -\frac{\partial \phi(t,x)}{\partial x}, \quad -\frac{\partial^2 \phi}{\partial x^2} = \rho(t,x). \tag{5.12}$$

Where $\rho(t,x)$ is defined from the distribution function $f(x,t)$ as

$$\rho(t,x) = \int_R f(t,x,v)dv - 1$$

In this model, we will study electron Landau damping, where the background charge is provided by stationary ions that are fixed in the background, attributable to the significant difference in mass.

5.3.1 Numerical modelling

The discretization of the proposed model will be carried out following the approach first proposed by [23] and later generalized in the framework of SL methods by Sonnendrücker in [24]. That is, to solve 5.11, we use a Strang Splitting [25]. First, we divide our PDE into the two components of advection in x and v.

$$\begin{cases} \dfrac{\partial f(t,x,v)}{\partial t} + v\dfrac{\partial f(t,x,v)}{\partial x} = 0, & x \in [0,4\pi],\ t > 0 \\ \dfrac{\partial f(t,x,v)}{\partial t} + E(t,x)\dfrac{\partial f(t,x,v)}{\partial v}, & v \in [-6,6] \end{cases} \tag{5.13}$$

Once the two new PDE to be solved separately are identified, the following steps are performed.

1. Compute a half time step in the x direction ($L \equiv -v\partial_x$)
2. Compute a full time step in the v direction ($N \equiv -E(x,t)\partial_v$)
3. Compute a half time step in the x direction ($L \equiv -v\partial_x$)

$$f(t^{n+1}) = \exp(L\Delta t/2)\exp(N\Delta t)\exp(L\Delta t/2)f(t^n) \tag{5.14}$$

By executing these steps in this order, we achieve a second-order approximation in time integration. In general, Strang-Splitting has significant properties as a symmetric integrator and is simple to understand and implement. However, it is possible to use more advanced higher-order time-splitting methods, such as those based on the theory proposed by Suzuki in [26]. The main message is that theoretically, time-splitting does not limit our solution to the second order of convergence.

5.3.2 Integration Solver

In order to move from step 1 to step 2 in the time splitting 5.14 we have to evaluate the self consistent Electric field using the Poisson equation. As a first thing we have to get the charge density $\rho(x,t)$ by integrating the distribution function $f(t,x,v)$ over the velocity in the phase space. In the continuos model we have that :

$$\rho(t,x) = \int_R f(t,x,v)dv - 1.$$

As first step to get our numerical scheme we define our domain in the phase space Ω.
Let Ω be the phase space domain defined as:

$$\Omega = \{(x,v) \in \mathbb{R}^2 : x_{\min} \leq x \leq x_{\max},\, v_{\min} \leq v \leq v_{\max}\}$$

Where:

$$\begin{aligned} x_{\min} &: \text{Minimum value of } x \\ x_{\max} &: \text{Maximum value of } x \\ v_{\min} &: \text{Minimum value of } v \\ v_{\max} &: \text{Maximum value of } v \end{aligned}$$

In this way we get that the previous integral became:

$$\rho(t,x) = \int_{v_{\min}}^{v_{\max}} f(t,x,v)dv - 1$$

To discretize the domain Ω, we introduce the following intervals:

Δx : Spatial discretization step, dividing the x-interval into N_x segments.
Δv : Velocity discretization step, dividing the v-interval into N_v segments.

Thus, the discretized phase space domain is represented by the grid points:

(x_i, v_j) where $x_i = x_{\min} + i\Delta x$, $v_j = v_{\min} + j\Delta v$, for $i = 0,1,\ldots,N_x$ and $j = 0,1,\ldots,N_v$.

After these definition the continous integral became a sum over the velocity dimension in the phase space.
Using the Composite Trapezoidal Rule for the velocity discretization, using all the discretization points, the charge density at a specific spatial point x_i is:

$$\rho(x_i,t) \approx \sum_{j=0}^{N_v-1} \frac{\Delta v}{2} \left[f(x_i,v_j,t) + f(x_i,v_{j+1},t)\right]$$

Using the Composite Simpson's 3/8 Rule for the velocity discretization, using all the discretization points, the charge density at a specific spatial point x_i is:

$$\rho(x_i,t) \approx \sum_{k=0}^{N_v/3-1} \frac{3\Delta v}{8} \left[f(x_i,v_{3k},t) + 3f(x_i,v_{3k+1},t) + 3f(x_i,v_{3k+2},t) + f(x_i,v_{3k+3},t)\right]$$

A more advanced approach can use a Composite Adaptive Simpson rule with Richardson extrapolation. Another more advanced is to use a Aitken-Neville extrapolation over the different levels. Overall exploiting using extrapolation and adaptivity we can reach a lower amount of evaluation of the function and improve the numerical performance.

5.3.3 Integration Solver: Numerical Experiments

Vlasov-Poisson solvers are complex solvers with multicomponent solvers, one of the most important part of writing code for this solver is to proof that each component of the solver is properly written and working as expected. In this brief section a functional test is presented, by doing this test we can evaluate the error in this procedure. After this point the integration procedure will be considered correct and just a function to call in the code.

We start our experiments by using a Manufactured Solution of the type:

$$f(x,v) = (2 + cos(2\pi x))e^{-v^2} \tag{5.15}$$

Where x and v are the normalized quantities.

Using this manufactured solution we can compute analytically the integral to get the charge density.

$$\rho(x) = \int_R f(t,x,v)dv - 1 = (-\sqrt{\pi})(2 + cos(2\pi x)) - 1 \tag{5.16}$$

Trapezoidal rule results give us already machine precision results:

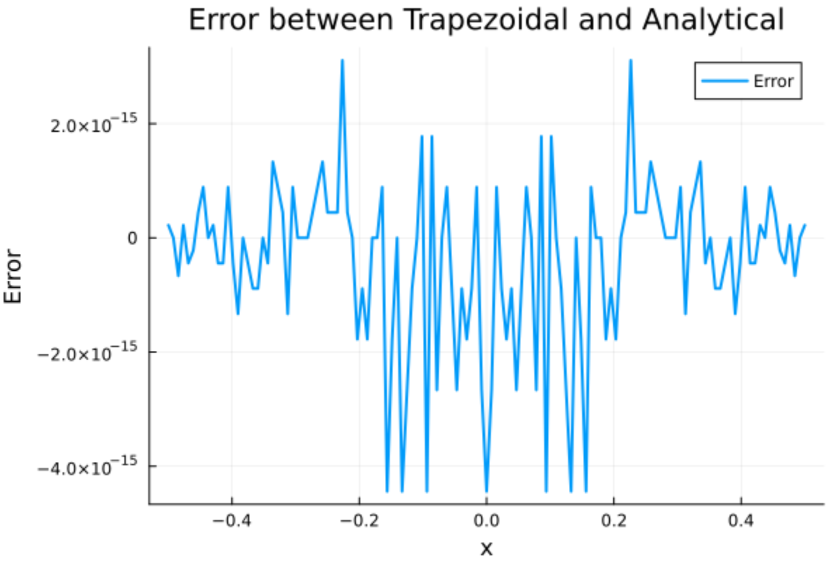


Figure 5.1: Composite Trapezoidal rule with all the points

A relevant result from this test is that the integration error will not be a leading error and that in future (see Fig-5.1) it would be possible to reduce the number of evaluation of the function. However it is important to note that this computational cost is not a leading term and does not have particular priority in the possible improvement of the solver.

5.3.4 Poisson Equation Solver

In order to obtain the self-consistent Electric Field we have to numerically solve the Poisson equation. Given the choice of the periodic boundary condition over the x-axis an easy possible choice is to use a spectral-solver using the computational advantageous Fast Fourier's Transform.

Given the Poisson equation on the domain $[-0.5, 0.5]$ with periodic boundary conditions:

$$-\frac{\partial^2 \phi}{\partial x^2} = \rho(x) \tag{5.17}$$

We can Fourier transform the equation getting and algebraic equation

$$k^2 \hat{u}(k) = \hat{\rho}(k) \tag{5.18}$$

where:

- $\hat{\rho}(k) = \mathcal{F}\{\rho(x)\}$
- Using the derivative property of the Fourier transform $k^2 \hat{u}(k) = \mathcal{F}\{-\frac{\partial^2 \phi}{\partial x^2}\}$

We recall that this approach is valid only with periodic boundary conditions. With other types of boundary conditions, such as Dirichlet, we would need to use different transformations or approaches. For instance, an Algebraic Multigrid approach could be used with frameworks available out-of-the-box. Geometric multigrid methods might be better suited given the simplicity of the domain, but it would require writing the solver from scratch for this particular case.

From equation 5.18 we can trivially solve this linear system in the frequency domain.

$$\hat{u}(k) = \frac{\hat{\rho}(k)}{k^2} \tag{5.19}$$

Compute the inverse Fourier transform to obtain the solution in the spatial domain:

$$u(x_j) = \mathcal{F}^{-1}\{\hat{u}(k)\} \tag{5.20}$$

In the numerical implementation all the Fourier Transforms are substituted by a Fast Fourier Transform(FFT) algorithm.

5.3.5 Runge-Kutta Time integration for NISL

In this section, we introduce one of the most interesting results of this thesis, which is the use of Runge-Kutta methods in this NISL framework.

The main idea lies in the fact that once we have decomposed the velocity according to Ritchie 2.13, in general, this scheme can be treated as a pure Eulerian scheme using Rančić's intuition and thus simply performing an Eulerian step. Let's start by writing the Butcher tableau for a simple RK2 integrator.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

Let's take for example the advection over the v dimension of our Vlasov-Poisson Equation.

$$\frac{\partial f}{\partial t} = -E(x,t)\frac{\partial f}{\partial v} \tag{5.21}$$

In this case we will only take one step of length Δt in time. Hence our decomposition of the advection velocity according to Ritchie becomes:

$$\frac{Df}{Dt} = -(E(x,t) - \frac{p\Delta v}{\Delta t})\frac{\partial f}{\partial v} \tag{5.22}$$

Now that we have our equation 5.22 in semi-Lagrangian form, we apply the trajectory correction with the parameter p. It is important to apply the p correction only in the first stage. By doing this, we will evolve points from the base of the characteristic curve, and it will not be necessary to evaluate intermediate points as in the case of 2.16. Therefore, we obtain that our RK2 algorithm becomes:

1. $f^* = f^n_{(i,j-p)} - \frac{\Delta t}{2}\left(E^n_i - \frac{p\Delta v}{\Delta t}\right)\frac{\partial f^n_{(i,j-p)}}{\partial v}$

2. $f^{n+1} = f^n_{(i,j-p^*)} - \frac{\Delta t}{2}\left(E^* - \frac{p^*\Delta v}{\Delta t}\right)\frac{\partial f^*}{\partial v}$

Where E^* is the electric field updated with the information at the midpoint Δt^*. Moreover, the p correction also depends on the electric field, so it needs to be recalculated accordingly. A very simple and effective improvement of this scheme is given by the TVD (Total Variation Diminishing) method proposed in [11].

