A FAST SOLVER FOR IMPLICIT INTEGRATION OF THE VLASOV-POISSON SYSTEM IN THE EULERIAN FRAMEWORK *

C. KRISTOPHER GARRETT† AND CORY D. HAUCK‡

Abstract. We present a domain decomposition algorithm to accelerate the solution of Eulerian-type discretizations of the linear, steady-state Vlasov equation. The steady-state solver then forms a key component in the implementation of fully implicit or nearly fully implicit temporal integrators for the nonlinear Vlasov-Poisson (VP) system. The solver relies on a particular decomposition of phase space that enables the use of sweeping techniques commonly used in radiation transport applications. The original linear system for the phase space unknowns is then replaced by a smaller linear system involving only unknowns on the boundary between subdomains, which can then be solved efficiently with Krylov methods such as GMRES. Steady state solves are combined to form an implicit Runge-Kutta time integrator, and the Vlasov equation is coupled self-consistently to the Poisson equation via a linearized procedure or a nonlinear fixed point method for the electric field. Numerical results for standard test problems demonstrate the efficiency of the domain decomposition approach when compared to the direct application of an iterative solver to the original linear system.

1. Introduction. The Vlasov Poisson (VP) system describes the evolution of a collisionless plasma in the electrostatic limit. For electrons embedded in a uniformly distributed ion background, the equations are

\[
\begin{align*}
\partial_t f + \mathbf{v} \cdot \nabla_x f + \frac{e}{m} \mathbf{E} \cdot \nabla_v f &= 0, \\
\Delta_x \Phi &= \frac{e}{\epsilon_0} (\rho - \bar{\rho}), \\
\mathbf{E} &= \nabla_x \Phi, \\
\rho &= \int_{\mathbb{R}^d} f \, dv,
\end{align*}
\]

along with appropriate boundary and initial conditions. Here \( f(x, v, t) \) is the number density of electrons with respect to the measure \( dv \, dx \); the independent variables \( x \in D_x \subset \mathbb{R}^d \), \( v \in \mathbb{R}^d \), and \( t \) represent space, velocity, and time, respectively. For real physical systems, \( d = 3 \); but in many test cases, \( d = 1 \) or \( d = 2 \). The positive constants \( e, m, \) and \( \epsilon_0 \) represent the elementary unit of charge, the electron mass, and the permeability of free space, respectively. The function \( \mathbf{E} = \mathbf{E}(x, t) \) is the electric field, and \( \Phi = \Phi(x, t) \) is the electric potential. The quantity \( \bar{\rho} \) is the background ion density, which is assumed to be constant with respect to \( x, v, \) and \( t \). We assume further that \( D_x \) is bounded and that the initial and boundary conditions for (1.1) are such that the system is globally charge neutral; that is,

\[
\bar{\rho} = \frac{1}{|D_x|} \int_{D_x} \rho \, dx.
\]

Following common convention, we will for the remainder of the paper scale the Vlasov-Poisson system to remove the constants that represent the charge, mass, and permeability of free space.

* This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

†This information has been authored by an employee or employees of Los Alamos National Security, operator of LANL, under contract DE-AC52-06NA25396 with the U.S. Department of Energy.

‡Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. De-AC05-00OR22725.
The cost of solving the Vlasov-Poisson system numerically comes primarily from the Vlasov equation for \( f \). For a general overview, see [17] and also [41] for a presentation with updated references. There are three basic approaches to discretizing the VP system: Lagrangian, Eulerian, and semi-Lagrangian. Lagrangian methods, such as particle-in-cell (PIC) [19,29], take advantage of the fact that \( f \) is constant along the characteristics

\[
\dot{X}(t) = V(t), \quad \dot{V}(t) = E(X(t), t).
\]  

These methods use representative shape functions (“particles”) that follow the characteristics. In this way, they are able to preserve sharp features of the flow, but may suffer from a lack of uniform accuracy, unless an adaptive procedure is used to divide and merge particles [30,35,43,46,49]. Eulerian methods [8,15,18,24,26], on the other hand, solve the kinetic equation on a mesh or grid that does not move, but may be coarsened or refined to capture local features [25]. Although Eulerian methods do not follow characteristics, they often rely on local characteristic information to define numerical traces for functions that are not continuous at cell interfaces. Semi-Lagrangian methods [12–14,36,37,42] strike a balance between the previous two: they evolve unknowns for a given time along characteristics and then project the solution back down to a solution space that is tied to a fixed grid.

In multi-scale settings, it may be desirable to numerically solve the Vlasov-Poisson system in a fully (or at least partially) implicit fashion. This may be to step over time scales associated with the electric field or high-velocity parts of the kinetic distribution or, for multi-species calculations, to step over the time scales of faster moving particles. Problems that are source or boundary driven may also require long-time integration to approach steady-state. Implicit methods can also be beneficial when additional physical processes, such as strong magnetic fields or stiff collision operators, are added to the system.

For a static electric field, an implicit solution to the Vlasov equation with a Lagrangian method amounts to finding an implicit solution of the characteristic ODEs in (1.3). However, for an Eulerian method, the situation is more difficult. In general, an implicit time stepping algorithm will require the solution of a large linear system of equations. In the case of zero electric field, this system of equations can be solved easily when using piece-wise smooth discretizations, so long as one uses upwind fluxes to define numerical traces at cell interfaces, by sweeping through the mesh along the local characteristic direction. Indeed, sweeping algorithms form the basis of many modern radiation transport codes [1,31,33]. However, for a non-zero electric field, the characteristics become quite complicated, and circular dependencies can arise that make sweeping impossible.

Many Eulerian and semi-Lagrangian methods use operator splitting to separate the Vlasov equation into two components:

\[
\dot{c}_1 f_1 + v \cdot \nabla_x f_1 = 0 \quad \text{and} \quad \dot{c}_2 f_2 + E \cdot \nabla_v f_2 = 0.
\]  

The characteristics for \( f_1 \) and \( f_2 \) are trivial, the latter because \( E \) does not depend on \( v \). This splitting introduces temporal errors that can be reduced by combining solutions of (1.4) with different initial conditions. For example, a second-order Strang splitting for semi-Lagrangian methods was initially proposed in [12], while higher-order time splitting schemes were studied more recently in [38]. However, these higher-order schemes require multiple solutions of the equations and may suffer from order reduction in multi-scale settings.
In the current paper, we propose a Schur complement, domain decomposition method \([44]\) with a specific decomposition that significantly increases the efficiency of implicit time integrators for a wide variety of Eulerian-type discretizations of the Vlasov-Poisson system. Specifically, we decompose the position-velocity phase space into \(2^d\) subdomains upon which the components of \(\mathbf{v}\) do not change sign. This decomposition breaks cyclic dependencies, thereby enabling the use of standard sweeping techniques on each subdomain. The initial linear system can then be reduced to a smaller system for the unknowns on the interfaces between subdomains.

We employ the domain decomposition technique as part of two different, known strategies for solving the Vlasov-Poisson system. The first strategy is to linearize the system by approximating \(E\) at later times by combining a Taylor series approximation with information from the equations \([12, 38]\). The second strategy is to solve the fully implicit, nonlinear system by formulating it as a fixed point problem for the electric field, as was done in \([11]\) in the context of PIC methods. This problem may be solved using, for instance, the Jacobian-Free Newton-Krylov (JFNK) method \([27]\) or Anderson acceleration \([2, 47]\). The advantage of solving for \(E\) instead of \(f\) is that the former depends only on \(x\) and \(t\). The result is that, at the discrete level, one arrives at a set of algebraic equations with far fewer unknowns.

2. Vlasov Discretization. The discretizations used in this paper are standard. We focus on an illustrative example, based on a second-order, method-of-lines approach. The discontinuous Galerkin method with linear elements is used for the discretization of position and velocity and a second-order implicit Runge-Kutta method for discretization of time. While this choice is made for simplicity and convenience, higher-order versions are also possible; the main difficulty in this respect is establishing a proper notion of upwinding when components of \(E\) change sign in a computational cell.

2.1. Time Integration. The temporal domain \([0, T]\) is divided into intervals with constant time step \(\Delta t\), and we use the notation \(f^n \approx f(\cdot, \cdot, t^n)\), where \(t^n = n\Delta t\). We employ the second order, singly diagonally implicit Runge-Kutta (SDIRK) method \([22, 23]\) with Butcher tableau

\[
\begin{array}{c|ccc}
  \gamma & 1 - \gamma & \gamma \\
  1 - \gamma & \gamma
\end{array}
\]

(2.1)

When applied to the Vlasov equation (1.1a), this method takes the form

\[
f^{(k)} = f^n - \Delta t \sum_{\ell=1}^k A_{k\ell} \left( \mathbf{v} \cdot \nabla_x f^{(\ell)} + E^{(\ell)} \cdot \nabla_v f^{(\ell)} \right), \quad k = 1, 2, \quad (2.2a)
\]

\[
f^{n+1} = f^n - \Delta t \sum_{k=1}^2 b_k \left( \mathbf{v} \cdot \nabla_x f^{(k)} + E^{(k)} \cdot \nabla_v f^{(k)} \right). \quad (2.2b)
\]

Here \(E^{(k)}\) is the electric field term that is either (i) determined by linearization using data from time \(t^n\) or (ii) coupled self-consistently via the Poisson equation. These two options are discussed later in Section 4.

The stages in (2.2a) can be rewritten in the following steady-state form

\[
\mathbf{v} \cdot \nabla_x f^{(k)} + E^{(k)} \cdot \nabla_v f^{(k)} + \lambda^{(k)} f^{(k)} = q^{(k)}.
\]

(2.3)
where
\[
q^{(k)} = \frac{1}{A_{kk}} \left( \frac{f^n}{\Delta t} - \sum_{\ell=1}^{k-1} A_{kl} \left( \mathbf{v} \cdot \nabla x f^{(\ell)} + \mathbf{E}^{(\ell)} \cdot \nabla v f^{(\ell)} \right) \right)
\] (2.4)
is a source depending on previous stages and \(\lambda^{(k)} = (A_{kk} \Delta t)^{-1}\) is an effective absorption constant. We drop the stage index \(k\) and focus on the linear, steady-state problem
\[
\mathbf{v} \cdot \nabla x f + \mathbf{E} \cdot \nabla v f + \lambda f = q,
\] (2.5)
where, in an abuse of notation, \(f = f(x, \mathbf{v})\) is the new steady-state unknown, \(q = q(x, \mathbf{v})\) is a general source, \(\lambda > 0\) is constant, and \(\mathbf{E} = \mathbf{E}(x)\) is assumed to be given.

