VLASOV SOLVER

MICHAEL UPDIKE, SINA ATALAY

https://github.com/MichaelUpdike/VlasovSolver

1 Introduction

The Vlasov equation is the fundamental equation of plasma physics. From it, every fluid equation and MHD are derived. The Vlasov equation and its approximations are used to simulate plasma confinement devices, study particle-wave interactions, and understand stellar interiors. Solvers for the Vlasov equation exist (e.g. Gekyll), but here we develop a python-based code from standard libraries.

Given a 1D phase-space density (distribution) of particles $f(x, v, t)$, the Vlasov equation evolves the particle system according to the standard rules of electrodynamics. Generally, given a f dependant Hamiltonian function(al) $H(x, v, t; f)$, the Vlasov equation in quaslinear and conservative form is

$$\frac{\partial f}{\partial t} + \frac{\partial H}{\partial v}\frac{\partial f}{\partial x} - \frac{\partial H}{\partial x}\frac{\partial f}{\partial v} = \frac{\partial f}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\partial H}{\partial v} f\right) - \frac{\partial}{\partial v}\left(\frac{\partial H}{\partial x} f\right) = 0. \tag{1}$$

Since this equation is a scalar conservation law, the Vlasov equation is a hyperbolic conservation law. The characteristics of the Vlasov equation are the solutions to the Hamiltonian flow

$$\dot{x} = \frac{\partial H}{\partial v}, \;\; \dot{v} = -\frac{\partial H}{\partial x}. \tag{2}$$

Thus, in the absence of collisions, the Vlasov equation describes the phase-space advection of particles by the Hamiltonian vector field $\mathbf{V}_H = \frac{\partial H}{\partial v}\frac{\partial}{\partial x} - \frac{\partial H}{\partial x}\frac{\partial}{\partial v}$. For example, if $H = v^2/2$ (the free Hamiltonian), then the Vlasov equation reads

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = 0 \tag{3}$$

which is nothing but a v-parameterized family of advection equations. The solution is $f(x, v, t) = f(x - vt, v, t)$. Parts of the distribution at higher v travel at higher speeds, causing the distribution f to advect and shear. Another famous example is choosing $H = v^2/2 + x^2/2$. Then the Vlasov equation reads

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} - x\frac{\partial f}{\partial v} = 0. \tag{4}$$

The solution is $f(x, v, t) = f(A^{-1}(t)(x, v))$ where $A(t)$ is the rotation matrix $A(t)(x, v) = (x\cos(t) + v\sin(t), v\cos(t) - x\sin(t))$. That is to say, each point of the distribution moves at a constant speed clockwise in phase space with a period 2π.

Like all advection equations, if $f(x, v, t = 0)$ is positive then $f(x, v, t)$ is positive. From a physical point of view, the positivity of f implies that the particle density is always nonnegative. Mathematically, when H is a functional of f, the Vlasov equation is no longer linear, and the positivity of f is important to ensuring the solution to the Vlasov equation is well-posed (i.e., to ensure the equation does not blow up). Thus, for any solver, keeping f nonnegative will be important. Another important property of the Vlasov equation is that

$$\frac{d}{dt}||f||_{L^1} = \frac{d}{dt}\int dv \int dx \; f(x, v, t) = 0. \tag{5}$$

This equation says that the total number of particles is invariant. More generally, given any function $F[f]$ of f,

$$\frac{d}{dt}\int dv \int dx \; F[f](x, v, t) = 0. \tag{6}$$

This implies, for example, that the L^2 norm of f is preserved. For any explicit solver, we must be okay with the L^2 norm of f decaying. It is a fact of life that explicit solvers for hyperbolic PDEs must include some diffusion to be stable. The L^1 norm of f should, however, always be preserved. This suggests we use a finite-volume method.

It should be noted that the Vlasov equation is much more complex than a simple advection equation because H must be solved for at each timestep. For the true Vlasov system $H = \frac{p^2}{2m} + q\phi(x)$ where

$$-\frac{\partial^2 \phi(x,t)}{\partial x^2} = C(x) + q\epsilon_0 \int dv\, f(x,v,t). \tag{7}$$

Here, q is the charge of the particle species, m is the mass, ϵ_0 is a numerical constant, and $C(x)$ is some prescribed background charge density. For testing purposes, we allow H to be a prescribed function. Moving forward, we will assume units such that $\epsilon_0 = -q = m = 1$ and $C(x) = C$. We interpret $f(x,v,t)$ to be a 1D distribution of electrons and C to be the charge from some stationary ions. We assume the net charge in our domain is zero.