In which we find that the second stage has the following form:

$$f^{n+1} = \frac{1}{2}f^n_{(i,j-p^*)} + \frac{1}{2}f^* - \frac{\Delta t}{2}\left(E^* - \frac{p^*\Delta v}{\Delta t}\right)\frac{\partial f^*}{\partial v} \tag{5.23}$$

.

In general, it is possible to write higher-order RK methods. However, care must be taken to correctly update the values of E and p accordingly. In our case, having used a second-order time splitting, it is not necessary to increase the order in this step.

5.4 Initial Value Problem: Weak Landau Damping

The first IVP on which to test our implementation is the Weak-Landau Damping, a widely used model for validating implementations thanks to the significant analytical results obtained by Landau [27], against which the obtained solution can be verified. Generally, Landau damping is a phenomenon of particle-wave electromagnetic interaction, where an exchange of energy between the wave and the particle occurs when the phase velocity of the electromagnetic wave is very close to the particle's velocity as it is possible to see in figure 5.2. Generally, this leads to an exponential decrease in the strength of the electric field, thereby altering the particles' velocities.

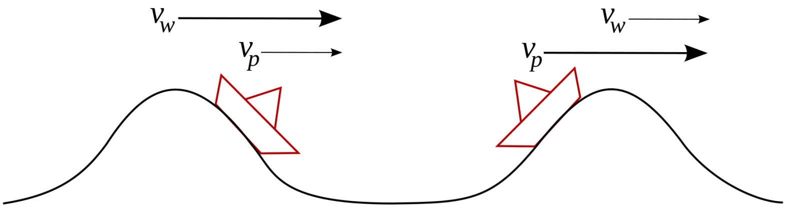


Figure 5.2: Sailboats (or surfers) catching an ocean wave are sped up and slowed down to match the wave speed. [28]

The initial condition to obtain the Weak Landau Damping is the following:

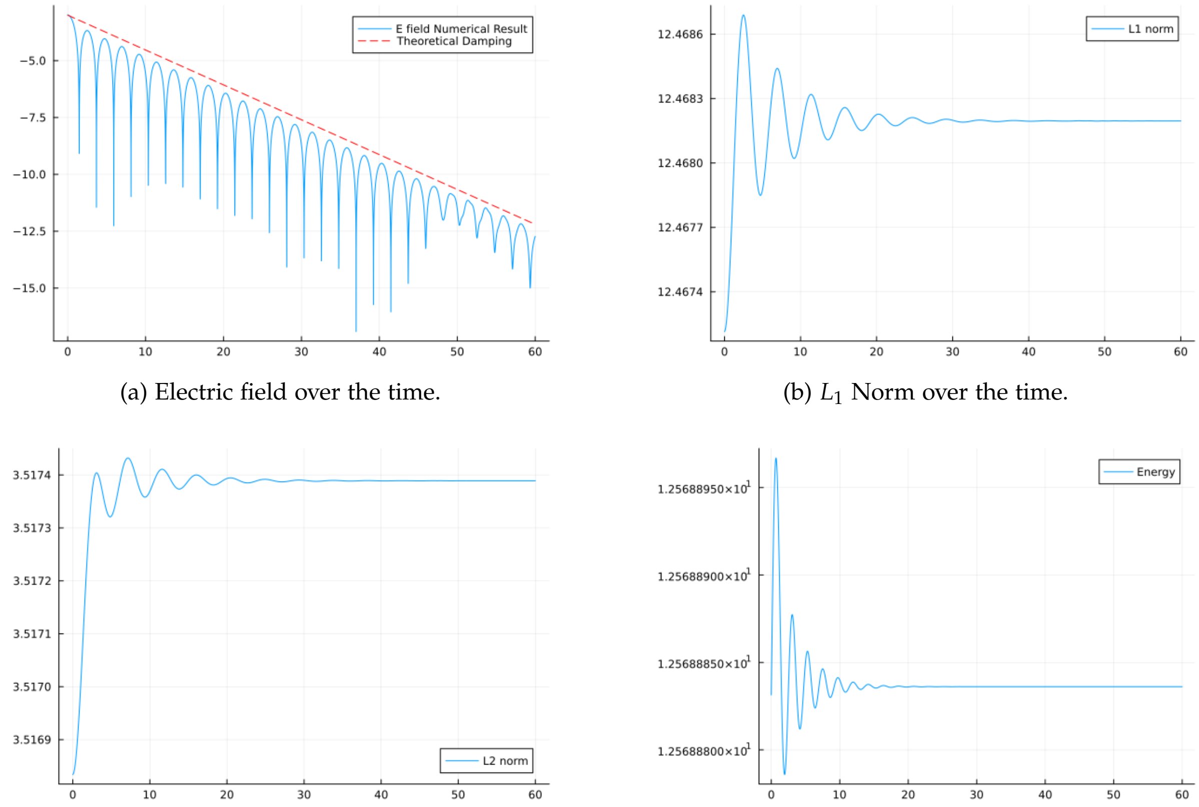
$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}}\left(1 + 0.01cos\left(\frac{1}{2}\right)\right)e^{-\frac{v^2}{2}} \tag{5.24}$$

5.4.1 Simulation Results: Respected CFL

In this first simulation (table 5.1), we show the results of the NISL RK2CD solver for an initial condition of weak Landau Damping with a contained CFL condition. However, in this case, since the CFL is $cfl > 0.5$, this means that the semi-Lagrangian correction will come into play at the limits of the simulation where the velocity assumes higher values.

Parameter Name	Value	Description
Model	$\partial_t f + x\partial_x f + E\partial_v f = 0$	1D Vlasov-Poisson
Boundary conditions	$f(0,v,t) = f(4\pi,v,t)$	Periodic Boundary Conditions
Initial conditions	$f(x,v,0) = \frac{1}{\sqrt{2\pi}}\left(1+0.01cos\left(\frac{1}{2}x\right)\right)e^{-\frac{v^2}{2}}$	Weak Landau Damping
Δt	0.01	Fixed Timestep for time discretization
$N_{x,v}$	128	Fixed Spatial Discretization
CFL	0.611	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	$[0, 4\pi]x[-6,6]$	Phase-Space Domain
t	[0, 60]	Starting Time Domain
Solver	NISL-RK2CD	Non interpolating Semi Lagrangian Solver

Table 5.1: Table with simulation parameters for Vlasov-Poisson model with respected CFL



(a) Electric field over the time.

(b) L_1 Norm over the time.

(c) L_2 Norm over the time.

(d) Energy over the time.

Figure 5.3: Simulation results for Weak-Landau Damping with respected CFL.

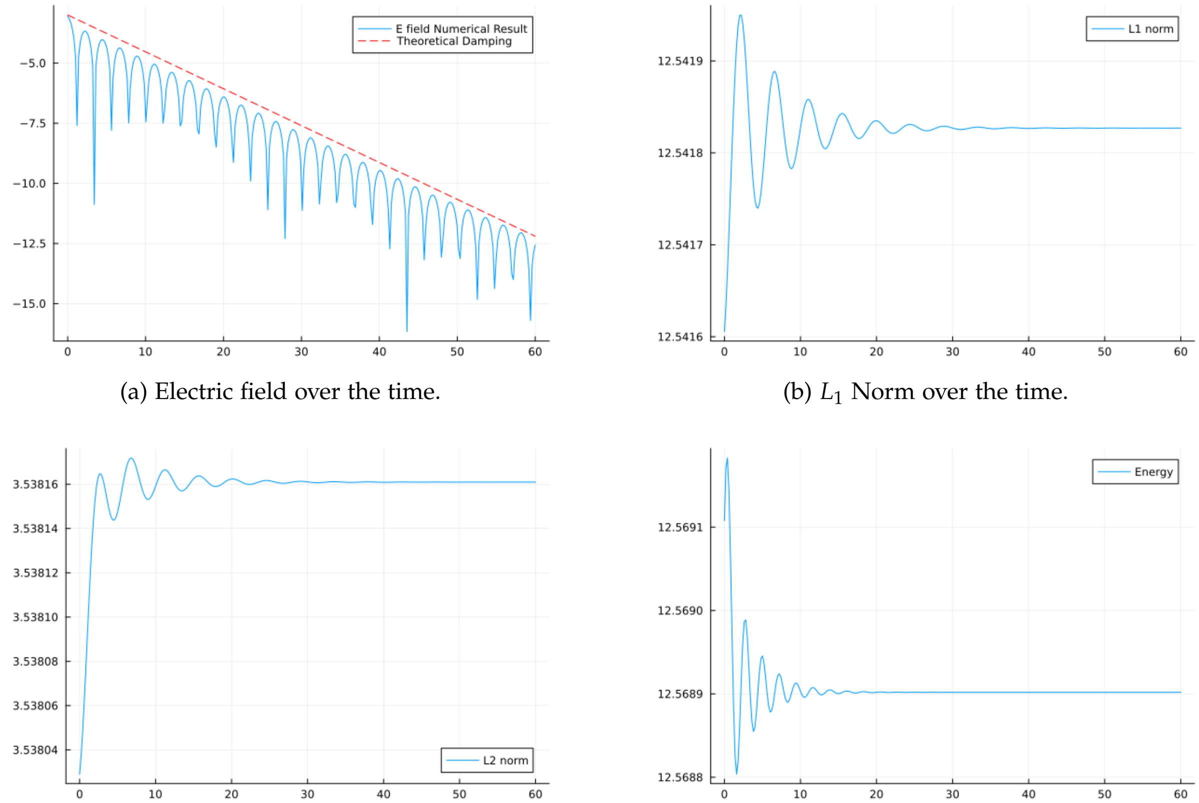
One of the most important quantities for this type of simulation is the electric field (Fig-5.3) and its evolution over time . Indeed, we can verify the physical correctness of the simulation by comparing the decrease in the electric field predicted by theory with that of our solver. In this case, the electric field very closely follows the reduction predicted by the theory. Particularly, other important quantities are the L_1 norm, the L_2 norm, and the energy norm of the simulation, which must be conserved. In this case, after a transient period, these quantities converge to a fixed value.

5.4.2 Simulation Results: Not Respected CFL

The second simulation (table 5.2) presented is under a condition of non-compliance with the CFL. Indeed, from this simulation onwards, the semi-Lagrangian correction starts to have a significant impact in the advection along the x-direction where the advection velocity is higher.