**2.2. Phase Space Discretization.** We discuss in detail the case \(d = 1\), which is typically referred to as 1D-1V. Generalization to a positive integer \(d > 1\) is straightforward. We let \(D_x = (a_x, b_x) \subset \mathbb{R}\) be the spatial domain and truncate the velocity domain from all of \(\mathbb{R}\) to the finite interval \(D_v = (a_v, b_v)\). In this setting, the steady-state Vlasov equation (2.5) becomes
\[
v \partial_x f + E \partial_v f + \lambda f = q.
\] (2.6)
In the initial formulation of the domain decomposition strategy, we will use zero inflow boundary conditions for both variables:
\[
\begin{cases}
f(a_x, v) = 0, & v > 0, \\
f(b_x, v) = 0, & v < 0,
\end{cases}
\quad \text{and} \quad
\begin{cases}
f(x, a_v) = 0, & E(x) > 0, \\
f(x, b_v) = 0, & E(x) < 0.
\end{cases}
\] (2.7)
However, in the numerical examples, we will assume periodic boundary conditions in \(x\). Some discussion will be given about how to leverage the zero inflow formulation to more general boundary conditions, such as the periodic case.

The phase space \(D = D_x \times D_v\) is divided into an \(N_x \times N_v\) mesh of rectangular cells. For \(i = 1, \ldots, N_x\) and \(j = 1, \ldots, N_v\), the cells
\[
C_{i,j} = X_i \times V_j = (x_{i-1/2}, x_{i+1/2}) \times (v_{j-1/2}, v_{j+1/2})
\] (2.8)
have centers \((x_i, v_j)\), uniform side lengths \(\Delta x\) and \(\Delta v\), and cell edges
\[
e_{i+1/2,j} = x_{i+1/2} \times V_j \quad \text{and} \quad e_{i,j+1/2} = X_i \times v_{j+1/2}.
\] (2.9)
By convention, we associate with each edge a normal vector \(n = (n_x, n_v)^T\) whose components are always non-negative.

Let \(Z = \{g \in L^2(dx dv) : g|_{D^c} = 0 \text{ a.e.}\}\), and let \(Z^h = \{g^h \in Z : g^h|_{C_{i,j}} \in P^1(C_{i,j})\}\), where \(P^1(C_{i,j})\) is the space of polynomials on \(C_{i,j}\) of degree at most one. For any \(g \in Z^h\) and \((x, v) \in \partial C_{i,j}\), let the traces on positive and negative sides of the cell boundary be given by
\[
g^\pm(x, v) = \lim_{\epsilon \to 0^+} g(x \pm \epsilon n_x, v \pm \epsilon n_v).
\] (2.10)

Then define the numerical trace of \(g\) by upwinding:\footnote{The traces in (2.11) are not defined when \(v_j = 0\) or \(E_i = 0\), but they do not contribute to the flux in such cases.}
\[
\tilde{g}(x_{i+1/2}, v) = \begin{cases} 
g^-(x_{i+1/2}, v), & v_j > 0, \\
g^+(x_{i+1/2}, v), & v_j < 0,
\end{cases}
\quad \text{for all } v \in V_j
\] (2.11a)
\[
\hat{g}(x, v_{j+1/2}) = \begin{cases} 
    g^-(x, v_{j+1/2}), & E_i > 0, \\
    g^+(x, v_{j+1/2}), & E_i < 0,
\end{cases} \quad \text{for all } x \in X_i,
\] (2.11b)

where \( E_i = E(x_i) \). Let the trace average \( \langle g \rangle \) and jump \([g]\) be given by
\[
\langle g \rangle = \frac{1}{2}(g^+ + g^-) \quad \text{and} \quad [g] = g^+ - g^-.
\] (2.12)

Then, with (2.11), the upwind fluxes take the form
\[
\begin{align*}
    v_j \hat{g} &= v_j \langle g \rangle - \frac{1}{2}v_j [g] \quad \text{on } e_{i+1/2,j}, \\
    E_i \hat{g} &= E_i \langle g \rangle - \frac{1}{2}E_i [g] \quad \text{on } e_{i,j+1/2}.
\end{align*}
\] (2.13a) (2.13b)

### 2.2.1. Variational Formulation

We seek an approximate solution \( f^h \in Z^h \) of (2.6) which satisfies
\[
\begin{align*}
    -v_j \int_{C_{i,j}} f^h \partial_x g^h \, dx \, dv + v_j \int_{e_{i+1/2,j}} \hat{f}^h g^h_+ \, dv - v_j \int_{e_{i-1/2,j}} \hat{f}^h g^h_- \, dv \\
    -E_i \int_{C_{i,j}} f^h \partial_v g^h \, dx + E_i \int_{e_{i,j+1/2}} \hat{f}^h g^h_+ \, dx - E_i \int_{e_{i,j-1/2}} \hat{f}^h g^h_- \, dx \\
    + \lambda \iint_{C_{i,j}} f^h g^h \, dx \, dv = \iiint_{C_{i,j}} qg^h \, dx \, dv
\end{align*}
\] (2.14)

for all \( g^h \in Z^h \) and all cells \( C_{i,j} \subset D \). Here \( v \) and \( E \) are approximated by their pointwise values at the center of each cell in order to ensure that the upwind direction is constant along each cell edge,\(^2\) and the notation \( \hat{f}^h = f^h \) is used as a matter of convenience.

A global formulation for \( f^h \) is found by summing (2.14) over all cells. It takes the form
\[
A(f^h, g^h) = Q(g^h) \quad \forall g^h \in Z^h,
\] (2.15)

where the bilinear form \( A \) is given by
\[
A(f^h, g^h) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} \iint_{C_{i,j}} (-v_j f^h \partial_x g^h - E_i f^h \partial_v g^h + \lambda f^h g^h) \, dx \, dv
\]
\[
- \sum_{i=0}^{N_x} \sum_{j=1}^{N_x} v_j \hat{f}^h [g^h] \, dv - \sum_{i=1}^{N_x} \sum_{j=0}^{N_x} E_i \hat{f}^h [g^h] \, dx
\] (2.16)

and the linear source \( Q \) is
\[
Q(g^h) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} \iiint_{C_{i,j}} qg^h \, dx \, dv.
\] (2.17)

\(^2\)This approximation, while convenient, limits the accuracy of the discretization. For \( P^1 \) elements, we observe numerically that the approximation is still second-order. However, because the emphasis of this paper is on the solution procedure, we forgo a rigorous analysis. The interested reader is referred to [3, 15, 34] for convergence analysis of DG methods for the Vlasov-Poisson and related systems.
Using the flux definitions in (2.13), one can show that the bilinear form gives rise to an energy:

$$A(f^h, f^h) = \frac{1}{2} \sum_{i=0}^{N_x} \sum_{j=1}^{N_v} \int_{e_{i+1/2,j}} |v_j||f^h|^2 \, dv + \frac{1}{2} \sum_{i=1}^{N_x} \sum_{j=1}^{N_v} \int_{e_{i+1/2,j}} |E_i||f^h|^2 \, dx$$

$$+ \sum_{i=1}^{N_x} \sum_{j=1}^{N_v} \int_{C_{i,j}} \lambda |f^h|^2 \, dxdv. \quad (2.18)$$

This formula is useful for deriving stability estimates in the following section.

2.2.2. Linear Algebra Implementation. To place (2.14) into a linear algebra framework, we expand \( f^h \) and \( \hat{f}^h \). For each \( i \) and \( j \), let \( \{b^k_{i,j}\}_{k=0,1,2} \) form an orthonormal basis of \( P^1(C_{i,j}) \) and expand

$$f^h_{i,j}(x, v) = \sum_{k=0}^{2} b^k_{i,j}(x, v) b^k_{i,j}, \quad (2.19)$$

where \( b_{i,j} = (b_0^{i,j}, b_1^{i,j}, b_2^{i,j})^T \) and \( f_{i,j} = (f_0^{i,j}, f_1^{i,j}, f_2^{i,j})^T \). Furthermore, let \( \{\tilde{b}^k_{i+1/2,j}\}_{k=0,1} \) and \( \{\tilde{b}^k_{i,j+1/2}\}_{k=0,1} \) form orthonormal bases on \( P^1(e_{i+1/2,j}) \) and \( P^1(e_{i,j+1/2}) \), respectively. Then expand

$$\hat{f}^h_{i+1/2,j}(v) = \sum_{k=0}^{1} \tilde{b}^k_{i+1/2,j}(v) \tilde{f}^k_{i+1/2,j}, \quad (2.20a)$$

$$\hat{f}^h_{i,j+1/2}(x) = \sum_{k=0}^{1} \tilde{b}^k_{i,j+1/2}(x) \tilde{f}^k_{i,j+1/2}, \quad (2.20b)$$

where \( \tilde{b}_{i+1/2,j} = (\tilde{b}_0^{i+1/2,j}, \tilde{b}_1^{i+1/2,j})^T \), \( \tilde{f}_{i+1/2,j} = (\tilde{f}_0^{i+1/2,j}, \tilde{f}_1^{i+1/2,j})^T \), and \( \tilde{b}_{i,j+1/2} \) and \( \tilde{f}_{i,j+1/2} \) are similarly defined. Using (2.19) and (2.20), (2.14) may be rewritten as

$$M_{i,j} f_{i,j} + M_{i+1/2,j}^- \hat{f}_{i+1/2,j} - M_{i-1/2,j}^+ \hat{f}_{i-1/2,j}$$