The true Vlasov equation supports a variety of interesting behaviors. We focus on two examples: plasma oscillations and the two-stream instability. To derive plasma oscillations, we define the number density of particles $n(x,t) = \int dv\ f$, the flow velocity, $un = \int dv\ vf$, the pressure $p = \int dv\ (v-u)^2 f$, and the electric field $E = -\frac{\partial \phi}{\partial x}$. The first two velocity space moments of the Vlasov equation are

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x}(nu) = 0, \;\; n\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x}\right) = -nE - \frac{\partial p}{\partial x}. \tag{8}$$

Assuming f takes the form $f(x,v,t) = f_0(v) + f_1(x,v,t)$ with f_1 a perturbation to a cold, non-flowing, equilibrium f_0, then we can ignore the pressure term to get the linearized cold-fluid equations

$$\frac{\partial n_1}{\partial t} + n_0\frac{\partial u_1}{\partial x} = 0, \;\; \frac{\partial u_1}{\partial t} = -E_1. \tag{9}$$

These equations imply that

$$\frac{\partial^2 n_1}{\partial t^2} = -n_1 n_0. \tag{10}$$

Thus, to a good approximation, n_1 oscillates at an angular frequency of $\omega_p = \sqrt{n_0}$ as long as f_1 and $\int dv\ v^2 f(v)$ are small. ω_p is called the plasma frequency, and defines the natural timescale of almost all phenomena in plasma physics. From this expression, we see that the natural timestep of any simulation depends on f itself. This leads to the conclusion that the timestep Δt of a simulation must depend on f itself.

The two-stream instability is fundamental to a kinetic viewpoint of a plasma fluid. It cannot be derived from a fluid equation. To derive it, assume that $f = f_0 + f_1$ where f_0 is something like $f_0 = \delta(v+v_0)\frac{n_0}{2} + \delta(v-v_0)\frac{n_0}{2}$ and where f_1 is sinusoidal in space with wave number k. One can show that f_1 grows exponentially fast if and only if

$$kv_0 < \alpha\omega_p, \tag{11}$$

where $\alpha \approx .7$. Otherwise, f_1 simply oscillates. Thus, we see that for large-wavelength perturbations, the plasma is unstable. For short-wavelength perturbations, the plasma is stable with f_1 only oscillating.

We explored the mathematical and physical aspects of the Vlasov equation because they will lend insight into how to accurately discretize the equation. Our analysis also led to some interesting and nontrivial tests of our Vlasov equation solver, both for a prescribed and dynamical Hamiltonian. Now we turn to numerical methods.

2 Numerical Methods and Analysis

2.1 General Method

We wish to solve the Vlasov equation on the domain $(x,v) \in [-1,1] \times [-1,1]$. For boundary conditions, we will assume that the domain, and hence f, is periodic in x and v. This removes any issues with boundary conditions. However, it should be noted that making v periodic is problematic for the true Vlasov equation. While the electric field $E(x,t)$ can be made periodic in x, $\frac{\partial H}{\partial v} = v$ cannot be made periodic. We therefore must always assume that $f(x,v,t)$ is zero near $v = \pm 1$. Moving forward, we will always assume the Hamiltonian H is of the form $H(x,v) = K(v) - U(x)$. This is the form of the true Vlasov equation, and of most Hamiltonians that occur in practice.

To numerically integrate the 1D Vlasov equation, we first write the equation in conservative form,

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\partial K}{\partial v} f\right) + \frac{\partial}{\partial v}\left(\frac{\partial U}{\partial x} f\right) = 0. \tag{12}$$

Because the total particle density f obeys a conservation law, and conservation of total particle number is important, we discretize the Vlasov equation using a finite volume scheme. We use a uniform mesh to discretize the domain. Let Δx and Δv be the mesh spacing in the x and v directions respectively. Let $(x_i, v_j) = (i\Delta x - 1, j\Delta v - 1)$ be an equispaced discretization of our domain into $N_x \times N_v$ points. By periodicity, we have that $x_{N_x} = x_0$ and $v_{N_V} = v_0$. For analysis purposes, we assume that Δx and Δv are comparably small. We define

$$f_{i+1/2,j+1/2} = \frac{1}{\Delta x \Delta v} \int_{x_i}^{x_{i+1}} dx \int_{v_i}^{v_{i+1}} dv \; f(x, v, t). \tag{13}$$

That is, $f_{i+1/2,j+1/2}$ is the average of f in the cell $[x_i, x_{i+1}] \times [v_i, v_{i+1}]$. Averaging the Vlasov equation over the cell $[x_i, x_{i+1}] \times [v_i, v_{i+1}]$ we have exactly that

$$\frac{df_{i+1/2,j+1/2}(t)}{dt} + \frac{\bar{F}^x_{i+1,j+1/2}(t) - \bar{F}^x_{i,j+1/2}}{\Delta x} + \frac{\bar{F}^v_{i+1/2,j+1}(t) - \bar{F}^v_{i+1/2,j}(t)}{\Delta v} = 0, \tag{14}$$

where the averaged fluxes $\bar{F}^x$ and $\bar{F}^v$ are defined by

$$\bar{F}^x_{i,j+1/2}(t) = \frac{1}{\Delta v} \int_{v_i}^{v_{j+1}} dv \; \left(\frac{\partial K}{\partial v} f \right) (x_i, v, t), \tag{15}$$

$$\bar{F}^v_{i+1/2,j}(t) = \frac{1}{\Delta x} \int_{x_i}^{x_{j+1}} dx \; \left(\frac{\partial U}{\partial x} f \right) (x, v_j, t). \tag{16}$$

Numerically solving the Vlasov equation amounts to numerically approximating the averaged fluxes, and fully discretizing the time derivative. We start by setting up a semidiscrete scheme.