Parameter Name	Value	Description
Model	$\partial_t f + x\partial_x f + E\partial_v f = 0$	1D Vlasov-Poisson
Boundary conditions	$f(0,v,t) = f(4\pi,v,t)$	Periodic Boundary Conditions
Initial conditions	$f(x,v,0) = \frac{1}{\sqrt{2\pi}}\left(1 + 0.01cos\left(\frac{1}{2}x\right)\right)e^{-\frac{v^2}{2}}$	Weak Landau Damping
Δt	0.1	Fixed Timestep for time discretization
N$_{x,v}$	256	Fixed Spatial Discretization
CFL	12.22	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	$[0, 4\pi]x[-6,6]$	Phase-Space Domain
t	[0, 60]	Starting Time Domain
Solver	NISL-RK2CD	Non interpolating Semi Lagrangian Solver

Table 5.2: Table with simulation parameters for Vlasov-Poisson model with not respected CFL



(a) Electric field over the time.

(b) L_1 Norm over the time.

(c) L_2 Norm over the time.

(d) Energy over the time.

Figure 5.4: Simulation results for Weak-Landau Damping with not respected CFL.

The results obtained are very positive as the presence of the semi-Lagrangian correction allows for a more accurate electric field even after 60s Fig-5.4. Moreover, to verify the correct implementation of the solver, convergence results at this CFL value are added Fig-5.6.

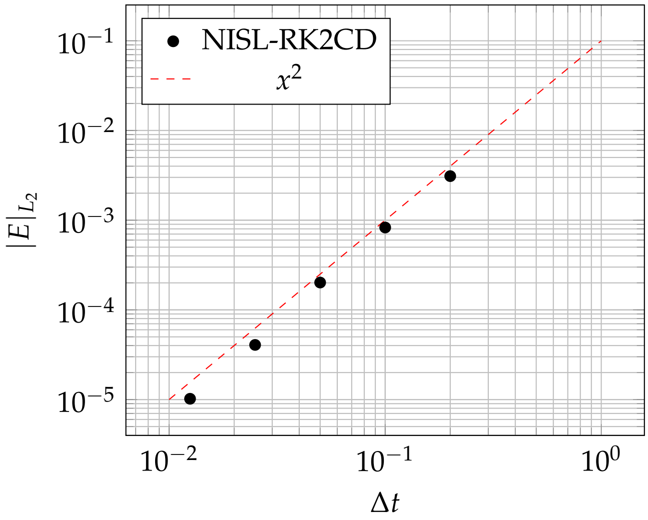


Figure 5.5: Error vs. Δt

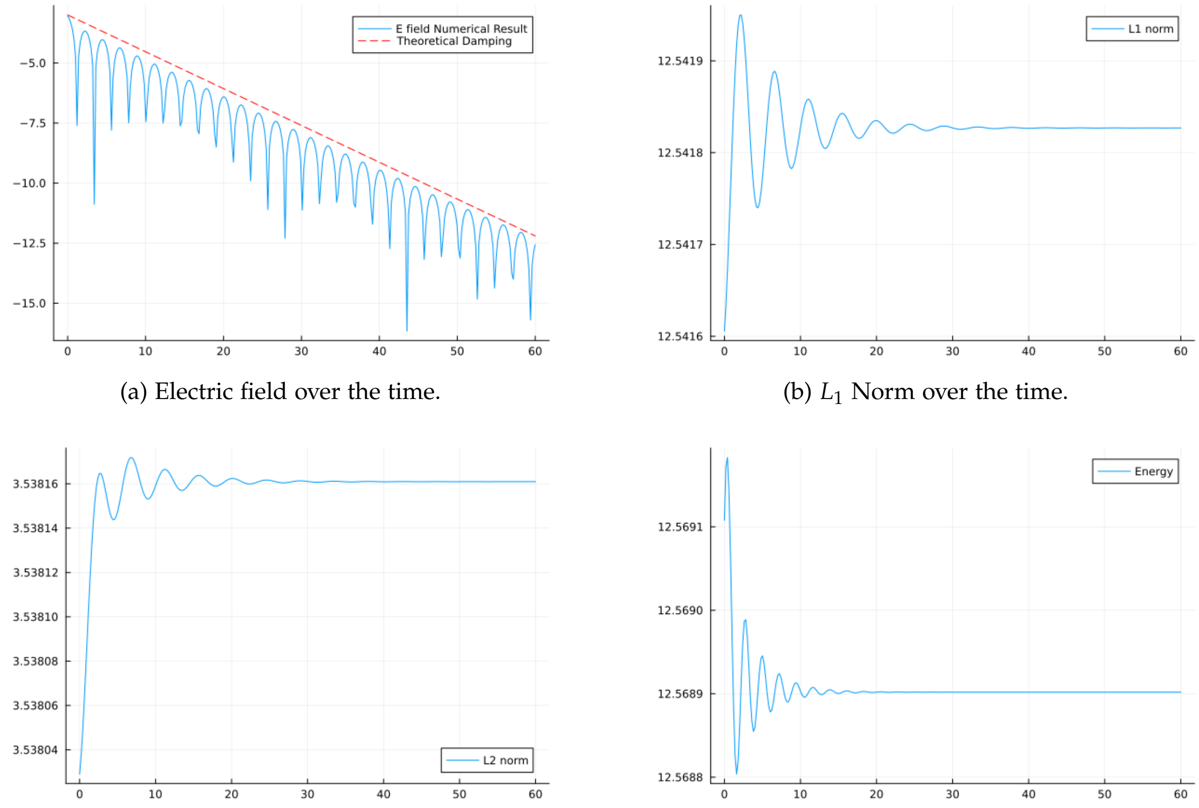
Figure 5.6: Convergence rate comparison with not respected CFL condition Vlasov Poisson

5.4.3 Simulation Results: High CFL

Experimentally, it has been verified that it is possible to work with very high CFL levels without too many problems while still maintaining good results (table 5.3).

Parameter Name	Value	Description
Model	$\partial_t f + x\partial_x f + E\partial_v f = 0$	1D Vlasov-Poisson
Boundary conditions	$f(0,v,t) = f(4\pi,v,t)$	Periodic Boundary Conditions
Initial conditions	$f(x,v,0) = \frac{1}{\sqrt{2\pi}}\left(1 + 0.01cos\left(\frac{1}{2}x\right)\right)e^{-\frac{v^2}{2}}$	Weak Landau Damping
Δt	0.2	Fixed Timestep for time discretization
N$_{x,v}$	512	Fixed Spatial Discretization
CFL	48.89	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	$[0, 4\pi]x[-6,6]$	Phase-Space Domain
t	[0, 60]	Starting Time Domain
Solver	NISL-RK2CD	Non interpolating Semi Lagrangian Solver

Table 5.3: Table with simulation parameters for Vlasov-Poisson model with high CFL



(a) Electric field over the time.

(b) L_1 Norm over the time.

(c) L_2 Norm over the time.

(d) Energy over the time.

Figure 5.7: Simulation results for Weak-Landau Damping with high CFL.

The results obtained in Fig-5.7 do not qualitatively differ from those obtained with the lower CFL earlier, making this simulation much more efficient for simulations that need to cover a larger time interval.

5.5 Initial Value Problem: Strong Landau Damping

In the simulations dedicated to weak Landau Damping (table 5.4), very promising results were shown. In particular, one of the reasons why the previously shown results are especially good is that the semi-Lagrangian correction primarily occurs in the x-direction of the spatial coordinates, especially in the PDE $\partial_t f + v\partial_x f = 0$. This PDE admits a simple analytical solution, so the characteristics are resolved analytically after a single iteration to obtain the coefficient p. This implies that the semi-Lagrangian method applied to that PDE is a semi-analytical scheme. Another interesting test presented in this section is with Strong Landau Damping, where the advection component along the velocity dimension plays a more important role.

Parameter Name	Value	Description
Model	$\partial_t f + x\partial_x f + E\partial_v f = 0$	1D Vlasov-Poisson
Boundary conditions	$f(0,v,t) = f(4\pi,v,t)$	Periodic Boundary Conditions
Initial conditions	$f(x,v,0) = \frac{1}{\sqrt{2\pi}}\left(1+0.5cos\left(\frac{1}{2}x\right)\right)e^{-\frac{v^2}{2}}$	Weak Landau Damping
Δt	0.04	Fixed Timestep for time discretization
N$_{x,v}$	256	Fixed Spatial Discretization
CFL	4.89	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	$[0, 4\pi]x[-6,6]$	Phase-Space Domain
t	[0, 60]	Starting Time Domain
Solver	NISL-RK2CD	Non interpolating Semi Lagrangian Solver

Table 5.4: Table with simulation parameters for Vlasov-Poisson model with not respected CFL Strong Landau Damping