$$+ M_{i,j+1/2}^+ \hat{f}_{i,j+1/2} - M_{i,j-1/2}^- \hat{f}_{i,j-1/2} = q_{i,j}, \quad (2.21)$$

where the local matrices \( M_{i,j}, M_{i+1/2,j}^\pm, \) and \( M_{i,j+1/2}^\pm \) are given in Appendix A, and the local source is given by

$$q_{i,j} = \int_{C_{i,j}} b_{i,j} q \, dxdv. \quad (2.22)$$

The equation in (2.21) expresses a balance between interior coefficients \( f_{i,j} \) in each cell volume and the trace coefficients \( \hat{f}_{i+1/2,j} \) and \( \hat{f}_{i,j+1/2} \) at cell edges. To see the upwind nature of the algorithm, we first define the indicator function

$$\chi_{\{S\}} = \begin{cases} 1, & \text{if the statement } S \text{ is true,} \\ 0, & \text{if the statement } S \text{ is false.} \end{cases} \quad (2.23)$$
Applying the trace definitions in (2.11) to (2.21) yields an equivalent linear system in terms of the interior coefficients only:

\[
\begin{align*}
\hat{M}_{i,j} f_{i,j} &+ \chi_{(v_j<0)} \hat{M}_{i+1/2,j}^- f_{i+1,j} + \chi_{(v_j>0)} \hat{M}_{i-1/2,j}^+ f_{i-1,j} \\
&+ \chi_{(E_i<0)} M_{i,j}^- f_{i,j+1} + \chi_{(E_i>0)} M_{i,j}^+ f_{i,j-1} = q_{i,j},
\end{align*}
\]

where, by convention, \( f_{i,0} = f_{i,N_x+1} = 0 \) for all \( i = 1, \ldots, N_x \) and \( f_{0,j} = f_{N_x+1,j} = 0 \) for all \( j = 1, \ldots, N_v \). The matrices \( \hat{M}_{i,j}, M_{i+1/2,j}^- \) and \( M_{i,j}^+ \) are given in Appendix A. The form of (2.24) shows how \( f_{i,j} \) is affected by neighboring cells, depending on the sign of \( v_j \) and \( E_i \).

3. Solving the Steady State Vlasov Problem. When using upwind fluxes, it is possible to decompose the phase-space domain \( \mathcal{D} \) into a small number of subdomains upon which (2.21) can be solved explicitly once all the inflow boundary conditions on the subdomain are known. In this approach, the traces on the boundary between the subdomains become the unknowns, resulting in a linear system that is smaller and easier to solve. To describe the approach in more detail, we continue with the 1D-1V formulation from the previous section and later extend the ideas to 2D-2V and 3D-3V settings.

To develop some intuition for the solution procedure, consider first the neutral-particle case, when \( E = 0 \) and (2.24) takes the form

\[
\begin{align*}
\hat{M}_{i,j} f_{i,j} &+ \chi_{(v_j<0)} \hat{M}_{i+1/2,j}^- f_{i+1,j} + \chi_{(v_j>0)} \hat{M}_{i-1/2,j}^+ f_{i-1,j} = q_{i,j},
\end{align*}
\]

where \( f_{N_x+1,j} = f_{0,j} = 0 \) by convention. This simplified system is decoupled in the index \( j \), so it is possible to solve for

\[
f_{j} := \{ f_{1,j}, \ldots, f_{N_x,j} \}
\]

independently for each \( j \). Furthermore, for each fixed \( j \), the sign of \( v_j \) is known so that one of the indicator functions in (3.1) vanishes. This makes it possible to solve for the components of \( f_{j} \) in an explicit, albeit serial, fashion—sweeping either from left to right when \( \chi_{(v_j>0)} = 1 \) or from right to left when \( \chi_{(v_j<0)} = 1 \). For multidimensional problems, a similar procedure is used: for each velocity multi-index \( j \), \( f_{j} \) can be determined independently; once \( j \) is fixed, the sign of each component of \( v_j \) is known, and it is possible to explicitly solve for the components of \( f_{j} \) by sweeping through the mesh in an appropriate fashion. Moreover, in the multidimensional case, there is additional parallelism that can be further exploited [4].

Difficulties can arise when applying the sweeping procedure in the charged-particle setting, when \( E \neq 0 \). Consider, for example, the dependencies generated by the two field profiles in Figure 3.1. For the profile in Figure 3.1a, it is possible to solve for

\[
f := \{ f_{ij} : i = 1, \ldots, N_x; j = 1, \ldots, N_v \}
\]

by first solving in the upper left and lower right quadrants to determine trace values on the axes and then using these values to solve in the other two quadrants. However, the field profile in Figure 3.1b creates a cyclical dependency in which the solution of \( f \) in every cell is dependent on the solution from another cell. In this case, it is not possible to explicitly solve for \( f \).

This convention will be employed throughout the paper.
lies with the fact that the electric field can change sign inside a cell. In this case, the higher order DG method in conjunction with the sweeping technique. The problem of explicitly for \( f \) where \( f \) is known, then it is possible to solve for \( c \) on \( \bar{M}_{i,j} \). This decomposition is based on the observation that if all the trace coefficients on \( \Gamma \) are known, then it is possible to solve for \( f \) in \( D^{(1)} \) and \( D^{(2)} \) explicitly and independently. For example, in \( D^{(1)} \), \( \chi_{\{v_j<0\}} = 0 \) and \( \chi_{\{v_j>0\}} = 1 \). In this case, (2.24) becomes (cf. Algorithm 1)

\[
\begin{align*}
\tilde{M}_{i,j} f_{i,j} + \tilde{M}_{i-1/2,j} f_{i-1,j} + \chi_{\{E_i<0\}} \tilde{M}_{i,j+1/2} f_{i,j+1} \\
+ \chi_{\{E_i>0\}} \tilde{M}_{i,j-1/2} f_{i,j-1} &= q_{i,j}, \quad j > j_0 + 1,
\end{align*}
\]

(3.5a)

\[
\begin{align*}
\tilde{M}_{i,j} f_{i,j} + \tilde{M}_{i-1/2,j} f_{i-1,j} + \chi_{\{E_i<0\}} \tilde{M}_{i,j+1/2} f_{i,j+1} \\
- \chi_{\{E_i>0\}} M_{i,j-1/2} \hat{f}_{i,j_0+1/2} &= q_{i,j}, \quad j = j_0 + 1.
\end{align*}
\]

(3.5b)

where \( f_{0,j} = 0 \) by convention and an asterisk has been added to the trace coefficient \( \hat{f}_{i,j_0+1/2} \) to indicate that it is a known quantity. These equations can now be solved explicitly for \( f_{i,j} \), starting from \( i = 1 \) and increasing to \( i = N_x \). For each \( i \), the sign of \( E_i \) is known and one can either sweep from \( j = j_0 + 1 \) to \( N_v \) if \( E_i > 0 \) or from \( j = N_v \) to \( j = j_0 + 1 \) if \( E_i < 0 \). For \( D^{(2)} \), the procedure is similar; see Algorithm 2.

Remark 1. At this point, one can see the main difficulty in implementing a higher order DG method in conjunction with the sweeping technique. The problem lies with the fact that the electric field can change sign inside a cell. In this case, the

![Fig. 3.1: Dependencies in phase space for two different electric field profiles. Solid lines indicate inflow boundaries and dotted lines indicate outflow boundaries. The decomposition of \( D \) into \( D^{(1)} \) and \( D^{(2)} \) removes cyclic dependencies. Dash-dotted lines indicate inflow boundaries from one subdomain to another.](image-url)
computation of $f$ in cells sharing an edge parallel to the x-axis both depend on each other, even when implementing an upwind numerical flux.

Algorithm 1: Sweep algorithm for $D^{(1)}$

Input: $\{q_{i,j}\}_{i=1,j=j_0}^{N_x,N_v}$, $\{\tilde{f}^h_{i,j_0+1/2} : E_i > 0\}$

Output: $\{f_{i,j}\}_{i=1,j=j_0}^{N_x,N_v}$

1) for $i = 1$ to $N_x$ do
2) if $E_i > 0$ then
3) Solve (in order) $f_{i,j_0+1}, f_{i,j_0+2}, \ldots, f_{i,N_v}$
4) end
5) if $E_i < 0$ then
6) Solve (in order) $f_{i,N_v}, f_{i,N_v-1}, \ldots, f_{i,j_0+1}$
7) end
8) end

Algorithm 2: Sweep algorithm for $D^{(2)}$

Input: $\{q_{i,j}\}_{i=1,j=j_0}^{N_x,N_v}$, $\{\tilde{f}^h_{i,j_0+1/2} : E_i < 0\}$

Output: $\{f_{i,j}\}_{i=1,j=j_0}^{N_x,N_v}$

1) for $i = N_x$ to 1 do
2) if $E_i > 0$ then
3) Solve (in order) $f_{i,1}, f_{i,2}, \ldots, f_{i,j_0}$
4) end
5) if $E_i < 0$ then
6) Solve (in order) $f_{i,j_0}, f_{i,j_0-1}, \ldots, f_{i,1}$
7) end
8) end

In the following subsections, we describe the domain decomposition approach in more detail. This includes both the variational formulation and the linear algebra implementation.