We can write using Taylor series that

$$\begin{aligned} \bar{F}^x_{i,j+1/2}(t) &= \frac{1}{\Delta v} \int_{v_i}^{v_{j+1}} dv \; \left(\frac{\partial K}{\partial v} f \right) (x_i, v, t) \\ &= \frac{1}{\Delta v} \int_{v_i}^{v_{j+1}} dv \; \left(\frac{\partial K}{\partial v} f \right) (x_i, v_{j+1/2}, t) + (v - v_{j+1/2}) \frac{\partial}{\partial v} \left(\frac{\partial K}{\partial v} f \right) (x_i, v_{j+1/2}, t) + O(\Delta v^2) \\ &= \left(\frac{\partial K}{\partial v} f \right) (x_i, v_{j+1/2}, t) + O(\Delta v^2). \end{aligned} \tag{17}$$

Similarly,

$$\bar{F}^v_{i+1/2,j}(t) = \left(\frac{\partial U}{\partial x} f \right) (x_{i+1/2}, v_j, t) + O(\Delta x^2) \tag{18}$$

In our numerical scheme, only $f_{i+1/2,j+1/2}$ is known. In order to get f at the cell edges, we need to Taylor expand the definition of $f_{i+1/2,j+1/2}$ to get that

$$\begin{aligned} f_{i+1/2,j+1/2}(t) &= \frac{1}{\Delta x \Delta v} \int_{x_i}^{x_{i+1}} \int_{v_j}^{v_{j+1}} f(x, v, t) \\ &= \frac{1}{\Delta x \Delta v} \int_{x_i}^{x_{i+1}} \int_{v_j}^{v_{j+1}} f(x_i, v_{j+1/2}, t) + (x - x_i) \frac{\partial f}{\partial x}(x_i, v_{j+1/2}, t) + (v - v_{j+1/2}) \frac{\partial f}{\partial x}(x_i, v_{j+1/2}, t) + O(h^2) \\ &= O(h^2) + f(x_i, v_{j+1/2}, t) + \frac{\Delta x}{2} \frac{\partial f}{\partial x}(x_i, v_{j+1/2}, t), \end{aligned} \tag{19}$$

where h refers to either Δx or Δv which are assumed to be of the same order. Inverting this expression, we learn that

$$f^+_{i,j+1/2}(t) = f_{i+1/2,j+1/2} - \frac{\Delta x}{2} \frac{\partial f}{\partial x}(x_i + 0, v_{j+1/2}, t) + O(h^2), \tag{20}$$

where the + symbol denotes that we derived $f^+_{i,j+1/2}(t)$ from the average of f over the cell $[x_i, x_{i+1}] \times [v_i, v_{i+1}]$. We would say that $f^+(x_i, v_{j+1/2}, t)$ is the right value of $f(x_i, v_{j+1/2}, t)$. Formally, $f^+(x_i, v_{j+1/2}, t) = \lim_{\epsilon \to 0^+} f(x_i + \epsilon, v_{j+1/2}, t)$. We can get $f^-_{i,j+1/2} = \lim_{\epsilon \to 0^-} f^-(x_i + \epsilon, v_{j+1/2}, t)$ using the Taylor series method again to find that

$$f^-_{i,j+1/2}(t) = f_{i-1/2,j+1/2} + \frac{\Delta x}{2} \frac{\partial f}{\partial x}(x_i - 0, v_{j+1/2}, t) + O(h^2). \tag{21}$$

Similarly,

$$f^{+}_{i+1/2,j}(t) = f_{i+1/2,j+1/2} - \frac{\Delta v}{2}\frac{\partial f}{\partial v}(x_{i+1/2}, v_j + 0) + O(h^2), \tag{22}$$

and

$$f^{-}_{i+1/2,j}(t) = f_{i+1/2,j-1/2} + \frac{\Delta v}{2}\frac{\partial f}{\partial v}(x_{i+1/2}, v_j - 0) + O(h^2). \tag{23}$$

The edge value $f_{i,j+1/2}$ must be chosen to be some convex combination of $f^{+}_{i,j+1/2}$ and $f^{-}_{i,j+1/2}$. Similarly, the edge value $f_{i+1/2,j}$ must be some convex combination of the edge values $f^{+}_{i+1/2,j}$ and $f^{-}_{i+1/2,j}$. Once the edge values of f are known, We can approximate the numerical fluxes using Equations (17) and (18).