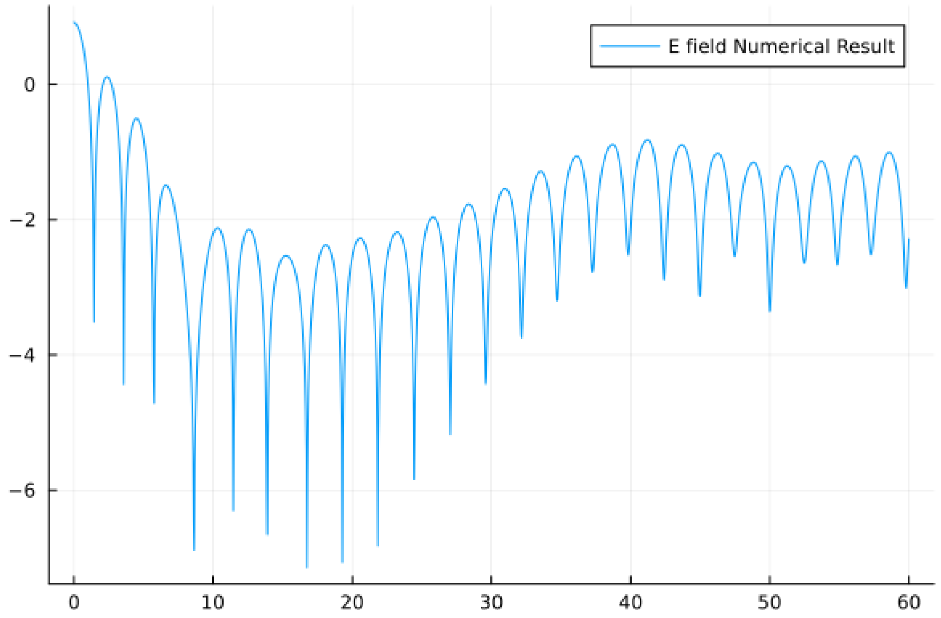
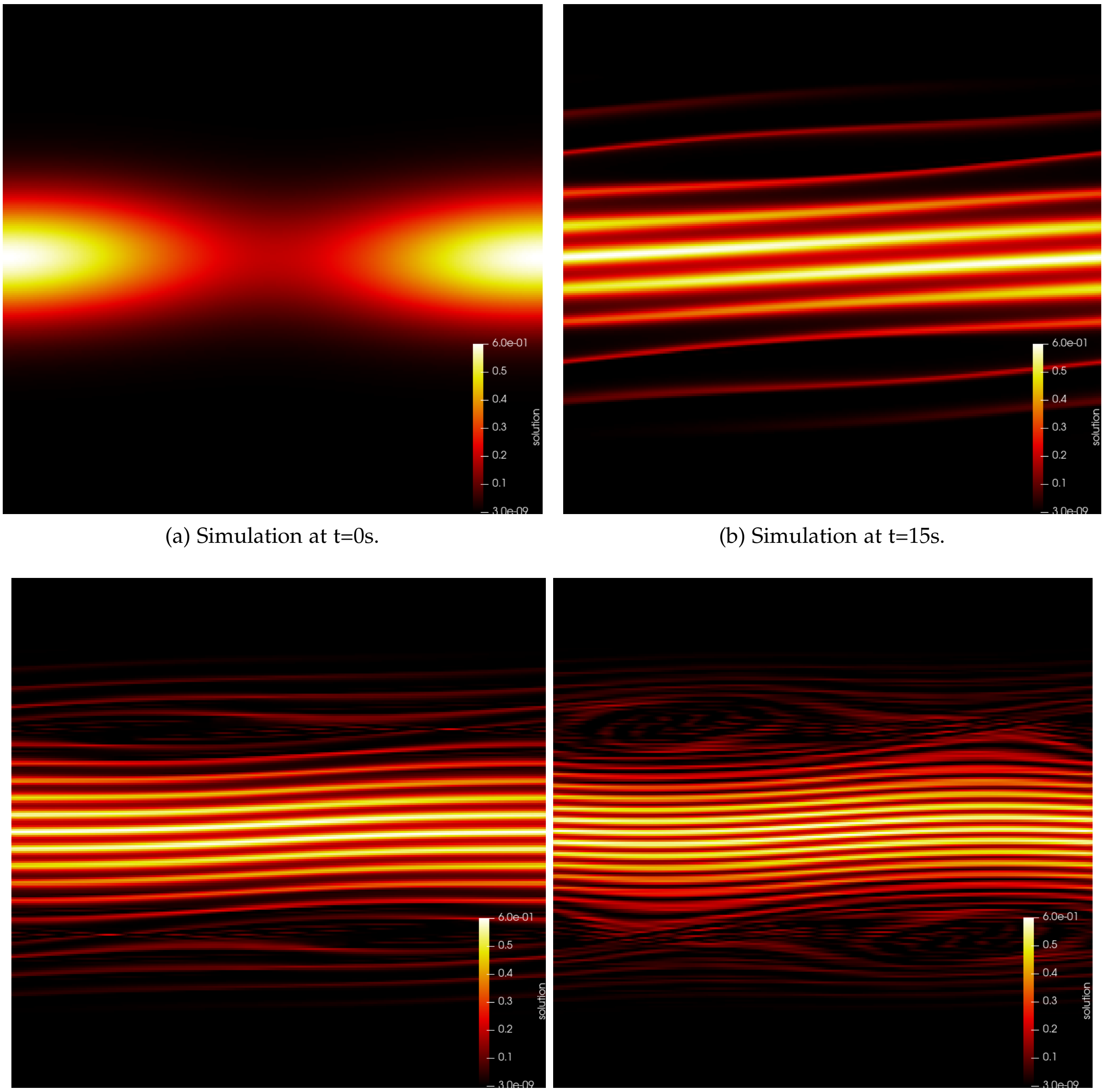


Figure 5.8: Electric field over the time for Strong landau damping

The electric field obtained in Fig-5.8, in line with theoretical results, assumes higher values, allowing also for the testing of the semi-Lagrangian correction with advection in the v-direction.



(a) Simulation at t=0s.

(b) Simulation at t=15s.

(c) Simulation at t=30s.

(d) Simulation at t=45s.

Figure 5.9: Simulation results for Strong-Landau Damping with not respected CFL

In these sequences of the simulation(Fig-5.9), we can observe the solver's ability to capture very small filaments and details. However, the presence of very thin plasma filaments represents a significant challenge for the central difference scheme proposed in the spatial discretization. In general, it has been possible to push to higher CFL values while still obtaining electric field values conforming to the theory. The main limitation of these simulations lies in the absolute size of the timestep. Indeed, using too high timesteps, like $\Delta t > 0.1$, even if not breaking the CFL condition, the simulation will significantly suffer from the error in time, causing oscillations that spoil and completely degrade the solution.

6 Plasma Sheath Formation

In the European Fusion Development Agreement [29], the physics of exhausted plasma is identified as a primary challenge. This aspect of nuclear fusion is not just a hurdle but a multidisciplinary quest. The power derived from the exhausted plasma poses a significant engineering challenge, especially when it redistributes on the plasma's edge, potentially damaging the tokamak's walls. Without a comprehensive understanding of these phenomena, achieving fusion remains very challenging.

In recent years, thanks to the work of the scientific community in attempting to physically model the behavior of plasma near the walls, interesting results have been obtained, and the literature on these models continuously enriched [30] [31]. Despite the remarkable achievements in recent years, conducting a global simulation of a tokamak that can resolve the physics of the walls with high fidelity remains very difficult. One of the fundamental reasons for this intrinsic difficulty is multiscale physics. Electrons, due to their lighter nature compared to ions, quickly reach internal surfaces, giving rise to the Debye sheath [32]. This non-neutral layer, although confined to a thin sheath, significantly impact on the wall with phenomenon like sputtering and erosion [33]. Addressing this requires a multiscale physics approach, spanning from the tokamak's meters-wide radius to the sheaths' few millimeters.

This multiscale physics is crucially related to the simulation when discussing the CFL condition. Indeed, to correctly resolve the boundary layers, a very high definition and therefore a very small Δx is necessary. In standard Eulerian codes, this necessitates very small Δt due to the restriction of the CFL condition. In particular, the possibility of overcoming this restriction through SL solvers has aroused much interest in the scientific community, leading to the creation of global solvers like GYSELA [34].

6.1 Model

In this chapter, we explore a more challenging model to assess the performance of the new schemes. We will investigate the generation of the plasma sheath, following the models proposed by Dr. Michael Hardman and his collaborators in [35]. This model generates the pre-sheath [36] [37], which is the layer of plasma that comes into contact with the wall, modeled by the wall boundary conditions. These boundary conditions and the ion source term create a diverging electric field near the wall [32], providing a perfect test for our solver to identify weaknesses and facilitate improvements.

We begin by defining the ion-drift kinetic equation in 1D-1V:

$$\frac{\partial f}{\partial t} + b_z v_\parallel \frac{\partial f}{\partial z} + \frac{b_z E_z}{2} \frac{\partial f}{\partial v_\parallel} = S \tag{6.1}$$

Here, we have a magnetic field in the form: $\mathbf{B} = b_z \hat{\mathbf{z}}$. This represents a guide magnetic field that is constant over the z direction.

In the context of this chapter, we will use the Boltzmann response to derive the potential from the ion distribution f following the report [35] Specifically:

$$n_i = n_e = N_e e^{\frac{\phi}{T_e}} \tag{6.2}$$

$$\phi = T_e log(\frac{n_i}{N_e}) \tag{6.3}$$

Once obtained the electric potential from the ion distribution we can compute the electric fields over the z direction.

$$E_z = -\frac{\partial \phi}{\partial z}. \tag{6.4}$$

The space charge density is obtained from the phase-space distribution by integration:

$$n_i = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} f dv_{\parallel}. \tag{6.5}$$

Where we have introduced the $1/\sqrt{\pi}$ as a normalization factor. To have a simulation that converges to a real solution of the model, we need to define an initial condition that is compatible with our boundary conditions. Following the reference [35], the following initial condition will be used:

$$\begin{split} f_i = \Bigg[H(v_{\parallel}) v_{\parallel}^4 \left(\frac{1}{2} + \frac{z}{L_z} \right) n_+(z) + H(-v_{\parallel}) v_{\parallel}^4 \left(\frac{1}{2} - \frac{z}{L_z} \right) n_-(z) \\ \left(\frac{1}{2} - \frac{z}{L_z} \right) \left(\frac{1}{2} + \frac{z}{L_z} n_0(z) \right) \Bigg] \exp\left(-v_{\parallel}^2 \right) \end{split} \tag{6.6}$$

The following elements are introduced:

- $H(v_{\parallel})$ which is a Heaviside step function
- $n_{\pm} = \exp\left[1 + \left(\epsilon + \frac{1}{2} \mp \frac{z}{L_z} \right)^{\frac{1}{2}} \right]$, the ϵ parameter is necessary to prevent the electric field from diverging inside the domain.
- $n_0 = \exp[1]$.

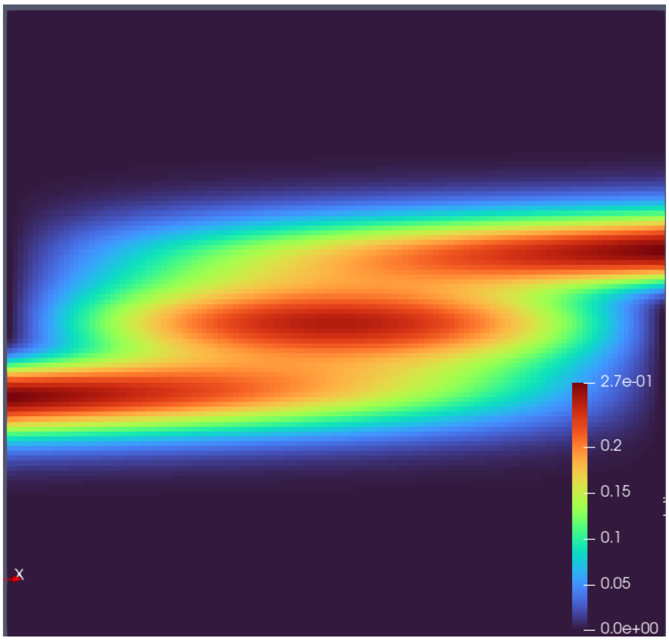


Figure 6.1: Initial condition

An image of the initial condition is also proposed in 6.1 to provide a visual idea.

In addition to the initial conditions, we also define a source term $S(v)$ that represents a constant and uniform source of ions originating from neutral-electron collisions.

$$S(v) = 2\exp\left[-4v^2\right] \tag{6.7}$$

6.1.1 Wall Boundary Conditions

The most interesting part of this model lies in the definition of the boundary conditions(Fig-6.2), which follow the idea of the logical sheath boundary conditions of Parker [38] and use the model proposed in [39] by F. I. Parra. Specifically, in our model with ions, these boundary conditions mean that no ion can return from the wall.