3.1.1. Variational Framework. Given any $g \in Z$, let

$$g_1 = \chi_{\{v>0\}}g \quad \text{and} \quad g_2 = \chi_{\{v<0\}}g$$  \hspace{1cm} (3.6)

and expand the formula in (2.16) for the bilinear form $A$:

$$A(f^h, g^h) = A(f^h_1, g^h_1) + A(f^h_2, g^h_2) + A(f^h_1, g^h_2) + A(f^h_2, g^h_1).$$  \hspace{1cm} (3.7)

A direct calculation shows that

$$A(f^h_1, g^h_1) = \sum_{i=1}^{N_x} \sum_{j=j_0}^{N_v} \int_{\mathcal{C}_{i,j}} (-v_j f^h \hat{\partial}_x g^h - E_i f^h \hat{\partial}_v g^h + \lambda f^h g^h) \, dx \, dv$$

$$- \sum_{i=0}^{N_x} \sum_{j=j_0+1}^{N_v} v_j \tilde{f}^h [g^h] \, dv - \sum_{i=1}^{N_x} \sum_{j=j_0+1}^{N_v} E_i \hat{\tilde{f}}^h [g^h] \, dx$$
\[ + \sum_{i, \epsilon_i < 0} \int_{e_i, j_0 + 1/2} |E_i| \hat{f}^{h, +}_i \, dx \]  
\(3.8a\)

\[ \mathcal{A}(f^h_2, g^h_2) = \sum_{i=1}^{N_d} \sum_{j=1}^{j_0} \int_{C_{i,j}} (v_j f^h \partial_z g^h - E_i f^h \partial_y g^h + \lambda f^h g^h) \, dv \]  
\(3.8b\)

\[ - \sum_{i=0}^{N_d} \sum_{j=1}^{j_0} v_j \hat{f}^h [g^h]_j \, dv - \sum_{i=1}^{N_d} \sum_{j=0}^{j_0 - 1} \int_{e_{i,j+1/2}} E_i \hat{f}^h [g^h] \, dx \]  
\(3.8c\)

\[ \mathcal{A}(f^h_1, g^h_1) = - \sum_{i, \epsilon_i > 0} \int_{e_i, j_0 + 1/2} |E_i| \hat{f}^{h, -}_i \, dx \]  
\(3.8d\)

In the domain decomposition approach, we replace \(\hat{f}^h\) in equations (3.8c) and (3.8d) by unknown numerical trace functions \(\hat{f}^{h,*} = \hat{f}^h_1 + \hat{f}^h_2\) and add the compatibility condition: for every \(x \in X_i\),

\[ \hat{f}^h_1(x) = \chi_{(\epsilon_i < 0)}(x) \hat{f}^h(x, v_{j_0 + 1/2}) =: \mathcal{P}_1(f^h_1)(x), \]  
\(3.9a\)

\[ \hat{f}^h_2(x) = \chi_{(\epsilon_i > 0)}(x) \hat{f}^h(x, v_{j_0 + 1/2}) =: \mathcal{P}_2(f^h_2)(x), \]  
\(3.9b\)

which we write as \(\hat{f}^{h,*} = \mathcal{P}(f^h)\). Then the original variational formulation (2.15) is equivalent to the coupled system

\[ \mathcal{A}(f^h_1, g^h_1) = \mathcal{Q}(g^h_1) - \mathcal{B}_2(\hat{f}^{h,*}_2, g^h_2) \]  
\(3.10a\)

\[ \mathcal{A}(f^h_2, g^h_2) = \mathcal{Q}(g^h_2) - \mathcal{B}_1(\hat{f}^{h,*}_1, g^h_1) \]  
\(3.10b\)

\[ \hat{f}^{h,*}_1 = \mathcal{P}_1(f^h_1) \]  
\(3.10c\)

\[ \hat{f}^{h,*}_2 = \mathcal{P}_2(f^h_2) \]  
\(3.10d\)

where

\[ \mathcal{B}_1(\hat{f}^{h,*}_1, g^h_1) = - \sum_{i, \epsilon_i < 0} \int_{e_i, j_0 + 1/2} |E_i| \hat{f}^{h,*}_1 g^{h,-} \, dx \]  
\(3.11a\)

\[ \mathcal{B}_2(\hat{f}^{h,*}_2, g^h_2) = - \sum_{i, \epsilon_i > 0} \int_{e_i, j_0 + 1/2} |E_i| \hat{f}^{h,*}_2 g^{h,+} \, dx. \]  
\(3.11b\)

We now show that (3.10) defines a contraction mapping on \(\mathcal{P}(Z^h)\). Given \(\lambda, h,\) and a positive constant \(C\), define the norm

\[ \| \cdot \|_{\lambda, h, C} = \left( \sum_{i=1}^{N_d} \int_{e_i, j_0 + 1/2} \left( \frac{|E_i|}{2} + C^{-1} \lambda h \right) |\cdot|^2 \, dx \right)^{1/2}. \]  
\(3.12\)

Then we have the following result.

**Lemma 3.1.** Given \(\hat{w}^{h,*} = \mathcal{P}(Z^h)\), let \(u^h \in Z^h\) satisfy

\[ \mathcal{A}(u^h_1, g^h_1) = - \mathcal{B}_2(\hat{w}^{h,*}_2, g^h_1) \]  
\(3.13a\)
\[
\mathcal{A}(u_2^h, g_2^h) = -B_1(\hat{w}_1^{h,*}, g_2^h),
\tag{3.13b}
\]

for any \(g^h \in \mathcal{Z}\) and let \(\hat{w}^{h,*} = \mathcal{P}(u^h)\). Then there exists a constant \(C > 0\), independent of \(\lambda\) and \(h\), such that

\[
\|\hat{w}^{h,*}\|_{\lambda, h, C} \leq \left(\frac{\frac{1}{2} E_{\text{max}}}{\frac{1}{2} E_{\text{max}} + C^{-1} \lambda h}\right)^{1/2} \|\hat{w}^{h,*}\|_{\lambda, h, C},
\tag{3.14}
\]

where \(E_{\text{max}} = \max_{1 \leq i \leq N_x} |E_i|\).

\textbf{Proof.} From the energy equation (2.18), we have

\[
\mathcal{A}(u_1^h, u_1^h) = \frac{1}{2} \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \int_{e_{i,j}+1/2} \left| v_j \right| [u^h]^2 dv + \frac{1}{2} \sum_{i=1}^{N_x} \sum_{j=0}^{N_y} \int_{e_{i,j+1/2}} \left| E_i \right| [u^h]^2 dx
\]

\[
+ \frac{1}{2} \sum_{i=1}^{N_x} \int_{e_{i,j}+1/2} \left| E_i \right| [u^{h,+}]^2 dx + \sum_{i=1}^{N_x} \sum_{j=0}^{N_y} \int_{C_{e_{i,j}}} \lambda |u^h|^2 dx dv
\tag{3.15}
\]

and, similarly,

\[
\mathcal{A}(u_2^h, u_2^h) = \frac{1}{2} \sum_{i=0}^{N_x} \sum_{j=1}^{N_y} \int_{e_{i,j}+1/2} \left| v_j \right| [u^h]^2 dv + \frac{1}{2} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \int_{e_{i,j+1/2}} \left| E_i \right| [u^h]^2 dx
\]

\[
+ \frac{1}{2} \sum_{i=1}^{N_x} \int_{e_{i,j}+1/2} \left| E_i \right| [u^{h,-}]^2 dx + \sum_{i=1}^{N_x} \sum_{j=0}^{N_y} \int_{C_{e_{i,j}}} \lambda |u^h|^2 dx dv.
\tag{3.16}
\]

Meanwhile, using (3.11) and applying Young’s inequality, we find that

\[
-B_1(\hat{w}_1^{h,*}, u_2^h) = \sum_{i: E_i < 0} \int_{e_{i,j}+1/2} \left| E_i \right| [\hat{w}_1^{h,*}]^2 u^{h,-} dx \leq \frac{1}{2} \sum_{i: E_i < 0} \int_{e_{i,j}+1/2} \left| E_i \right| ([\hat{w}_1^{h,*}]^2 + [u^{h,-}]^2) dx
\tag{3.17}
\]

and

\[
-B_2(\hat{w}_2^{h,*}, u_1^h) = \sum_{i: E_i > 0} \int_{e_{i,j}+1/2} \left| E_i \right| [\hat{w}_2^{h,*}]^2 u^{h,+} dx \leq \frac{1}{2} \sum_{i: E_i > 0} \int_{e_{i,j}+1/2} \left| E_i \right| ([\hat{w}_2^{h,*}]^2 + [u^{h,+}]^2) dx.
\tag{3.18}
\]

We sum the two equations in (3.13) and then substitute the four expressions above in the result. After dropping several of the positive jump terms on the left-hand side, we find that

\[
\frac{1}{2} \sum_{i=1}^{N_x} \int_{e_{i,j}+1/2} \left| E_i \right| ([u^{h,+}]^2 + [u^{h,-}]^2) dx + \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \int_{C_{e_{i,j}}} \lambda |u^h|^2 dx dv
\]

\[
\leq \frac{1}{2} \sum_{i: E_i > 0} \int_{e_{i,j}+1/2} \left| E_i \right| ([\hat{w}_2^{h,*}]^2 + [u^{h,+}]^2) dx + \frac{1}{2} \sum_{i: E_i < 0} \int_{e_{i,j}+1/2} \left| E_i \right| ([\hat{w}_1^{h,*}]^2 + [u^{h,-}]^2) dx.
\tag{3.19}
\]

Standard inverse inequalities [48] imply that for each \(i\),

\[
\int_{e_{i,j}+1/2} |u^{h,-}|^2 dx \leq C h^{-1} \int_{C_{e_{i,j}}} |u^h|^2 dx dv
\tag{3.20a}
\]
\[
\int_{e_{i,j_0+1/2}} |u^{h,+}|^2 dx \leq Ch^{-1} \int_{e_{i,j_0+1}} |u^h|^2 dx dv \tag{3.20b}
\]

for some constant \( C \) that is independent of \( h \). We substitute these inequalities into (3.19) and cancel out common terms in \( u^{h,+} \) and \( u^{h,-} \). The result is

\[
\sum_{i:E_i<0} \int_{e_{i,j_0+1/2}} \left( \frac{|E_i|}{2} + C^{-1} \lambda h \right) |u^{h,+}|^2 dx + \sum_{i:E_i>0} \int_{e_{i,j_0+1/2}} \left( \frac{|E_i|}{2} + C^{-1} \lambda h \right) |u^{h,-}|^2 dx
\]

\[
\leq \sum_{i:E_i>0} \int_{e_{i,j_0+1/2}} \frac{|E_i|}{2} |\hat{w}_1^h|^2 dx + \sum_{i:E_i<0} \int_{e_{i,j_0+1/2}} \frac{|E_i|}{2} |\hat{w}_1^h|^2 dx. \tag{3.21}
\]