First-Order Scheme: Accuracy Analysis, Stability Analysis, Monotonicity Analysis, and Modified Equation

To setup our first-order method, we drop any $O(h)$ terms from the numerical fluxes. This amounts to approximating $f^{\pm}_{i,j+1/2} = f_{i\pm 1/2,j+1/2}$ and $f^{\pm}_{i+1/2,j} = f_{i+1/2,j\pm 1/2}$. But which choice do we make for $f_{i,j+1/2}$? The simple (wrong) answer is to average the two choices. To see why this choice is wrong, suppose that $H = K(v)$. Suppose we approximate $f_{i,j+1/2} = \frac{1}{2}(f_{i+1/2,j+1/2} + f_{i-1/2,j+1/2})$. The Vlasov equation in semi-discrete form is (up to an O(h^2) error from discretizing the derivatives)

$$\frac{df_{i+1/2}}{dt} = -K'(v_{j+1/2})\frac{f_{i+3/2} - f_{i-1/2}}{2\Delta x} = \mathbb{A}(f_{i+1/2}), \tag{24}$$

where we drop the j index since these are completely decoupled. $\mathbb{A}$ is a maxtrix acting on the cell averages $f_{i+1/2}$. To get the eigenvalues of $\mathbb{A}$, define the vector $(f_{n+1/2}) = \mathbf{v}_k = (e^{ik\Delta x(n+1/2)})$ with k a multiple of π. Then

$$\mathbb{A}\mathbf{v}_k = -i\frac{K'(v_{j+1/2})}{\Delta x}\sin(k\Delta x)\mathbf{v}_k = a_k\mathbf{v}_k, \tag{25}$$

so $\mathbf{v}_k$ is an eigenvector of $\mathbb{A}$ with strictly imaginary eigenvalue a_k. This is problematic for explicit methods. For explicit methods, the part of the imaginary axis near zero is not well contained in the stability region. At best, we would need to use something like $RK(3)$ or $RK(4)$, and even then, the imaginary axis is tangent to the stability region. If we tried to use, say, forward Euler to integrate the semi-discrete form, then our scheme would quickly blow up. To see this, we write $(f_{i+1/2}(t)) = \sum_k c_k(t)\mathbf{v}_k$. Forward Euler integration of Equation (24) reads

$$c_k^{n+1} = (1 + a_k\Delta t)c_k^n, \tag{26}$$

implying that $|c_k^n| = |c_k^0||1 + a_k\Delta t|^n$. Since $|1 + a_k\Delta t| > 1$, $|c_k^n|$ blows up geometrically.

To understand the correct choice of f at the edges, suppose that f is piecewise constant on each cell. That is, $f(x,v) = f_{i+1/2,j+1/2}$ for $(x,v) \in [x_i, v_{i+1}] \times [v_j, v_{j+1}]$. In this limit, the edge values for f become exact. Let's focus on the edge $x_i \times [v_j, v_{j+1}]$. If $\frac{1}{\Delta v}\int_{v_j}^{v_{j+1}} dv\ f(x_i, v)\frac{\partial K}{\partial v}(v) \approx f(x_i, v_{j+1/2})\frac{\partial K}{\partial v}(v_{j+1/2}) > 0$, then $\bar{F}^x_{i,j+1/2}$ is positive implying at the edge f flows from the cell $[x_{i-1}, x_i] \times [v_j, v_{j+1}]$ to the cell $[x_i, x_{i+1}] \times [v_j, v_{j+1}]$. Hence, in this case, only $f_{i-1/2,j+1/2} = f^{-}_{i,j+1/2}$ contributes to the flux at the edge. Conversely, if $K'(v_{j+1/2}) < 0$ then only $f_{i+1/2,j+1/2} = f^{+}_{i,j+1/2}$ contributes to the flux. This suggests that we take

$$\begin{aligned} f_{i,j+1/2} = f^{-}_{i,j+1/2} &= f_{i-1/2,j+1/2}, \quad \frac{\partial K}{\partial v}(v_{j+1/2}) \geq 0, \\ = f^{+}_{i,j+1/2} &= f_{i+1/2,j+1/2}, \quad \frac{\partial K}{\partial v}(v_{j+1/2}) < 0. \end{aligned} \tag{27}$$

That is, the value for $f_{i,j+1/2}$ is determined from $f_{i\pm 1/2,j+1/2}$ by upwinding. Plugging this into our approximate expression for $\bar{F}^x_{i,j+1/2}$, we get that the numerical flux is (up to an $O(h)$ error)

$$\bar{F}^x_{i,j+1/2} = \frac{\partial K}{\partial v}(v_{j+1/2})\frac{f_{i+1/2,j+1/2} + f_{i-1/2,j+1/2}}{2} - \left|\frac{\partial K}{\partial v}(v_{j+1/2})\right|\frac{f_{i+1/2,j+1/2} - f_{i-1/2,j+1/2}}{2}. \tag{28}$$

We see that while the choice of $f_{i,j+1/2}$ is not continuous, the choice of numerical flux is continuous. We see that the upwinded fluxes differ from the symmetric choice of flux by an extra term proportional to $\left|\frac{\partial K}{\partial v}(x_i, v_{j+1/2})\right|$. We will show that this term is dissipative, and hence stabilizes the discrete method.