Analyzing the initial condition (see the equation 6.6), it is simple to verify the compatibility with these wall boundary conditions. In fact for $z = -L_z/2$ and $z = L_z/2$, the ion distribution respectively is $f_i(z, v_\parallel > 0) = 0$ and $f_i(z, v_\parallel < 0) = 0$.

$$f(z = -2\pi, v_\parallel > 0, t) = 0 \tag{6.8}$$

$$f(z = 2\pi, v_\parallel < 0, t) = 0 \tag{6.9}$$

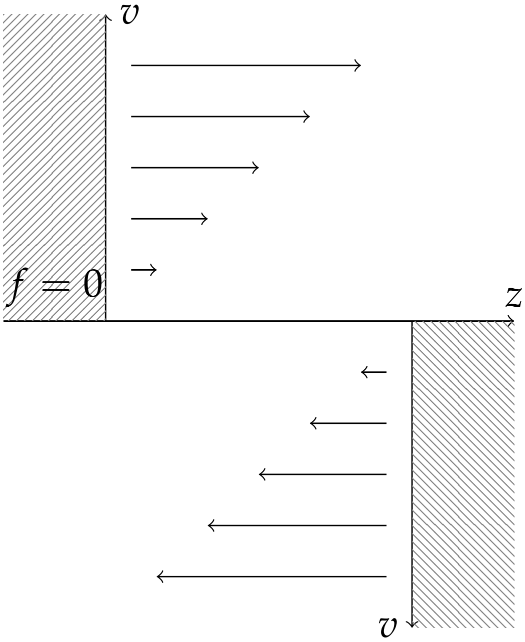


Figure 6.2: Example of Boundary Conditions

6.2 Numerical Modelling: Central Difference approximation over $v_\parallel$

In this first simulation, it will be presented the results using the same solver that was employed in 5.3.1 for the Landau damping.

$$\begin{cases} \dfrac{\partial f}{\partial t} + b_z v_\parallel \dfrac{\partial f}{\partial z} = 0, & z \in [-2\pi, 2\pi],\ t > 0 \\ \dfrac{\partial f}{\partial t} + E\dfrac{\partial f}{\partial v_\parallel}, & v_\parallel \in [-6, 6] \end{cases} \tag{6.10}$$

Using a Strang Splitting we obtain the following steps:

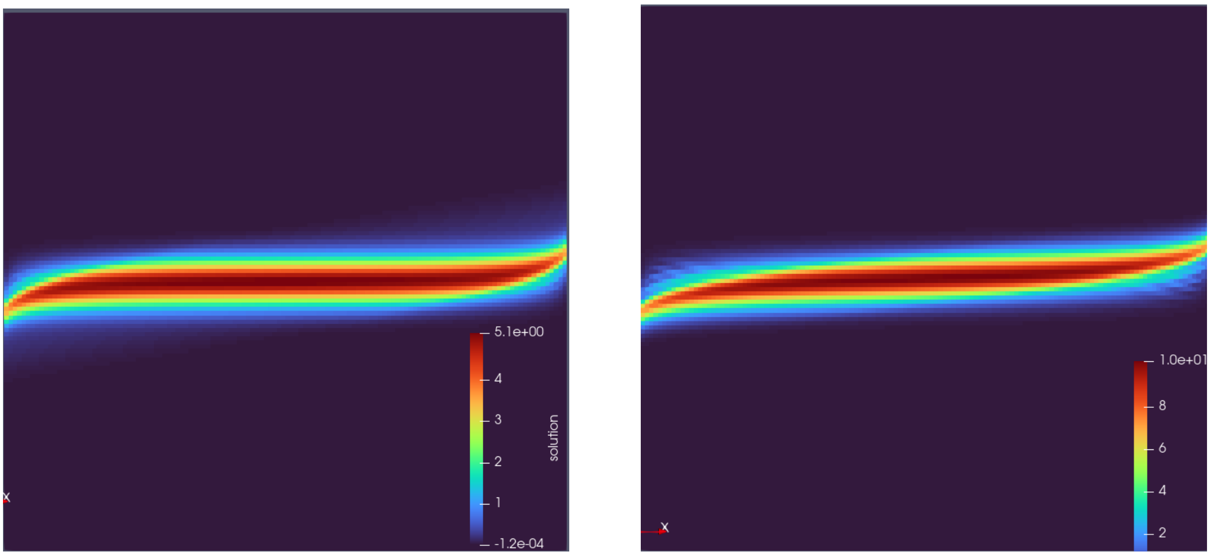
- Compute half time step in the z direction ($L \equiv -v_\parallel \partial_z$) LW-NISL
- Compute a full time step in the v direction ($N \equiv -E\partial_{v_\parallel}$) RK2-CD-NISL
- Compute half timestep in the z direction ($L \equiv -v_\parallel \partial_z$) LW-NISL

6.2.1 Simulation results: not respected CFL

The first simulation is conducted with a non-compliant CFL to analyze the obtained results(table 6.1).

Parameter Name	Value	Description
Model	$\partial_t f + v_\parallel \partial_z f + E\partial_{v_\parallel} f = 0$	1D-1V Drift Kinetic with source
Boundary conditions	$f(z = -2\pi, v_\parallel > 0, t) = 0$	Wall Boundary Conditions
Initial conditions	See refernce update me	Sheat Formation
Δt	0.1	Fixed Timestep for time discretization
N$_{x,v}$	128	Fixed Spatial Discretization
CFL	6.0638	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	[-$2\pi, 2\pi$]$x[-6, 6]$	Phase-Space Domain
t	[0, 10]	Starting Time Domain
Solver	LW-NISL / NISL-RK2CD	Non interpolating Semi Lagrangian Solver

Table 6.1: Table with simulation parameters for Sheath Formation model with not respected CFL

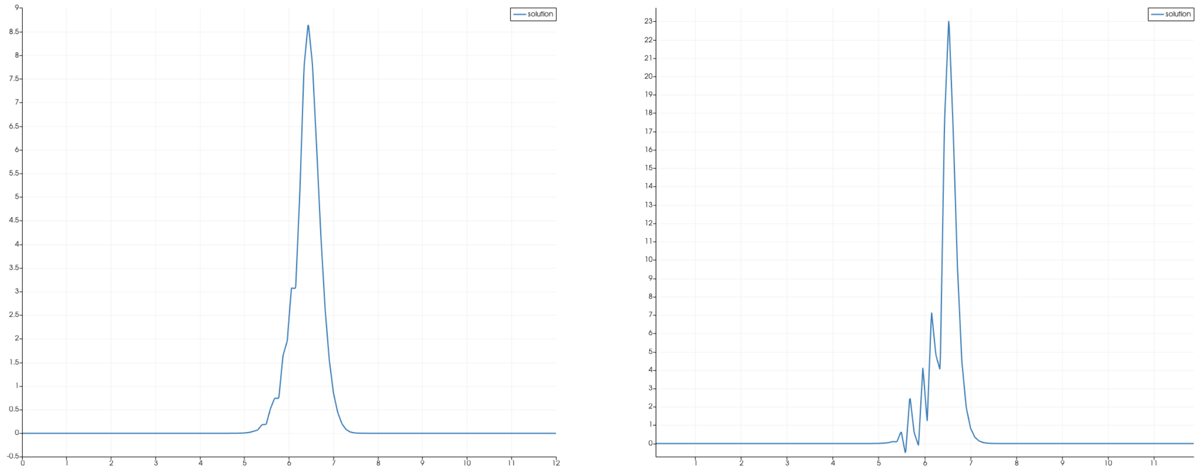


(a) Simulation at normalized time t=5.0s

(b) Simulation at normalized time t=10.0s

Figure 6.3: Simulation results RK2-CD

In the obtained simulation, it is possible to observe vertical oscillations propagating in the v direction, near the sheath entrance . These oscillations in Fig-6.3 tend to increase in number over time until they destroy the solution. However, the first observation to make is that these oscillations are not due to the CFL condition, but rather to oscillations caused by the spatial discretization used in the v direction. Indeed, instabilities arising from the CFL condition typically increase the energy in the system, creating oscillations that amplify until reaching infinite values, while these oscillations have a finite amplitude.



(a) Distribution function at the sheath entrance $f(z = -2\pi, v_{\|}, t)$ at t = 10s

(b) Distribution function at the sheath entrance $f(z = -2\pi, v_{\|}, t)$ at t = 30s

Figure 6.4: Distribution function at the entrance of the sheath RK2-CD

To better understand these oscillations, we examine the solution f_i at the entrance of the sheath Fig-6.4. From these figures, it can be observed that these oscillations are well-known and are due to the inability of the low-order central difference scheme to capture very steep slopes, as is used in this solver.

6.3 Numerical Modelling: WENO solver over $v_{\|}$

To resolve the oscillations found in the simulation section 6.2, this section repeats the simulations with a much more advanced solver using the WENO solver. By utilizing the solver introduced in 3.3.2, we can improve our ability to resolve very steep slopes in the v direction. In fact, the WENO solver was initially introduced in this thesis to address this specific issue in this experiment. The initial PDE is split as in eq-6.2, but the steps are modified as follows:

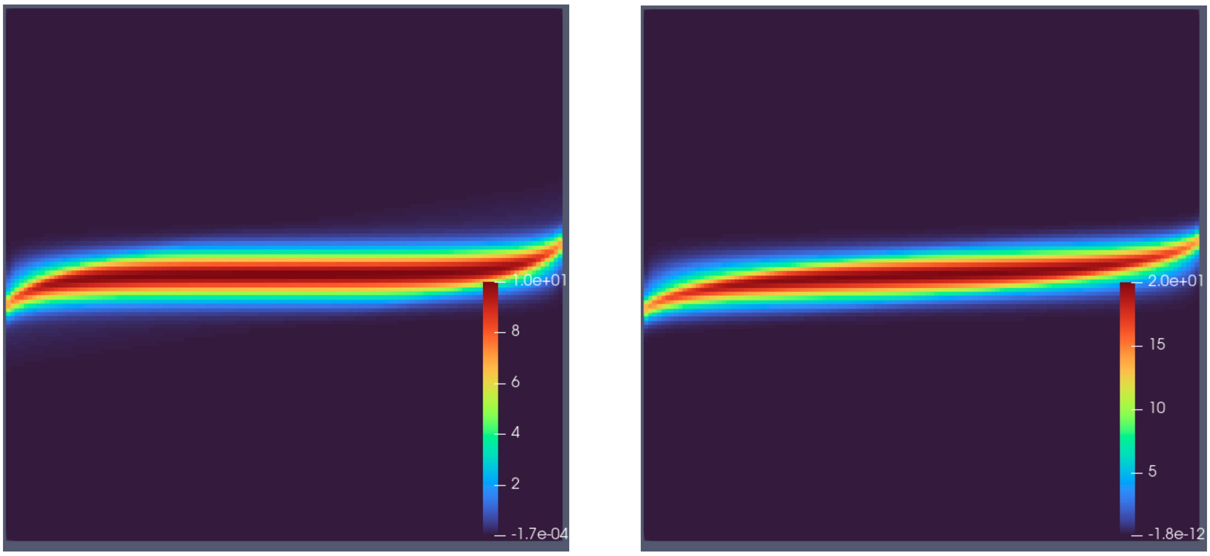
- Compute half time step in the z direction ($L \equiv -v_{\|}\partial_z$) LW-NISL
- Compute a full time step in the v direction ($N \equiv -E\partial_{v_{\|}}$) WENO-NISL
- Compute half timestep in the z direction ($L \equiv -v_{\|}\partial_z$) LW-NISL

6.3.1 Simulation results: not respected CFL

In this simulation (table 6.2) with the NISL-RK2-WENO solver, we use the same initial conditions and parameters as the simulation conducted in the subsection 6.2.1 to enable a comparison of the results.