By applying the definition of the upwind numerical trace (cf. (2.11)) and the projection \( P \) (cf. (3.9)) to the left-hand side of (3.21), we find that (3.21) is equivalent to

\[
\sum_{i=1}^{N_x} \int_{e_{i,j_0+1/2}} \left( \frac{|E_i|}{2} + C^{-1} \lambda h \right) |\hat{w}_1^h|^2 dx \leq \sum_{i=1}^{N_x} \int_{e_{i,j_0+1/2}} \frac{|E_i|}{2} |\hat{w}_1^h|^2 dx. \tag{3.22}
\]

Finally, a simple calculation shows that

\[
\frac{1}{2} |E_i| \leq \frac{\frac{1}{2} E_{\text{max}}}{\frac{1}{2} E_{\text{max}} + C^{-1} \lambda h} \left( \frac{1}{2} |E_i| + C^{-1} \lambda h \right). \tag{3.23}
\]

Therefore, the right-hand side of (3.22) can be bounded as

\[
\sum_{i=1}^{N_x} \int_{e_{i,j_0+1/2}} \frac{|E_i|}{2} |\hat{w}_1^h|^2 dx \leq \frac{\frac{1}{2} E_{\text{max}}}{\frac{1}{2} E_{\text{max}} + C^{-1} \lambda h} \sum_{i=1}^{N_x} \int_{e_{i,j_0+1/2}} \left( \frac{|E_i|}{2} + C^{-1} \lambda h \right) |\hat{w}_1^h|^2 dx. \tag{3.24}
\]

Combining (3.22) and (3.24) gives the desired result. \( \square \)

### 3.1.2. Linear Algebra Implementation.

To utilize the tools of linear algebra, we write the system (3.10) in operator form. Invoking notation similar to that of the variational formulation, we let

\[
f_1 = \{ f_{i,j} | j > j_0 \} \quad \text{and} \quad f_2 = \{ f_{i,j} | j \leq j_0 \} \tag{3.25}
\]

be vectors containing the volumetric coefficients in the expansion of \( f_1^h \) and \( f_2^h \), respectively, and similarly define the vector of basis functions

\[
b_1 = \{ b_{i,j} | j > j_0 \} \quad \text{and} \quad b_2 = \{ b_{i,j} | j \leq j_0 \} \tag{3.26}
\]

so that

\[
f_1^h = b_1^T f_1 \quad \text{and} \quad f_2^h = b_2^T f_2. \tag{3.27}
\]

We further introduce the trace coefficients

\[
\hat{f}_1^* = \{ \hat{f}_{i,j_0+1/2} | E_i < 0 \} \quad \text{and} \quad \hat{f}_2^* = \{ \hat{f}_{i,j_0+1/2} | E_i \geq 0 \} \tag{3.28}
\]

and trace basis functions

\[
\hat{b}_1^* = \{ \hat{b}_{i,j_0+1/2} | E_i < 0 \} \quad \text{and} \quad \hat{b}_2^* = \{ \hat{b}_{i,j_0+1/2} | E_i \geq 0 \} \tag{3.29}
\]
so that
\[ \hat{f}_1^*(x) = (\hat{b}_1^*)^T \hat{f}_1^* \quad \text{and} \quad \hat{f}_2^*(x) = (\hat{b}_2^*)^T \hat{f}_2^*. \] (3.30)

An operator form of (3.10a) and (3.10b) is derived by setting \( g_i^h \) equal to each component of \( b_i \) in those equations. The resulting matrix equations are
\[ A_1 f_1 = q_1 - B_2 f_2^* \quad \text{and} \quad A_2 f_2 = q_2 - B_1 f_1^*, \] (3.31)
where, for \( i = 1, 2, \)
\[ A_i^T = \mathcal{A}(b_i, b_i^T) \quad \text{and} \quad B_i^T = \mathcal{B}(\hat{b}_i^*)^T \quad \text{and} \quad q_i = \mathcal{Q}(b_i) \] (3.32)
with \( \sigma(1) = 2 \) and \( \sigma(2) = 1. \) Meanwhile, in operator form, the compatibility condition (cf. (3.9)) can be written as
\[ \hat{f}_1^* = P_1 f_1 \quad \text{and} \quad \hat{f}_2^* = P_2 f_2, \] (3.33)
where the matrices \( P_1 \) and \( P_2 \) are expressed in terms of the projection \( \mathcal{P}_1 \) and \( \mathcal{P}_2, \) respectively:
\[ P_i = \left( \int_{\Gamma} \hat{b}_i^*(\hat{b}_i^*)^T \, dx \right)^{-1} \int_{\Gamma} \hat{b}_i^* \mathcal{P}_i(b_i^T) \, dx = \int_{\Gamma} \hat{b}_i^* \mathcal{P}_i(b_i^T) \, dx, \] (3.34)
with the final equality above holding because the basis functions \( \hat{b}_{i,j}^k \) are all \( L^2 \)-orthogonal over \( \Gamma. \)

We write the full linear system for \( f \) and \( f^* \) as
\[ Af = q - B\hat{f}^* \] (3.35a)
\[ Pf = \hat{f}^* \] (3.35b)
where \( f \in \mathbb{R}^{3N_x N_e}, q \in \mathbb{R}^{3N_x N_e}, \hat{f}^* \in \mathbb{R}^{2N_x}, A \in \mathbb{R}^{(3N_x N_e) \times (3N_x N_e)}, B \in \mathbb{R}^{(3N_x N_e) \times (2N_e)}, \) and \( P \in \mathbb{R}^{(2N_e) \times (3N_x N_e)} \) are given by
\[ f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \quad q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}, \quad \hat{f}^* = \begin{pmatrix} \hat{f}_1^* \\ \hat{f}_2^* \end{pmatrix}, \] \[ A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & B_1 \\ B_2 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} P_1 & 0 \\ 0 & P_2 \end{pmatrix}. \] (3.36)

To reduce (3.35) into an equation for \( \hat{f}^* \), we apply \( PA^{-1} \) to (3.35a), invoke (3.35b) in the result, and then collect terms in \( \hat{f}^* \). The result is a Schur complement method by domain decomposition [44] for (3.35):
\[ (I + PA^{-1}B)\hat{f}^* = PA^{-1}q. \] (3.38)

We apply a Krylov solver to (3.38), which requires the application of \( P, A^{-1}, \) and \( A^{-1}B. \) The application of \( P \) is straightforward: its action is to first extract the volumetric coefficients of the cells adjacent to the boundary \( \Gamma \) and then merge them to the trace coefficients in \( \hat{f}_1^* \) and \( \hat{f}_2^* \), respectively. The application of \( A^{-1} \) and \( A^{-1}B \) can be derived from the sweep operator outlined in Algorithms 1 and 2 and used to solve (3.35a). In the linear algebra framework, the sweep operator can be
defined by \( S(z_0, z_1) := A^{-1}(z_0 + Bz_1) \) for any \( z_0 \in \mathbb{R}^{3N_xN_v} \), and \( z_1 \in \mathbb{R}^{2N_v} \). Hence, \( A^{-1}z_0 = S(z_0, 0) \) and \( A^{-1}Bz_1 = S(0, z_1) \).

In general, the size of \( \hat{f}^* \) is much less than the size of \( f \). As a consequence, the required memory footprint for the Krylov solver is substantially less than what would be required to solve (3.35a) directly. Once \( \hat{f}^* \) is calculated, a final sweep is used to compute \( f = S(q, -\hat{f}^*) \).

In order to ensure the algorithm based on (3.38) is well-posed, \( (I + PA^{-1}B) \) must be invertible. This fact is guaranteed by the following theorem, which is a direct consequence of Lemma 3.1.

**Theorem 3.2.** The spectral radius \( g(PA^{-1}B) \leq K < 1 \), where

\[
K = K(\lambda, h, C) = \left( \frac{1}{2}E_{\max} + C^{-1}\lambda h \right)^{1/2}
\]

and \( C \) is the positive constant in the statement of Lemma 3.1.

**Proof.** Let \( \hat{w}^* = \{\tilde{w}_{i,j_0+1/2}\}_{i=1}^{N_x} \neq 0 \) and \( \hat{u}^* = \{\tilde{u}_{i,j_0+1/2}\}_{i=1}^{N_x} \) be any two elements of \( \mathbb{R}^{2N_v} \) such that

\[
-PA^{-1}B\hat{w}^* = \hat{u}^*.
\]

Since the spectral radius is bounded above by any induced matrix norm (see for example, [45, Corollary 1.6]), it is sufficient to show that there exists a norm \( \| \cdot \| \) on \( \mathbb{R}^{2N_v} \) such that \( \| \hat{u}^* \| \leq K\| \hat{w}^* \| \). To identify such a norm, we write (3.40) as \( \hat{u}^* = Pu \), where \( Au = -B\hat{w}^* \). We then set \( u^h = b^Tu \), \( \hat{u}^{h,*} = (\hat{b}^*)^T\hat{u}^* \), and \( \hat{w}^{h,*} = (\hat{b}^*)^T\hat{w}^* \). A direct calculation, using (3.32) and (3.34), shows that \( u^h \) and \( \hat{w}^{h,*} \) satisfy (3.13). Thus according to Lemma 3.1, \( \| \hat{w}^{h,*} \|_{\lambda, h, C}^2 \leq K\| \hat{w}^{h,*} \|_{\lambda, h, C}^2 \) for some constant \( C > 0 \). In terms of \( \hat{w}^* \) and \( \hat{u}^* \), this relation is expressed equivalently as \( \| \hat{w}^* \|_{\lambda, h, C}^2 \leq K\| \hat{u}^* \|_{\lambda, h, C}^2 \), where for any \( v = (v_1, \ldots, v_{N_x}) \), with each \( v_i \in \mathbb{R}^2 \),

\[
\| v \|_{\lambda, h, C}^2 = \sum_{i=1}^{N_x} \int_{x_{i,j_0+1/2}} \left( \frac{|E_i|}{2} + C^{-1}\lambda h \right) |\hat{b}_{i,j_0+1/2}^T v_i|^2 dx.
\]

It is straightforward to verify that \( \| \cdot \|_{\lambda, h, C} \) is in fact a norm on \( \mathbb{R}^{2N_v} \). Hence,

\[
g(PA^{-1}B) \leq \| PA^{-1}B \|_{\lambda, h, C} \leq K.
\]

A direct consequence of Theorem 3.2 is that the matrix \( (I + PA^{-1}B) \) is positive definite. Thus it is invertible and, moreover, inversion with restarted GMRES (and hence GMRES) is guaranteed to converge [40, Section 6.11.4].