Preforming unwinding to get $f_{i+1/2,j}$, we similarly learn that

$$\bar{F}^v_{i+1/2,j} = \frac{\partial U}{\partial x}(x_{i+1/2})\frac{f_{i+1/2,j+1/2} + f_{i+1/2,j-1/2}}{2} - \left|\frac{\partial U}{\partial x}(x_{i+1/2})\right|\frac{f_{i+1/2,j+1/2} - f_{i+1/2,j-1/2}}{2}. \tag{29}$$

Our first-order, upwinded, semi-discrete scheme therefore reads

$$\frac{df_{i+1/2,j+1/2}}{dt} + \frac{\partial K}{\partial v}(v_{j+1/2})\frac{f_{i+3/2,j+1/2} - f_{i-1/2,j+1/2}}{2\Delta x} + \frac{\partial U}{\partial x}(x_{i+1/2})\frac{f_{i+1/2,j+3/2} - f_{i+1/2,j-1/2}}{2\Delta v} \tag{30}$$

$$= \left|\frac{\partial K}{\partial v}(v_{j+1/2})\right|\frac{f_{i+3/2,j+1/2} - 2f_{i+1/2,j+1/2} + f_{i-1/2,j+1/2}}{2\Delta x} + \left|\frac{\partial U}{\partial v}(x_{i+1/2})\right|\frac{f_{i+1/2,j+3/2} - 2f_{i+1/2,j+1/2} + f_{i+1/2,j-1/2}}{2\Delta v}.$$

The left-hand side looks like the centered difference scheme we considered earlier. The right-hand side, however, is new and clearly describes some sort of dissipation. Equation (30) differs from the Vlasov equation by $O(h)$ terms. We therefore have a first-order semi-discrete scheme.

To discretize the time derivative, we use forward Euler, which amounts to using the first-order Taylor series $f_{i+1/2,j+1/2}(t+\Delta t) = f_{i+1/2,j+1/2}(t) + \Delta t \frac{d}{dt} f_{i+1/2,j+1/2}(t) + O(\Delta t^2)$ to approximate $\frac{df_{i+1/2,j+1/2}}{dt}(n\Delta t) = \frac{f^{n+1}_{i+1/2,j+1/2} - f^n_{i+1/2,j+1/2}}{\Delta t} + O(\Delta t)$ where $f^n_{i+1/2,j+1/2} = f_{i+1/2,j+1/2}(n\Delta t)$. Our fully discrete form is therefore

$$\frac{f^{n+1}_{i+1/2,j+1/2} - f^n_{i+1/2,j+1/2}}{\Delta t} + \frac{\partial K}{\partial v}(v_{j+1/2})\frac{f^n_{i+3/2,j+1/2} - f^n_{i-1/2,j+1/2}}{2\Delta x} + \frac{\partial U}{\partial x}(x_{i+1/2})\frac{f^n_{i+1/2,j+3/2} - f^n_{i+1/2,j-1/2}}{2\Delta v} \tag{31}$$

$$= \left|\frac{\partial K}{\partial v}(v_{j+1/2})\right|\frac{f^n_{i+3/2,j+1/2} - 2f^n_{i+1/2,j+1/2} + f^n_{i-1/2,j+1/2}}{2\Delta x} + \left|\frac{\partial U}{\partial v}(x_{i+1/2})\right|\frac{f^n_{i+1/2,j+3/2} - 2f^n_{i+1/2,j+1/2} + f^n_{i+1/2,j-1/2}}{2\Delta v}.$$

Equation (31) differs from the semi-discrete scheme by terms of order $O(\Delta t)$. Our fully discrete method is therefore first-order in both space and time. At first-order, $f_{i+1/2,j+1/2}$ is nothing but the value of f at the point $(x_{i+1/2}, v_{j+1/2})$. We can therefore associate $f_{i+1/2,j+1/2}$ with $f(x_{i+1/2}, v_{j+1/2})$, at least in our first-order method. Expanding the terms in Equation (31) using a Taylor series, and ignoring any quadratic terms in h and Δt, we have the modified equation

$$\begin{aligned}\frac{\partial f}{\partial t} + \frac{\partial K}{\partial v}\frac{\partial f}{\partial x} + \frac{\partial U}{\partial x}\frac{\partial f}{\partial v} &= -\frac{\Delta t}{2}\frac{\partial^2 f}{\partial t^2} + \frac{\Delta x}{2}\left|\frac{\partial K}{\partial x}\right|\frac{\partial^2 f}{\partial x^2} + \frac{\Delta v}{2}\left|\frac{\partial U}{\partial x}\right|\frac{\partial^2 f}{\partial v^2} \\ &= -\frac{\Delta t}{2}\left(\frac{\partial K}{\partial v}\frac{\partial}{\partial x} + \frac{\partial U}{\partial x}\frac{\partial}{\partial v}\right)^2 f + \frac{\Delta x}{2}\left|\frac{\partial K}{\partial v}\right|\frac{\partial^2 f}{\partial x^2} + \frac{\Delta v}{2}\left|\frac{\partial U}{\partial x}\right|\frac{\partial^2 f}{\partial v^2} \\ &= \hat{D}f.\end{aligned} \tag{32}$$