Parameter Name	Value	Description
Model	$\partial_t f + v_{\parallel}\partial_z f + E\partial_{v_{\parallel}} f = 0$	1D-1V Drift Kinetic with source
Boundary conditions	$f(z = -2\pi, v_{\parallel} > 0, t) = 0$	Wall Boundary Conditions
Initial conditions	See refernce update me	Sheat Formation
Δt	0.1	Fixed Timestep for time discretization
N$_{x,v}$	128	Fixed Spatial Discretization
CFL	6.0638	Courant–Friedrichs–Lewy $CFL = \frac{\Delta t}{\Delta x}$
Ω	[-$2\pi, 2\pi$]$x[-6,6]$	Phase-Space Domain
t	[0, 10]	Starting Time Domain
Solver	LW-NISL / NISL-RK2-WENO	Non interpolating Semi Lagrangian Solver

Table 6.2: Table with simulation parameters for Sheath Formation model with not respected CFL and WENO solver



(a) Simulation at normalized time t=5.0s

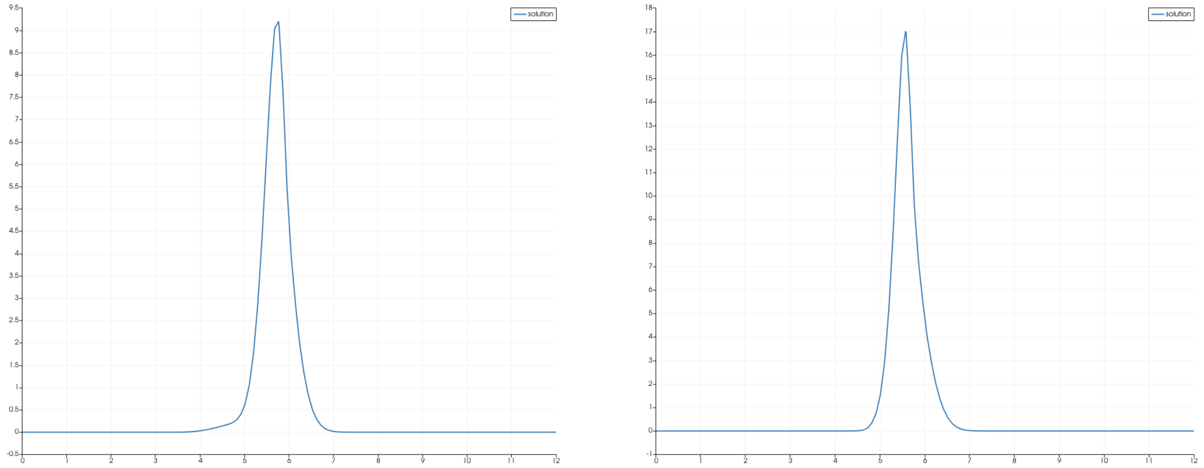
(b) Simulation at normalized time t=10.0s

Figure 6.5: Simulation results NISL-WENO-RK2

As can be observed from the 1D-1V space simulation Fig-6.5, the oscillations noted near the sheath entrance in Fig-6.3 are absent. Furthermore, by presenting the same graphs of f_i at the entrance of the sheath as in Fig-6.4, this time we obtain Fig-6.7. The profile of the ion distribution function is very smooth and without any oscillations, demonstrating the great effectiveness of the WENO discretization in eliminating this type of oscillation.

Having obtained a solution without non-physical oscillations, we can analyze the physics of the obtained results. In particular, for this type of experiment, the quantities of greatest interest are the electric field E and the electric potential ϕ. With the imposition of wall boundary conditions, as expected from the theory [38], we obtain a diverging electric field and an electric potential that follows a $\sqrt{x - x_{wall}}, \quad x \to x_{wall}$ trend.

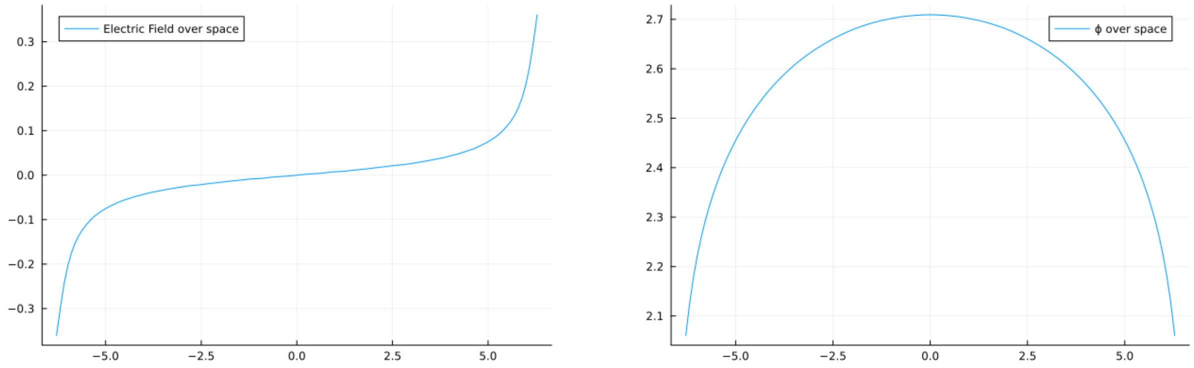
These results are also in line with those obtained in the report [35]. The main difference lies in the fact that the discretization used in this solver along the x-dimension is uniform, which implies that the diverging electric field will tend to assume lower values compared to those obtained in the report [35], where a greater number of points near the walls are used.



(a) Distribution function at the sheath entrance $f(z = -2\pi, v_{\parallel}, t)$ at t = 10s

(b) Distribution function at the sheath entrance $f(z = -2\pi, v_{\parallel}, t)$ at t = 30s

Figure 6.6: Distribution function at the entrance of the sheath WENO-RK



(a) Electric field over the space at t = 10s

(b) Potential electric field over the space at t = 10s

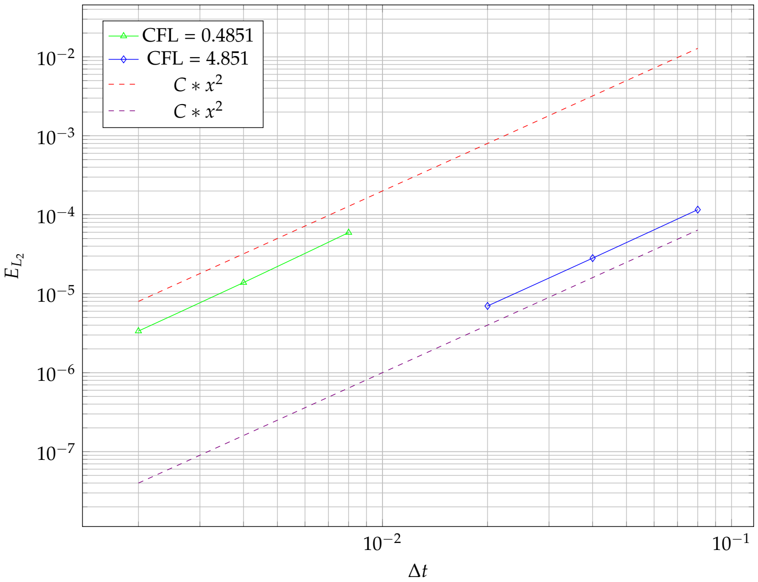
Figure 6.7: Electric and Potential fields for NISL-WENO-RK2

6.4 Manufactured Solution

To validate the correct implementation of the NISL-WENO-RK2 solver, this chapter proposes a test with a manufactured solution. Using the initial condition reported in the equation 6.6, we can use it as a solution in the PDE 6.1. In this way, we obtain a source term to add to our model that will give us our manufactured solution. To calculate the source term, the highly efficient Symbolics.jl framework [40] has been used in addition to the ion density equation6.11 calculated in [35].

$$n_i = an_+(z)\left(\frac{1}{2}+\frac{z}{L_z}\right) + an_-(z)\left(\frac{1}{2}-\frac{z}{L_z}\right) + n_0(z)\left(\frac{1}{2}+\frac{z}{L_z}\right)\left(\frac{1}{2}-\frac{z}{L_z}\right) \tag{6.11}$$

In the experiments conducted, as shown in Fig-6.4, it was verified that the NISL-WENO-RK2 solver converges with second order both with respected and non-respected CFL. Moreover, it is notable that for this manufactured solution, using higher CFL values, it was possible to achieve a good error relative to the size of the timestep Δt.



6.5 Bohm-Chodura condition

The concept of sheath conditions was first clearly interpreted by Bohm in 1949 [41], a concept now known as Bohm's criterion. This criterion specifies that ions can enter the Debye sheath only if their speed is at or above a certain threshold, known as Bohm's velocity. The criterion sheaths was then contributed also by Chodura [36], who demonstrated that ions enter the magnetic presheath at Bohm velocity along the magnetic field line.