**Remark 2.** The bound in (3.39) suggests that \( g(PA^{-1}B) \) increases as the time step (recall \( \Delta t = \lambda^{-1} \)) increases and as the spatial mesh decreases. In practice, we observe that the spectral radius is more sensitive to the time step than the spatial mesh.

### 3.2. GMRES Performance Estimate

Theorem 3.2 can be used to estimate the performance of GMRES for solving (3.38). When solving \( My = c \) with initial guess \( y_0 \), GMRES produces at iteration \( k \) an estimate \( y_k \) satisfying

\[
\min_{\hat{y} \in y_0 + \mathbb{R}^k(M_r)} \| c - M\hat{y} \|_2 \quad \text{where} \quad \mathbb{R}^k(M_r) = \text{span}\{r_0, Mr_0, M^2r_0, \ldots, M^{k-1}r_0\}
\]

(3.43)
and \( r_0 = c - My_0 \).

**Corollary 1.** Let \( M = I + PA^{-1}B \), let \( K = K(\lambda,h,C) \) be defined as in Theorem 3.2. Then for any \( \varepsilon > 0 \), there exists an integer \( k_\varepsilon > 0 \), such that

\[
\frac{||c - My_k||_2}{||r_0||_2} \leq (K + \varepsilon)^k.
\]  

(3.44)

for all integers \( k \geq k_\varepsilon > 0 \).

**Proof.** Let \( H = -PA^{-1}B \) so that \( M = I - H \). According to Theorem 3.2, \( \rho(H) < 1 \). Thus \( M^{-1} \) has a convergent Neumann series, and

\[
\begin{align*}
  \| \mathbf{y} - \mathbf{y}_0 \|_2 &= \| (I - H)^{-1} r_0 \|_2 = \| (I + H + H^2 + …) r_0 \|_2 = \| H^k r_0 \|_2 \leq \| H^k \|_2 \| r_0 \|_2.
\end{align*}
\]  

(3.46)

According to Gelfand’s formula [21] (see also [39, Theorem 10.13] or [32, Theorem 16.4]), \( \rho(H) = \lim_{k \to \infty} (\| H^k \|_2)^{1/k} \). Hence, given \( \varepsilon > 0 \) there exists an integer \( k_\varepsilon > 0 \) such that, for all integers \( k \geq k_\varepsilon \),

\[
\| H^k \|_2 \leq (\rho(H) + \varepsilon)^k \leq (K + \varepsilon)^k.
\]  

(3.47)

To obtain (3.44), we simply divide the left and right expressions in (3.46) by \( \| r_0 \|_2 \) and invoke the bound in (3.47). \( \square \)

**Remark 3.** Another strategy for solving (3.38) is to use the fixed point iteration \( \mathbf{y}_{k+1} = \mathbf{c} + H \mathbf{y}_k \). The Lipschitz constant for this iteration is bounded by \( K \) in the \( \| \cdot \|_{\lambda,h,C} \) norm; hence it converges to a solution \( \mathbf{y}_\ast \). Moreover, the error \( \mathbf{e}_k = \mathbf{y}_\ast - \mathbf{y}_k \) satisfies \( \| \mathbf{e}_k \|_{\lambda,h,C}^2 \leq K^k \| \mathbf{e}_0 \|_{\lambda,h,C}^2 \), \( k = 1,2, \ldots \).

### 3.3. Extensions.

#### 3.3.1. Other boundary conditions.

For a sufficiently large computational domain, it is reasonable to impose a zero incoming boundary condition at the boundary of the velocity domain. However, the spatial boundary may be equipped with non-zero inflow data and, for some test problems, the boundary conditions may be periodic in \( x \). In the case of a non-zero inflow boundary, one need only modify the source term in the variational formulation (2.15) to include the incoming data. More specifically, given data \( f_a \) on \( \{(x,v) : x = a_x, v > 0\} \) and \( g_b \) on \( \{(x,v) : x = b_x, v < 0\} \), we replace \( Q \) in (2.17) by \( Q^d \), where

\[
Q^d(g^h) = Q(g^h) + \sum_{j=j_0+1}^{N_x} \int_{\gamma_{1/2,j}} \left| v_j \right| f_a g^h \, dv + \sum_{j=1}^{j_0} \int_{\gamma_{N_x+1/2,j}} \left| v_j \right| f_b g^h \, dv. \]  

(3.48)

In the case of a periodic boundary condition in \( x \), we introduce new functions on the outflow boundaries \( \{(x,v) : x = a_x, v < 0\} \) and \( \{(x,v) : x = b_x, v > 0\} \) which will serve as sources at the inflow boundaries \( \{(x,v) : x = b_x, v < 0\} \) and \( \{(x,v) : x = a_x, v > 0\} \) respectively. Specifically, let

\[
\tilde{f}_3(g^h) = \chi_{\{v \geq 0\}} f^h (v, x_{N+1/2}) =: \mathcal{P}_3(f_3^h), \]  

(3.49a)
\[
\hat{f}^h_4(v) = \chi_{\{v \leq 0\}} f^h(v, x_{1/2}) =: \mathcal{P}_4(f^h_2),
\] (3.49b)

and define the bilinear forms
\[
\mathcal{B}_3(\hat{f}^h_3, g^h_1) = -\sum_{j=1}^{N_{\text{e}}} \int_{x_{1/2,j}} \hat{f}^h_3 g^h_1 dv - \sum_{j=1}^{N_{\text{e}}} \int_{x_{1/2,j}} |v_j| \hat{f}^h_3 g^h_+ dv,
\] (3.50a)
\[
\mathcal{B}_4(\hat{f}^h_4, g^h_2) = -\sum_{j=1}^{N_{\text{e}}} \int_{x_{1/2,j}} \hat{f}^h_4 g^h_2 dv - \sum_{j=1}^{N_{\text{e}}} \int_{x_{1/2,j}} |v_j| \hat{f}^h_4 g^h_- dv.
\] (3.50b)

Then the new coupled system to solve is
\[
\mathcal{A}(f^h_1, g^h_1) = \mathcal{Q}(g^h_1) - \mathcal{B}_2(\hat{f}^h_2, g^h_1) - \mathcal{B}_3(\hat{f}^h_3, g^h_1),
\] (3.51a)
\[
\mathcal{A}(f^h_2, g^h_2) = \mathcal{Q}(g^h_2) - \mathcal{B}_1(\hat{f}^h_1, g^h_2) - \mathcal{B}_4(\hat{f}^h_4, g^h_2),
\] (3.51b)
\[
\hat{f}^h_i = \mathcal{P}_i(f^h_i), \quad i = 1, 2, 3, 4.
\] (3.51c)

As before, one can derive a linear algebra system from this variational formulation and implement a solution in terms of the expansion coefficients \( \hat{f}^h_i, i = 1, \ldots, 4 \) that correspond to each function \( f^h_i \). We leave these details to the reader.

3.3.2. 2D-2V and 3D-3V cases. The approach extends to the 2D-2V and 3D-3V cases in a straight-forward way. We assume a phase space domain of the form \([a_x, b_x]^d \times [a_v, b_v]^d\). For the 2D-2V case, we split the domain into four subdomains upon which the components of the vector \( v \) do not change sign:
\[
D^{(1,1)} = (a_x, b_x)^2 \times (0, b_v) \times (0, b_v), \quad D^{(1,2)} = (a_x, b_x)^2 \times (0, b_v) \times (a_v, 0),
\] (3.52a)
\[
D^{(2,1)} = (a_x, b_x)^2 \times (a_v, 0) \times (0, b_v), \quad D^{(2,2)} = (a_x, b_x)^2 \times (a_v, 0) \times (a_v, 0).
\] (3.52b)

The 3D-3V case can be similarly split into eight subdomains. We then modify the variational formulation, by introducing auxiliary functions onto the boundary between these subdomains, derive a linear algebra system, and apply the same iterative procedure. All of the results in Section 3.1 carry through in the same fashion.

4. Coupling to the Poisson Equation. We consider two approaches for evaluating the field \( E^{(k)} \) in (2.3). The first approach (see for example [12] and [38] in a semi-Lagrangian context) is a linearization based on Taylor series approximation:
\[
E^{(k)} \approx E^n + c_k \Delta t (\partial_t E)^n,
\] (4.1)
where the time derivative of \( E \) is determined by solving the equation
\[
\Delta_x (\partial_t \Phi)^n = -\nabla_x \cdot \left( \int_{\mathbb{R}^d} v f^n dv \right)
\] (4.2)
for \((\partial_t \Phi)^n\) and then computing the gradient \((\partial_t E)^n = \nabla_x (\partial_t \Phi)^n\). Equation (4.2) comes directly from the Vlasov-Poisson system. It is found by integrating the Vlasov equation in \( v \) and then using the Poisson equation to eliminate \( \rho \).

The second approach (see for example [11] in a Lagrangian context) is to solve the nonlinear system as a fixed-point problem for \( E \). If \( \mathcal{V}: E \mapsto f \) is the Vlasov solution operator for a given electric field \( E \) and \( \mathcal{P}: f \mapsto E \) is the Poisson operator for a given distribution \( f \), then the fixed-point problem is \( E = \mathcal{G}(E) \), where \( \mathcal{G} = \mathcal{P} \circ \mathcal{V} \). We
solve this fixed-point problem using Anderson acceleration [2], following directly the prescription given in [47].

Both approaches above require the numerical solution of the Poisson equation, which is calculated using a standard (continuous) finite element method with $Q^1$ elements on the same spatial mesh used to discretize the Vlasov equation. Since the method is standard (see, for example, [10] or [16]), we do not go into the details here.