The principal symbol of $\hat{D}$ is

$$\text{Symb}(\hat{D}) = (\frac{\Delta x}{2}|K'| - \frac{\Delta t}{2}|K'|^2)\partial_x^2 + (\frac{\Delta v}{2}|U'| - \frac{\Delta t}{2}|U'|^2)\partial_v^2 - \Delta t K'U'\partial_x\partial_v. \tag{33}$$

Viewing Symb($\hat{D}$) as a symmetric matrix, the determinant of the symbol is

$$\det(\text{Symb}(\hat{D})) = (\frac{\Delta x}{2}|K'| - \frac{\Delta t}{2}|K'|^2)(\frac{\Delta v}{2}|U'| - \frac{\Delta t}{2}|U'|^2) - \frac{\Delta t^2}{4}|K'|^2|U'|^2 = \frac{|U'||K'|\Delta x\Delta v}{2}(1 - \frac{|K'|\Delta t}{\Delta x} - \frac{|U'|\Delta t}{\Delta v}), \tag{34}$$

while

$$\text{tr}(\text{Symb}(\hat{D})) = \frac{|K'|}{2}(\Delta x - \Delta t|K'|) + \frac{|U'|}{2}(\Delta v - \Delta t|U'|). \tag{35}$$

The eigenvalues of Symb($\hat{D}$) are both positive iff both the determinant and the trace are positive. The symbol having positive eigenvalues implies that $-\hat{D}$ is elliptic and thus that the modified equation is an advection-diffusion equation. Defining $a = \max(|K'|)$ and $b = \max(|U'|)$ we see that $\hat{D}$ is elliptic iff

$$C = \Delta t\left(\frac{a}{\Delta x} + \frac{b}{\Delta v}\right) < 1. \tag{36}$$

We refer to C as the Courant number, since C naturally generalizes the Courant number for the advection equation. We have shown for $C < 1$ that the modified equation is stable. Heuristically, for $C < 1$, the fully discrete scheme should be stable. We will prove directly that this is true. We first prove a few important properties of our scheme. We write the fully discrete form in conservative form

$$\frac{f^{n+1}_{i+1/2,j+1/2}-f^{n}_{i+1/2,j+1/2}}{\Delta t}+\frac{\bar{F}^{x,n}_{i+1,j+1/2}-\bar{F}^{x,n}_{i,j+1/2}}{\Delta x}+\frac{\bar{F}^{v,n}_{i+1/2,j+1}-\bar{F}^{v,n}_{i+1/2,j}}{\Delta v}=0. \tag{37}$$

Summing over i and j and noting the fluxes are periodic, we get that

$$\sum_{i,j} f^{n+1}_{i+1/2,j+1/2}=\sum_{i,j} f^{n}_{i+1/2,j+1/2}, \tag{38}$$

so the total particle number $\int dxdv\ f$ is exactly preserved. Notice that we never used the definition of the fluxes (only periodicity), so $\sum_{i,j} f_{i+1/2,j+1/2}$ is preserved by any finite volume scheme. For the method at hand, we show that if $f^n>0$ then so is f^{n+1}. Indeed, assume that $f^n_{i+1/2,j+1/2}\geq 0$. Then we have that

$$f^{n+1}_{i+1/2,j+1/2}=f^{n}_{i+1/2,j+1/2}-\Delta t\left(\frac{\bar{F}^{x,n}_{i+1,j+1/2}-\bar{F}^{x,n}_{i,j+1/2}}{\Delta x}+\frac{\bar{F}^{v,n}_{i+1/2,j+1}-\bar{F}^{v,n}_{i+1/2,j}}{\Delta v}\right). \tag{39}$$

Suppose that $K'(v_{j+1/2})>0$ and $U'(x_{i+1/2})>0$ (the other cases are similar). Then all the fluxes are positive. This implies that

$$f^{n+1}_{i+1/2,j+1/2}\geq f^{n}_{i+1/2,j+1/2}-\Delta t\left(\frac{\bar{F}^{x,n}_{i+1,j+1/2}}{\Delta x}+\frac{\bar{F}^{v,n}_{i+1/2,j+1}}{\Delta v}\right). \tag{40}$$