For a stable sheath to exist, the kinetic Bohm condition, must be satisfied [42][35] [43]. This condition is crucial for understanding the dynamics and stability of sheaths in various plasma environments. In this experiment we can use this condition in order to understand the quality of the physics and possible improvements to the solver.

$$\frac{1}{2\pi}\int_{\Omega}\frac{f_i(z=z_{wall})}{v_{\parallel}^2}dv_{\parallel} \leq n_i(z=z_{wall}) \tag{6.12}$$

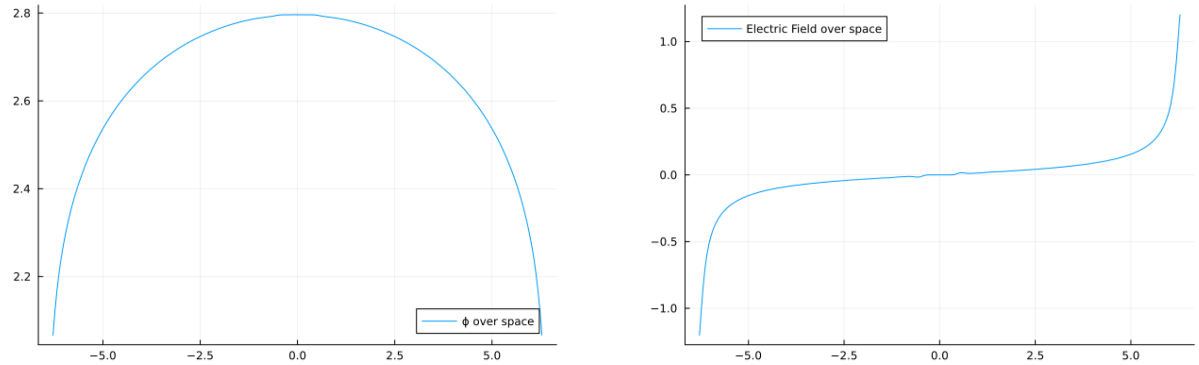
Specifically, in the results we present, we will focus solely on the fraction of the LHS and RHS, referred to as β.

6.5.1 Numerical results

β	Δt	N_x
1.813	0.16	64
1.623	0.08	128
1.576	0.04	256

Table 6.3: Bohm condition results

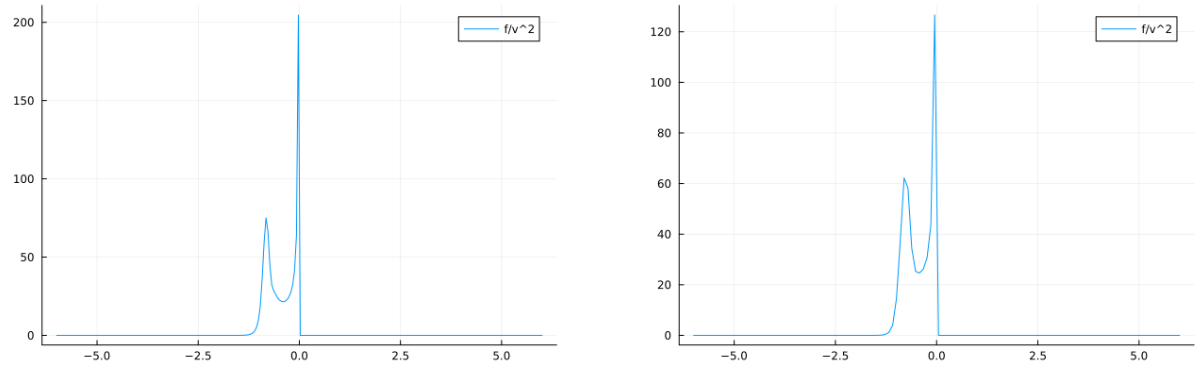
Particularly from the table, it can be observed that the Bohm condition is not satisfied (table 6.3). However, it is noticeable that the situation tends to improve with increasing resolution. To understand the reason for this non-compliance, we present in Fig-6.9 the profile of the quantity f_i/v^2 for $z = z_{wall}$. There is a very steep spike at values of v close to zero. As the resolution increases, the spike becomes thinner and higher. However, as can be seen from Fig-6.8, the solution remains valid concerning the approximation of the electric field and the potential.



(a) Electric field over the space at t = 10s

(b) Potential electric field over the space at t = 10s

Figure 6.8: Electric and Potential fields for the Bohm condition test at 256x256



(a) f/v^2 at the wall for 256x256

(b) f/v^2 at the wall for 128x128

Figure 6.9: Visualization of the quantity f/v^2 at the wall

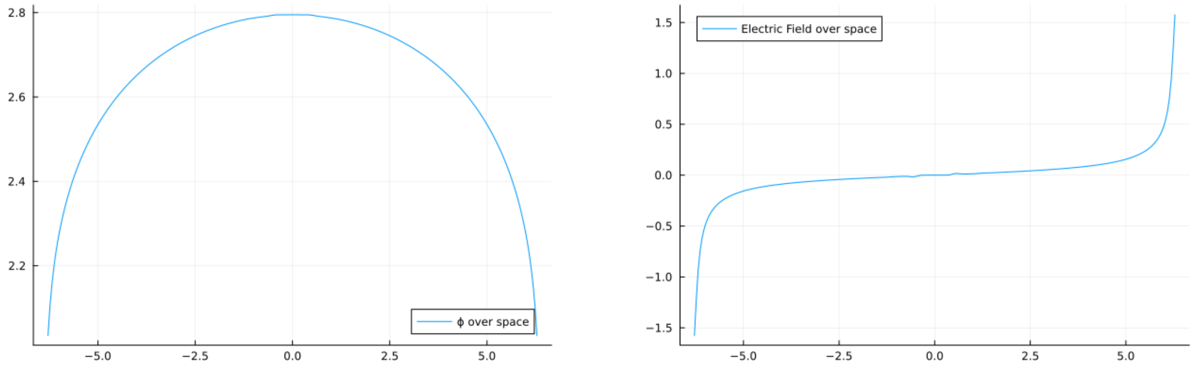
A numerical explanation for this spike is as follows. For the quantity f/v^2 to tend towards low values, it must be that as $v \to 0$, f also approaches zero, and specifically, the distribution should tend towards zero more rapidly to yield small values. In our numerical solver, we can simplify our error to the order of $O(\Delta v^2)$. However, it should be noted that $O(\Delta t)$ is greater due to the CFL condition.

Considering the best-case scenario, the quantity f/v^2 near the zero points of v takes the form $O(\Delta v^2)/\Delta v^2$, assuming the distribution tends towards zero at those points. This form implies that for the distribution function to decrease faster than the grid resolution, the solution at those points must converge to at least the third order. This possible explanation aligns with the observation that reducing Δv in the simulation results in thinner spikes with higher values.

A possible solution to this problem, besides increasing the convergence order of the solver, is to add an ϵ to the wall boundary conditions: $f_i(z, v_\parallel > \epsilon) = 0$ and $f_i(z, v_\parallel < \epsilon) = 0$. This approach avoids points too close to zero to numerically improve the evaluation of f_i/v^2. By doing so, we obtain results that respect the Bohm condition 6.4. In particular, from the profiles of f_i/v^2 in Fig-6.11, it is evident that the convergence improves significantly with an $\epsilon = 0.098$, which is equivalent to 1 Δx in the 128x128 simulation and 2 Δx in the 256x256 simulation. For completeness, the electric field and the potential of the 256x256 simulation with the ϵ correction are also shown Fig-6.10.

β	Δt	N_x
1.042	0.08	128
0.9741	0.04	256

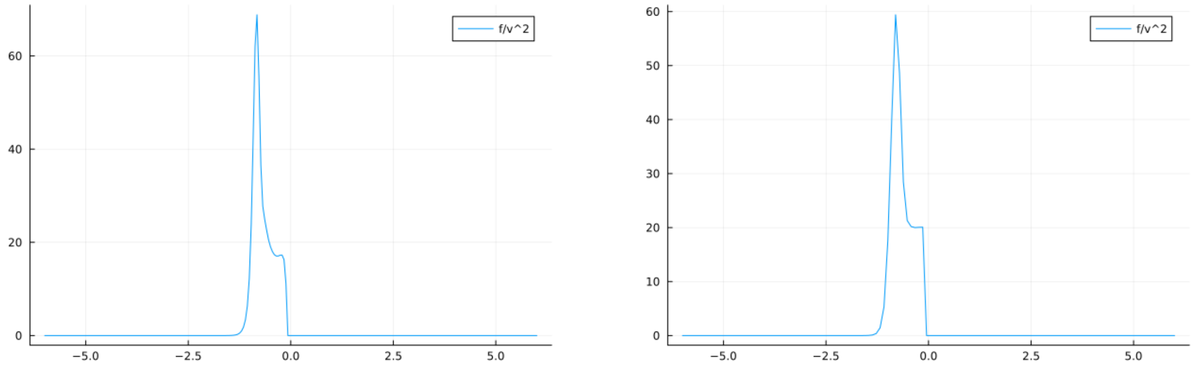
Table 6.4: Bohm condition results with ϵ shift



(a) Electric field over the space at t = 10s

(b) Potential electric field over the space at t = 10s

Figure 6.10: Electric and Potential fields for the Bohm condition test at 256x256



(a) f_i/v^2 at the wall for 256x256

(b) f_i/v^2 at the wall for 128x128

Figure 6.11: Visualization of the quantity f/v^2 at the wall with the ϵ shift

7 Conclusions

In this thesis, several numerical schemes based on the NISL approach for solving hyperbolic PDE have been analyzed. Starting from the schemes proposed by Rančić's and Ritchie in the second chapter, Runge Kutta and WENO schemes within the NISL framework were introduced. The third chapter presents an in-depth study in the case of a simple advection model, revealing interesting results about convergence and performance. It demonstrates that NISL methods converge correctly and perform better than SL methods with global interpolation, moreover, the potential better stability at higher Δt is suggested by the experiment 3.2.

In the fifth chapter, experiments confirmed the excellent results of the NISL-RK-CD method for solving the Vlasov-Poisson model and its ability to resolve the electric field over a very long period, considering the accuracy of spatial discretization. The final chapter, with the Sheath Formation experiment and the presence of a diverging electric field near the boundaries, verified the compatibility and effectiveness of WENO methods in the NISL formulation with RK integrators.

The results obtained in this thesis are very promising, particularly the final solver proposed with RK-WENO, which acquires the robustness of Eulerian components while removing the CFL condition thanks to its semi-Lagrangian component. This formulation can be easily scaled to higher dimensions, leveraging the excellent performance of the finite-difference-based WENO scheme. A key feature of these solvers is that once the advection velocity is split and the p trajectory correction applied, they implement similarly to their Eulerian counterparts. This implies that parallelizing these algorithms is significantly straightforward and similar to that of FD methods.

It would be feasible to extend the formulation obtained for FD models to solvers based on Finite Volume and Finite Element methods. Among the various potential improvements opened in this thesis is certainly the expansion of the last solver to use WENO discretization also in the x direction. This improvement would allow for greater detail and higher convergence in areas near the walls and would likely improve the Bohm condition. Another interesting topic would be to demonstrate that NISL time integrators based on RK can actually converge at orders higher than the second. In fact, in the experiment 3.7, the almost total removal of time error due to RK4 was observed, but this is not enough to demonstrate its fourth-order convergence in general. Among the most interesting open topics is the application of this formulation to parabolic PDE models.