5. Results. In this section, we report the results of simulations for 1D-1V and 2D-2V test problems using the domain decomposition (DD) method and compare timing results to direct application of an iterative solver (DI) for the steady-state Vlasov equation. We also compare the linear and non-linear approaches, outlined in Section 4, for coupling the Vlasov and Poisson equations. Linear systems arising from the Vlasov equation are solved using the generalized minimal residual method (GMRES) while linear systems arising from the Poisson equation are solved using the conjugate gradient method (CG). In both cases, we use implementations from the software package PETSc [5–7]. All computations are done with single threading to avoid complications in comparing different parallelized algorithms. The source code can be found at [20].

For each test problem, we report the average number of iterations per GMRES function call. For problems that are coupled to an electric field, we also give the average number of CG iterations needed to solve the Poisson equation. For problems that are coupled in a nonlinear fashion, we further report the average number of iterations in the Anderson acceleration (AA) algorithm. The relative tolerance used for the three solvers is: $10^{-10}$ for the GMRES solver, $10^{-8}$ for the CG solver, and $10^{-12}$ for the AA solver. In each case, the absolute tolerance is the square of the relative tolerance.

We tested convergence of the code using the manufactured solution in [38]. The results were as expected with $L_1$ and $L_2$ errors for both $f$ and $E$ converging at second order. The $L_{\infty}$ error converged at first order for $f$ and second order for $E$. We also tested mass, momentum, and energy conservation of our code. As expected, mass is conserved. Momentum is nearly conserved with the maximum absolute change in momentum for all 1D tests within $10^{-11}$. Energy is conserved within a relative change of approximately $10^{-3}$ for all 1D tests. For the 2D Landau test, the absolute change in momentum is approximately $10^{-10}$ and the relative change in energy is approximately $10^{-2}$. The lower conservation numbers for the 2D case are possibly due to the coarse grid used.

5.1. 1D-1V Results.

<table>
<thead>
<tr>
<th></th>
<th>$N_x$</th>
<th>$N_v$</th>
<th>$L$</th>
<th>$\Delta t$</th>
<th>$t_{\text{final}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D-1V squares</td>
<td>400</td>
<td>400</td>
<td>1.2</td>
<td>0.01 or 0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>two stream</td>
<td>400</td>
<td>400</td>
<td>$2\pi$</td>
<td>0.2</td>
<td>5.0 or 45.0</td>
</tr>
<tr>
<td>Landau 400 400</td>
<td>1.2</td>
<td>0.1 or 1.1</td>
<td>30.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2D-2V hypercubes</td>
<td>30</td>
<td>30</td>
<td>1.2</td>
<td>0.01 or 0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>Landau 30 30</td>
<td></td>
<td></td>
<td>$2\pi$</td>
<td>0.1 or 0.5</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Table 5.1: Parameters for the various numerical tests in Section 5. The value $L$ is the length of the domain in one dimension.
5.1.1. Advection of Squares. This test explores only the advection capabilities of the sweep algorithm by solving (1.1a) with the electric field set to $E(x) := -x$, which creates clockwise rotation in phase space. The initial condition is shown in Figure 5.1a where $f(x, v, 0) = 1$ on the four squares and $f(x, v, 0) = 0$ otherwise. The entire boundary is set to zero inflow, with the relevant parameters given in Table 5.1. The end result of the clockwise rotation is shown in Figures 5.1b and 5.1c. The timing data in Table 5.2 demonstrates that the DD solver is significantly faster than the DI solver, especially for larger time steps. Indeed, the DD solver requires only one GMRES iteration, whereas the number of iterations in the DI solver increases with the size of time step.

![Initial Condition](image1)

(a) Initial Condition

![Delta t = 0.01](image2)

(b) $\Delta t = 0.01$

![Delta t = 0.1](image3)

(c) $\Delta t = 0.1$

Fig. 5.1: Advection of squares example rotated clockwise. Color represents $f(x, v, t)$.

| Solver | $\Delta t$ | $\frac{v_{\text{max}} \Delta t}{E_{\text{max}}} | \frac{E_{\text{max}} \Delta t}{|\Delta t|} | \text{GMRES} | \text{Time} | \text{Speedup} |
|--------|------------|---------------------------------|-------------------------------|-------|-------|
| DI     | 0.01       | 1.995                          | 1.995                         | 21.92 | 91.6s | – |
| DD     | 0.01       | 1.995                          | 1.995                         | 1.00  | 15.5s | 5.9x |
| DI     | 0.10       | 19.95                          | 19.95                         | 269.20| 121.2s| – |
| DD     | 0.10       | 19.95                          | 19.95                         | 1.00  | 1.7s  | 71.3x|

Table 5.2: Timing data for the advection of squares test using a direct iterative (DI) solver and the domain decomposition (DD) approach.

5.1.2. Two Stream Instability. The two-stream instability is a standard test problem in which two streams of electrons interact to create a vortex with filamentation [9]. This test uses a periodic boundary in $x$ and a zero inflow boundary in $v$. The initial condition is

$$f(x, v, 0) = \frac{v^2}{\sqrt{8\pi}} (2 - \cos(x/2)) e^{-v^2/2},$$

and relevant test parameters are given in Table 5.1.

Table 5.3 contains iteration and timing data for various configurations. The DD solver is once again much faster than the DI solver. Figure 5.2 displays representative results at different times using the nonlinear coupling strategy for the electric field. This strategy takes about five times longer to compute than the linear approach, but for the cases shown here, we do not observe a significant difference in the two solutions.
Fig. 5.2: Two stream instability test at various times using the nonlinear electric field coupling algorithm and $\Delta t = 0.2$. Color represents $f(x, v, t)$.

<table>
<thead>
<tr>
<th>E-field Coupling</th>
<th>Solver</th>
<th>$t_{\text{final}}$</th>
<th>Average Iterations</th>
<th>Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear DI</td>
<td>5.0</td>
<td>0.7</td>
<td>508.78 1138.9s</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Linear DD</td>
<td>5.0</td>
<td>0.7</td>
<td>2.22  9.5s</td>
<td>120x</td>
<td></td>
</tr>
<tr>
<td>Nonlinear DI</td>
<td>5.0</td>
<td>0.5</td>
<td>182.07 2503.6s</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Nonlinear DD</td>
<td>5.0</td>
<td>0.5</td>
<td>0.92  42.1s</td>
<td>59x</td>
<td></td>
</tr>
<tr>
<td>Linear DD</td>
<td>45.0</td>
<td>0.7</td>
<td>1.44  70.8s</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Nonlinear DD</td>
<td>45.0</td>
<td>0.4</td>
<td>5.95  396.4s</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 5.3: Timing data for the two stream instability test using a direct iterative (DI) solver and the domain decomposition (DD) approach. The time step used is $\Delta t = 0.2$, and the CFLs are $\frac{v_{\text{max}} \Delta t}{\Delta x} = 39.9$ and $\frac{E_{\text{max}} \Delta t}{\Delta v} = 6.37$.

5.1.3. Landau Damping. Landau damping [28] provides another standard test problem. In this case, the boundary is periodic in $x$ and zero inflow in $v$. The initial condition is

$$f(x, v, 0) = \frac{1}{\sqrt{2\pi}} (1 + \alpha \cos(kx)) e^{-v^2/2},$$

(5.2)

where $\alpha = 0.01$ and $k = 0.5$. With these parameters, the $L^2$ norm of the electric field should decay exponentially at the rate $-0.1533$. The relevant parameters are given in Table 5.1.

The timings in Table 5.4 demonstrate that the DD solver is again faster than the DI solver and that the gap in performance increases with $\Delta t$. For the DD solver, the larger time step did not require significantly more GMRES iterations (or Anderson iterations in the nonlinear case) than the smaller time step. Hence the larger time steps resulted in a faster solution. On the other hand, with the DI solver, the solution took much longer with the larger time steps.

For the larger time steps, the linear electric field coupling to the Vlasov equation fails to achieve the correct Landau damping rate, while the nonlinear coupling achieves the correct decay rate for both time steps shown. However, for the DD solver, the time to solution for the linear coupling with $\Delta t = 0.1$ is nearly identical to the time to solution for the nonlinear coupling for $\Delta t = 1.1$. Hence, it is difficult to determine
which approach might be superior.

Fig. 5.3: Landau damping for linear/nonlinear electric field coupling and two different time steps. Green line shows the expected decay rate of the electric field. The nonlinear strategy retains the correct decay rate regardless of time step size. The nonlinear case for $\Delta t = 0.1$ shows the same behavior as the linear case.

<table>
<thead>
<tr>
<th>E-Field Coupling</th>
<th>Solver</th>
<th>$\Delta t$</th>
<th>CG Anderson</th>
<th>GMRES Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear DI</td>
<td>0.1</td>
<td>0.7</td>
<td>–</td>
<td>57.1</td>
<td>380.0s</td>
</tr>
<tr>
<td>Linear DD</td>
<td>0.1</td>
<td>0.7</td>
<td>–</td>
<td>1.5</td>
<td>24.2s</td>
</tr>
<tr>
<td>Nonlinear DI</td>
<td>0.1</td>
<td>0.4</td>
<td>5.1</td>
<td>14.9</td>
<td>612.0s</td>
</tr>
<tr>
<td>Nonlinear DD</td>
<td>0.1</td>
<td>0.5</td>
<td>4.1</td>
<td>0.5</td>
<td>95.2s</td>
</tr>
<tr>
<td>Linear DI</td>
<td>1.1</td>
<td>0.7</td>
<td>–</td>
<td>976.2</td>
<td>613.4s</td>
</tr>
<tr>
<td>Linear DD</td>
<td>1.1</td>
<td>0.7</td>
<td>–</td>
<td>2.4</td>
<td>2.8s</td>
</tr>
<tr>
<td>Nonlinear DI</td>
<td>1.1</td>
<td>0.3</td>
<td>12.8</td>
<td>225.4</td>
<td>1954.9s</td>
</tr>
<tr>
<td>Nonlinear DD</td>
<td>1.1</td>
<td>0.3</td>
<td>13.2</td>
<td>0.4</td>
<td>24.1s</td>
</tr>
</tbody>
</table>

Table 5.4: Timing data for the Landau damping test using a direct iterative (DI) solver and the domain decomposition (DD) approach. The CFL, $\frac{v_{\text{max}} \Delta t}{\Delta x}$, is 9.95 for $\Delta t = 0.1$ and 109.45 for $\Delta t = 1.1$. The CFL, $\frac{E_{\text{max}} \Delta t}{\Delta v}$, is 0.032 for $\Delta t = 0.1$ and 0.350 for $\Delta t = 1.1$.