By upwinding, we have that $\bar{F}^{x}_{i+1,j+1/2}/K'(v_{j+1/2})=\bar{F}^{v}_{i+1/2,j+1}/K'(x_{i+1/2})=f_{i+1/2,j+1/2}$. Thus

$$f^{n+1}_{i+1/2,j+1/2}\geq f^{n}_{i+1/2,j+1/2}\left(1-\Delta t\left(1-\frac{a}{\Delta x}-\frac{b}{\Delta v}\right)\right). \tag{41}$$

Hence, if $C\leq 1$, $f^{n+1}_{i+1/2,j+1/2}$ is positive for all times. Combing this with the conservation of $\int dxdv\ f$, we learn that $||f||_{L^1}$ is invariant. Thus, for $C<1$, we have proven our first-order numerical scheme is stable, particle-number conserving, and positivity preserving. From the modified equation, we also learn that our scheme is always dissipative, with the strength of dissipation being proportional to h. The Courant condition $C\leq 1$ implies that we must always choose $\Delta t\leq 1/(a/\Delta x+b/\Delta v))$. In our simulations using first-order methods, we always set $C=1$.

Implementing this first-order method with a prescribed Hamiltonian, we see that this scheme is indeed stable and positivity preserving. However, this first-order method is very diffusive as seen by our simulations of the simple harmonic oscillator.

2.2 Adding an Electric Field Solver w/ Complexity Analysis

Thus far, we have assumed the Hamiltonian is given. For the true Vlasov equation, $U'(x,t)=E(x,t)$ must be obtained from the equation

$$\frac{\partial E}{\partial x}=C-\int dv\ f=-n(x,t). \tag{42}$$

Integrating averaging this equation from x_i to x_{i+1} we have that

$$E(x_{i+1})-E(x_i)=C-\Delta v\Delta x\sum_j f_{i+1/2,j+1/2}. \tag{43}$$

Summing over i, noting that $E(x)$ is periodic, we have the solvability condition

$$C=\frac{1}{2}\Delta v\Delta x\sum_i\sum_j f_{i+1/2,j+1/2}. \tag{44}$$

Provided this condition is satisfied, there are formally an infinite number of solutions to Equation (43). Namely, if $E(x_i)$ is a solution so is $E(x_i)+c$ for any constant c. We thus additionally demand that $E(x_i)$ has zero-mean. To a good approximation, this translates into the condition that $\sum_i E(x_i)=0$. This condition is needed to prevent runaway solutions to the Vlasov equation. Equation (43) is very easily solvable. Let $\tilde{E}$ be the solution to Equation (43) with $\tilde{E}(0) = 0$. This implies we can use a simple loop to successively get

$$\tilde{E}(x_{i+1}) = \tilde{E}(x_i) + C - \Delta v \Delta x \sum_j f_{i+1/2,j+1/2}. \tag{45}$$

We can then subtract the mean of $\tilde{E}(x)$ to get $E(x)$. In order to get $E(x_{i+1/2})$, we use Taylor series to learn that

$$E(x_{i+1/2}) = \frac{1}{2}\left(E_i + E_{i+1}\right) + O(\Delta x^2). \tag{46}$$

Thus, to second-order, we can take $U'(x_{i+1/2}, t) = \frac{1}{2}\left(E_i + E_{i+1}\right)$. For all of our methods, U is solved at each time before a forward timestep is taken. Taking $b^n = \max(|\frac{\partial U^n}{\partial x}|)$, the Courant condition becomes

$$\Delta t^n \leq 1/(1/\Delta x + b^{(n)}/\Delta v), \tag{47}$$

where we used that $a = \max|v| = 1$. As predicted, for the true Vlasov equation, the timestep must always be adaptive since b is time-dependent. Without an explicit solution to the Vlasov equation, Δt cannot be known a priori. Therefore, in our program, we calculate $\Delta t^n = 1/(1/\Delta x + b^{(n)}/\Delta v)$ after solving for E^n but before preforming the Forward Euler step. As long as Δt^n is chosen this way at every timestep, then our numerical schemes will be stable (all our schemes have the same Courant conditions). Further, $f^n_{i+1/2,j+1/2}$ will always be positive provided our integrator is positivity preserving, such as in the case of our first-order method.

A more complicated method for getting E at the half-grid points is Fourier interpolation. In this method, we write

$$E(x_i) = \sum_k e^{ikx_i} E(k), \tag{48}$$

where k are the multiples of π in the range $[-N_x/2, N_x/2)$, and where $E(k)$ is obtained by a fast Fourier transform. We can then interpolate $E(x)$ to be the (necessarily periodic) function

$$E(x) = \sum_k e^{ikx} E(k). \tag{49}$$

From this expression, we can easily obtain $E(x_{i+1/2})$ using an inverse fast Fourier transform. While the Fourier interpolation method is $O(N_x \log N_x)$ vs. $O(N_x)$ for the 2-point averaging interpolation, Fourier interpolation converges exponentially fast as $\Delta x \to 0$. Further, if N_x is a power of 2, the Cooley-Turkey algorithm ensures the FFT is very fast. Since $N \log(N)$ complexity is much smaller than the N^2 complexity of a single Euler step, Fourier interpolation does not pose a significant slowdown.