List of Figures

List of Tables

Acronyms

Bibliography

[1] A. Robert. "A stable numerical integration scheme for the primitive meteorological equations". In: *Atmosphere-Ocean* 19.1 (Mar. 1981). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07055900.1981.9649098, pp. 35–46. ISSN: 0705-5900. DOI: 10.1080/07055900.1981.9649098. URL: https://doi.org/10.1080/07055900.1981.9649098 (visited on 04/24/2023).

[2] A. Staniforth and J. Côté. "Semi-Lagrangian Integration Schemes for Atmospheric Models—A Review". EN. In: *Monthly Weather Review* 119.9 (Sept. 1991). Publisher: American Meteorological Society Section: Monthly Weather Review, pp. 2206–2223. ISSN: 1520-0493, 0027-0644. DOI: 10.1175/1520-0493(1991)119<2206:SLISFA>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/mwre/119/9/1520-0493_1991_119_2206_slisfa_2_0_co_2.xml (visited on 04/17/2023).

[3] W. Barsukow. *Introduction to Conservation Laws: Theory and Numerics*. Online. 2022. URL: http://user.math.uzh.ch/barsukow/exercises2122/conservationlaws.pdf.

[4] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. crc Press, 2021.

[5] J. M. Burgers. "A mathematical model illustrating the theory of turbulence". In: *Advances in applied mechanics* 1 (1948), pp. 171–199.

[6] R. J. LeVeque and R. J. Leveque. *Numerical methods for conservation laws*. Vol. 214. Springer, 1992.

[7] H. Ritchie. "Eliminating the Interpolation Associated with the Semi-Lagrangian Scheme". In: *Monthly Weather Review* 114.1 (1986), pp. 135–146. DOI: https://doi.org/10.1175/1520-0493(1986)114<0135:ETIAWT>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/mwre/114/1/1520-0493_1986_114_0135_etiawt_2_0_co_2.xml.

[8] M. Rančić and G. Sindjić. "Noninterpolating Semi-Lagrangian Advection Scheme with Minimized Dissipation and Dispersion Errors". EN. In: *Monthly Weather Review* 117.8 (Aug. 1989). Publisher: American Meteorological Society Section: Monthly Weather Review, pp. 1906–1911. ISSN: 1520-0493, 0027-0644. DOI: 10.1175/1520-0493(1989)117<1906:NSLASW>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/mwre/117/8/1520-0493_1989_117_1906_nslasw_2_0_co_2.xml (visited on 04/17/2023).

[9] G. S. Dietachmayer. "Comments on "Noninterpolating Semi-Lagrangian Advection Scheme with Minimized Dissipation and Dispersion Errors"". EN. In: *Monthly Weather Review* 118.10 (Oct. 1990). Publisher: American Meteorological Society Section: Monthly Weather Review, pp. 2252–2253. ISSN: 1520-0493, 0027-0644. DOI: 10.1175/1520-0493(1990)118<2252:COSLAS>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/mwre/118/10/1520-0493_1990_118_2252_coslas_2_0_co_2.xml (visited on 05/05/2023).

[10] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

[11] S. Gottlieb and C.-W. Shu. "Total variation diminishing Runge-Kutta schemes". In: *Mathematics of computation* 67.221 (1998), pp. 73–85.

[12] W. Hundsdorfer and J. G. Verwer. "A note on splitting errors for advection-reaction equations". In: *Applied Numerical Mathematics* 18.1-3 (1995), pp. 191–199.

[13] X.-D. Liu, S. Osher, and T. Chan. "Weighted essentially non-oscillatory schemes". In: *Journal of computational physics* 115.1 (1994), pp. 200–212.

[14] G.-S. Jiang and C.-W. Shu. "Efficient implementation of weighted ENO schemes". In: *Journal of computational physics* 126.1 (1996), pp. 202–228.

[15] C.-W. Shu. "High order weighted essentially nonoscillatory schemes for convection dominated problems". In: *SIAM review* 51.1 (2009), pp. 82–126.

[16] L. Bonaventura and R. Ferretti. "Semi-Lagrangian methods for parabolic problems in divergence form". In: *SIAM Journal on Scientific Computing* 36.5 (2014), A2458–A2477.

[17] P. J. Roache. "Code verification by the method of manufactured solutions". In: *J. Fluids Eng.* 124.1 (2002), pp. 4–10.

[18] E. Horst and H. Neunzert. "On the classical solutions of the initial value problem for the unmodified non-linear Vlasov equation I general theory". In: *Mathematical Methods in the Applied Sciences* 3.1 (1981), pp. 229–248. DOI: https://doi.org/10.1002/mma.1670030117. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mma.1670030117. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mma.1670030117.

[19] K. Pfaffelmoser. "Global classical solutions of the Vlasov-Poisson system in three dimensions for general initial data". In: *Journal of Differential Equations* 95.2 (1992), pp. 281–303.

[20] J.-M. Qiu and A. Christlieb. "A conservative high order semi-Lagrangian WENO method for the Vlasov equation". en. In: *Journal of Computational Physics* 229.4 (Feb. 2010), pp. 1130–1149. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2009.10.016. URL: https://www.sciencedirect.com/science/article/pii/S0021999109005610 (visited on 08/13/2023).

[21] F. Filbet and E. Sonnendrücker. "Comparison of Eulerian Vlasov solvers". In: *Computer Physics Communications* 150.3 (Feb. 2003), pp. 247–266. ISSN: 0010-4655. DOI: 10.1016/S0010-4655(02)00694-X. URL: https://www.sciencedirect.com/science/article/pii/S001046550200694X (visited on 08/27/2023).

[22] D. Yi and S. Bu. "A mass conservative scheme for solving the Vlasov–Poisson equation using characteristic curve". In: *Journal of Computational and Applied Mathematics* 324 (Nov. 2017), pp. 1–16. ISSN: 0377-0427. DOI: 10.1016/j.cam.2017.04.019. URL: https://www.sciencedirect.com/science/article/pii/S0377042717301899 (visited on 08/15/2023).

[23] C. Cheng and G. Knorr. "The integration of the vlasov equation in configuration space". en. In: *Journal of Computational Physics* 22.3 (Nov. 1976), pp. 330–351. ISSN: 00219991. DOI: 10.1016/0021-9991(76)90053-X. URL: https://linkinghub.elsevier.com/retrieve/pii/002199917690053X (visited on 09/19/2023).

[24] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo. "The semi-Lagrangian method for the numerical resolution of the Vlasov equation". In: *Journal of computational physics* 149.2 (1999), pp. 201–220.

[25] G. Strang. "On the construction and comparison of difference schemes". In: *SIAM journal on numerical analysis* 5.3 (1968), pp. 506–517.

[26] M. Suzuki. "Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations". In: *Physics Letters A* 146.6 (1990), pp. 319–323.

[27] L. D. Landau. *Collected papers of LD Landau*. Pergamon, 1965.

[28] *Landau damping*. Accessed January 8, 2024. 2024. URL: https://en.wikipedia.org/wiki/Landau_damping.

[29] *H2020-Euratom-1.9 Program*. Accessed: 2024-01-12. 2024. URL: https://cordis.europa.eu/programme/id/H2020-Euratom-1.9.

[30] F. Militello. *Boundary Plasma Physics: An Accessible Guide to Transport, Detachment, and Divertor Design*. Vol. 123. Springer Nature, 2022.

[31] H. Meyer, T. Eich, M. Beurskens, et al. "Overview of progress in European medium sized tokamaks towards an integrated plasma-edge/wall solutiona". In: *Nuclear Fusion* 57.10 (June 2017), p. 102014. DOI: 10.1088/1741-4326/aa6084. URL: https://dx.doi.org/10.1088/1741-4326/aa6084.

[32] K.-U. Riemann. "The Bohm criterion and sheath formation". In: *Journal of Physics D: Applied Physics* 24.4 (1991), p. 493.

[33] N. Mellet, B. Pégourié, C. Martin, J. Gunn, H. Bufferand, and P. Roubin. "Magnetic sheath effect on the gross and net erosion rates due to impurities". In: *Physica Scripta* 2016.T167 (2016), p. 014064.

[34] V. Grandgirard. "The GYSELA project: A semi-Lagrangian code addressing gyrokinetic full-f global simulations of flux driven tokamak plasmas". PhD thesis. Université de Strasbourg, IRMA, 2016.

[35] M. R. Hardman, J. Omotani, M. Barnes, S. L. Newton, and F. I. Parra. *Convergence of the solutions of the 2D-1V ion drift kinetic equation with wall boundary conditions: manufactured solutions tests*. Excalibur-Neptune Documents. Accessed: 2024-01-12. 2023. URL: https://excalibur-neptune.github.io/Documents/MomentKinetics_Oxford-2_TN-04_ConvergenceSolutionsD1VIonDriftkineticEquationWallBoundaryConditions.html.

[36] R. Chodura. "Plasma–wall transition in an oblique magnetic field". In: *The Physics of Fluids* 25.9 (1982), pp. 1628–1633.

[37] A. Geraldini. *Kinetic treatment of magnetized and collisionless plasma near a wall*. 2018. arXiv: 1811.09901 [physics.plasm-ph].

[38] S. Parker, R. Procassini, C. Birdsall, and B. Cohen. "A suitable boundary condition for bounded plasma simulation without sheath resolution". In: *Journal of Computational Physics* 104.1 (1993), pp. 41–49.

[39] F. I. Parra, M. Barnes, and M. Hardman. *1D Drift Kinetic Models with Wall Boundary Conditions*. Excalibur-Neptune Documents. Accessed: 2024-01-12. 2021. URL: https://excalibur-neptune.github.io/Documents/MomentKinetics_Oxford_TN-05_1DDriftKineticWallBoundaryConditions.html.

[40] S. Gowda, Y. Ma, A. Cheli, M. Gwozdz, V. B. Shah, A. Edelman, and C. Rackauckas. "High-performance symbolic-numerics via multiple dispatch". In: *arXiv preprint arXiv:2105.03949* (2021).

[41] D. Bohm. "The characteristics of electrical discharges in magnetic fields". In: *Qualitative Description of the Arc Plasma in a Magnetic Field* (1949).

[42] K.-U. Riemann. “The Bohm criterion and sheath formation”. en. In: *Journal of Physics D: Applied Physics* 24.4 (Apr. 1991), p. 493. ISSN: 0022-3727. DOI: 10.1088/0022-3727/24/4/001. URL: https://dx.doi.org/10.1088/0022-3727/24/4/001 (visited on 09/16/2023).

[43] M. Barnes, F. I. Parra, M. R. Hardman, and J. Omotani. *Numerical study of 1+1D drift kinetic model with wall boundary conditions*. Tech. rep. United Kingdom: Rudolf Peierls Centre for Theoretical Physics, University of Oxford; Culham Centre for Fusion Energy, 2021. URL: https://excalibur-neptune.github.io/Documents/MomentKinetics_Oxford_TN-08_Numerical11DDriftKineticModelWallBoundaryConditions.html.