5.2. 2D-2V Results.

5.2.1. Advection of Hypercubes. We again test the advection capabilities of the sweep algorithm by solving the 2D-2V version of (1.1a) with the electric field components $E_1(x) := -x_1$ and $E_2(x) := 0$ that create a clockwise rotation of the initial phase space distribution in the $x_1$-$v_1$ plane. Let $\Omega = [-0.7, -0.4] \cup [0.4, 0.7]$ and $\Omega^{(p)}$ be the Cartesian product of $p$ copies of $\Omega$. Then the initial condition is given by

$$f(x_1, x_2, v_1, v_2, 0) = \begin{cases} 1, & \text{if } (x_1, x_2, v_1, v_2) \in \Omega^{(4)} \\ 0, & \text{otherwise.} \end{cases} (5.3)$$
The entire boundary is set to zero inflow, with the relevant parameters given in Table 5.1. Table 5.5 demonstrates that the DD solver is again faster than the DI solver in the 2D-2V case, with the former only requiring two GMRES iterations to converge. The graphs of the initial and end conditions of $\int f dx_2 dv_2$ look similar to those in Figure 5.1 and hence are not shown.

<table>
<thead>
<tr>
<th>Solver</th>
<th>$\Delta t$</th>
<th>$\frac{v_{\text{max}} \Delta t}{\Delta x}$</th>
<th>$\frac{E_{\text{max}} \Delta t}{\Delta v}$</th>
<th>GMRES</th>
<th>Time (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI</td>
<td>0.01</td>
<td>0.145</td>
<td>0.145</td>
<td>13.87</td>
<td>610.6s</td>
<td>–</td>
</tr>
<tr>
<td>DD</td>
<td>0.01</td>
<td>0.145</td>
<td>0.145</td>
<td>2.00</td>
<td>249.0s</td>
<td>2.5x</td>
</tr>
<tr>
<td>DI</td>
<td>0.10</td>
<td>1.450</td>
<td>1.450</td>
<td>32.40</td>
<td>165.7s</td>
<td>–</td>
</tr>
<tr>
<td>DD</td>
<td>0.10</td>
<td>1.450</td>
<td>1.450</td>
<td>2.00</td>
<td>27.0s</td>
<td>6.1x</td>
</tr>
</tbody>
</table>

Table 5.5: Timing data for advection of hypercubes test using a direct iterative (DI) solver and the domain decomposition (DD) approach.

5.2.2. Landau Damping. For the 2D-2V Landau damping test [18], we use the initial condition

$$f(x_1, x_2, v_1, v_2, 0) = \frac{1}{2\pi} (1 + \alpha(\cos(k_1 x_1) + \cos(k_2 x_2))) e^{-(v_1^2 + v_2^2)/2}, \quad (5.4)$$

where $\alpha = 0.01$ and $k = 0.5$. The boundary is set to be periodic in $x_1$ and $x_2$, with zero inflow for $v_1$ and $v_2$. The $L^2(dx_1)$ and $L^2(dx_2)$ norms of

$$\bar{E}_1(x_1) = \frac{1}{b_x - a_x} \int E_1 dx_2 \quad \text{and} \quad \bar{E}_2(x_2) = \frac{1}{b_x - a_x} \int E_2 dx_1, \quad (5.5)$$

respectively, are both predicted to decay at the same rate as the 1D-1V case, which we observe in numerical tests. In Table 5.6, it is once again evident that the DD algorithm is superior to the DI algorithm, especially for larger time steps.

<table>
<thead>
<tr>
<th>E-Field Coupling</th>
<th>Solver</th>
<th>$\Delta t$</th>
<th>Average Iterations</th>
<th>Time (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>DI</td>
<td>0.1</td>
<td>19.5</td>
<td>20.4</td>
<td>1155s</td>
</tr>
<tr>
<td>Linear</td>
<td>DD</td>
<td>0.1</td>
<td>19.5</td>
<td>2.9</td>
<td>359s</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>DI</td>
<td>0.1</td>
<td>8.8</td>
<td>3.5</td>
<td>2202s</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>DD</td>
<td>0.1</td>
<td>9.0</td>
<td>1.4</td>
<td>1291s</td>
</tr>
<tr>
<td>Linear</td>
<td>DI</td>
<td>0.5</td>
<td>19.0</td>
<td>169.9</td>
<td>2044s</td>
</tr>
<tr>
<td>Linear</td>
<td>DD</td>
<td>0.5</td>
<td>19.0</td>
<td>4.1</td>
<td>89s</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>DI</td>
<td>0.5</td>
<td>6.4</td>
<td>35.2</td>
<td>4897s</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>DD</td>
<td>0.5</td>
<td>6.5</td>
<td>1.0</td>
<td>544s</td>
</tr>
</tbody>
</table>

Table 5.6: Timing data for the Landau damping test in 2D-2V using a direct iterative (DI) solver and the domain decomposition (DD) approach. The CFL, $\frac{v_{\text{max}} \Delta t}{\Delta x}$, is 1.45 for $\Delta t = 0.1$ and 7.25 for $\Delta t = 0.5$. The CFL, $\frac{E_{\text{max}} \Delta t}{\Delta v}$, is 0.0048 for $\Delta t = 0.1$ and 0.024 for $\Delta t = 0.5$. 
6. Discussions for Future Research. Several questions about the new approach are left to be answered. First, because the linear solver iterates on trace unknowns, one question is how to set the tolerance in order to guarantee an acceptable level of accuracy throughout the phase space. A second question is whether the improvement in performance makes implicit integration of Eulerian-type discretizations competitive with other methods such as time split semi-Lagrangian, time split Eulerian, and particle-in-cell. A third question that arises is what happens to the convergence of the domain decomposition method when the CFL with respect to velocity and electric field are large. Our tests only had large CFLs with respect to velocity. Higher CFLs with respect to the electric field may induce stronger cyclic dependencies and hence reduce the convergence rate. These questions are all the subject of further research. In addition, we plan to investigate whether this algorithm can be applied to other kinetic equations that include collision operators and/or magnetic fields.

Appendix A. Definition of Matrices for Section 2.2.2. In this appendix, we give the definitions for matrices introduced in Section 2.2.2.

\[ M_{i,j} = -v_j \int_{\xi_{i,j}} (\vec{e}_x b_{i,j}) \tilde{b}_{i,j}^T dxdv - E_i \int_{\xi_{i,j}} (\vec{e}_x b_{i,j}) \tilde{b}_{i,j}^T dxdv + \lambda \int_{\xi_{i,j}} b_{i,j} \tilde{b}_{i,j}^T dxdv, \]

\[ M_{i+1/2,j}^- = v_j \int_{\xi_{i+1/2,j}} b_{i,j} \tilde{b}_{i+1/2,j}^T dv, \quad M_{i-1/2,j}^+ = v_j \int_{\xi_{i-1/2,j}} b_{i,j} \tilde{b}_{i-1/2,j}^T dv, \]

\[ M_{i,j+1/2}^- = E_i \int_{\xi_{i,j+1/2}} b_{i,j} \tilde{b}_{i,j+1/2}^T dx, \quad M_{i,j-1/2}^+ = E_i \int_{\xi_{i,j-1/2}} b_{i,j} \tilde{b}_{i,j-1/2}^T dx. \]

\[ R_{i+1/2,j}^- = \int_{\xi_{i+1/2,j}} \hat{b}_{i+1/2,j}(v) \tilde{b}_{i+1/2,j}^T (x_{i+1/2}, v) dv, \]  \hspace{1cm} (A.2a)

\[ R_{i+1/2,j}^+ = \int_{\xi_{i+1/2,j}} \hat{b}_{i+1/2,j}(v) \tilde{b}_{i+1/2,j}^T (x_{i+1/2}, v) dv, \]  \hspace{1cm} (A.2b)

\[ R_{i,j+1/2}^- = \int_{\xi_{i,j+1/2}} \hat{b}_{i,j+1/2}(x) \tilde{b}_{i,j+1/2}^T (x, v_{j+1/2}) dx, \]  \hspace{1cm} (A.2c)

\[ R_{i,j+1/2}^+ = \int_{\xi_{i,j+1/2}} \hat{b}_{i,j+1/2}(x) \tilde{b}_{i,j+1/2}^T (x, v_{j+1/2}) dx. \]  \hspace{1cm} (A.2d)

\[ \tilde{M}_{i,j} = M_{i,j} + \chi_{\{v_j>0\}} M_{i+1/2,j}^- R_{i+1/2,j}^- - \chi_{\{v_j<0\}} M_{i-1/2,j}^+ R_{i-1/2,j}^+ \]

\[ + \chi_{\{E_i>0\}} M_{i,j+1/2}^- R_{i,j+1/2}^- - \chi_{\{E_i<0\}} M_{i,j-1/2}^+ R_{i,j-1/2}^+. \]  \hspace{1cm} (A.3a)

\[ \tilde{M}_{i+1/2,j}^- = M_{i+1/2,j}^- R_{i+1/2,j}^+, \quad \tilde{M}_{i-1/2,j}^+ = -M_{i-1/2,j}^- R_{i-1/2,j}^+, \]  \hspace{1cm} (A.3b)

\[ \tilde{M}_{i,j+1/2}^- = M_{i,j+1/2}^- R_{i,j+1/2}^+, \quad \tilde{M}_{i,j-1/2}^+ = -M_{i,j-1/2}^- R_{i,j-1/2}^+. \]  \hspace{1cm} (A.3c)

REFERENCES


1945.