Fourier interpolation (typically) leads to a smoother electric field. This is important, for example, in plasma oscillations where the electric field is necessarily a global variable, and smoothness of the E field ensures the oscillations are symmetric. The smoothness of E can break down at low N_x due to Gibbs Phenomena. Typically, however, the charge distribution is sufficiently smooth that this is not a problem at large enough N_x.

We implement both field interpolators and compare them for plasma oscillations. We see that Fourier interpolation slightly outperforms the simple averaging interpolation method, but only marginally. We therefore use the simple interpolator for our first-order solver, and the Fourier interpolator for our second-order solver.

2.3 Second Order Method: Accuracy Analysis, Stability Analysis, and Modified Equation

So far, we have a second-order expression for the fluxes and the electric field. However, our values of f at the edge of the cells have been first-order. To remedy this, we recall our second-order expressions for the edge values,

$$f^+_{i+1/2,j}(t) = f_{i+1/2,j+1/2} - \frac{\Delta v}{2}\frac{\partial f}{\partial v}(x_{i+1/2}, v_j) + O(h^2), \tag{50}$$

and

$$f^-_{i+1/2,j}(t) = f_{i+1/2,j-1/2} + \frac{\Delta v}{2}\frac{\partial f}{\partial v}(x_{i+1/2}, v_j) + O(h^2). \tag{51}$$

We must choose how to approximate the derivatives of f. The obvious (but wrong) choice is to use that

$$\begin{aligned} f_{i+1/2,j+1/2} - f_{i+1/2,j-1/2} &= \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} dx \int_{v_j}^{v_{j+1}} \frac{f(x,v)-f(x,v-\Delta v)}{\Delta v} \\ &= \int_{v_j}^{v_{j+1}} \frac{f(x_{i+1/2},v)-f(x_{i+1/2},v-\Delta v)}{\Delta v} + O(\Delta x^2) \\ &= O(h^2) + \int_{v_j}^{v_{j+1}} \frac{\partial f}{\partial v}(x_{i+1/2},v) \\ &= O(h^2) + \frac{\partial f}{\partial v}(x_{i+1/2},v_j)\Delta v, \end{aligned} \tag{52}$$

to obtain that

$$\frac{\partial f}{\partial v}(x_{i+1/2},v_j) = O(h) + \frac{f_{i+1/2,j+1/2} - f_{i+1/2,j-1/2}}{\Delta v}. \tag{53}$$

We can plug this into Equations (50) and (51) to get that

$$f^+_{i+1/2,j}(t) = f^-_{i+1/2,j}(t) = \frac{f_{i+1/2,j+1/2} + f_{i+1/2,j-1/2}}{2} + O(h^2). \tag{54}$$

Thus, using this choice for the derivative leads to the unstable central difference scheme we ruled out earlier. We must use a different scheme for the derivatives. The other choices are

$$\left(\frac{\partial f}{\partial x}(x_{i+1/2},v_j)\right)^- = \left(\frac{\partial f}{\partial x}(x_{i+1/2},v_{j-1})\right) + O(\Delta v) = \frac{f_{i+1/2,j-1/2} - f_{i+1/2,j-3/2}}{\Delta v} + O(\Delta v), \tag{55}$$

$$\left(\frac{\partial f}{\partial x}(x_{i+1/2},v_j)\right)^+ = \left(\frac{\partial f}{\partial x}(x_{i+1/2},v_{j+1})\right) + O(\Delta v) = \frac{f_{i+1/2,j+3/2} - f_{i+1/2,j+1/2}}{\Delta v} + O(\Delta v). \tag{56}$$

The ± denotes that the value of the derivative depends only on the values of f to the right or left of the edge. Consistent with upwinding, it would make sense to use $\left(\frac{\partial f}{\partial x}\right)^{\pm}$ in the approximation of $f^{\pm}$. Indeed, this leads to an analog of the Beam-Warming scheme without artificial diffusion. A more accurate and less dispersive method, however, is to average the symmetric (Lax-Wendorff-like) choice of the derivative with the left/right derivatives of equations (56) and (57). This corresponds to a Fromm-like scheme without artificial diffusion. Averaging Equations (55) and (53), we therefore get the Fromm-like choices

$$f^-_{i+1/2,j} = f_{i+1/2,j-1/2} + \frac{f_{i+1/2,j+1/2} - f_{i+1/2,j-3/2}}{4} + O(h^2), \tag{57}$$

$$f^+_{i+1/2,j} = f_{i+1/2,j+1/2} - \frac{f_{i+1/2,j+3/2} - f_{i+1/2,j-1/2}}{4} + O(h^2). \tag{58}$$

As before, the choice between $f^{\pm}$ comes down to upwinding. That is, when $U' > 0$, we use f^- to get the flux and when $U' < 0$ we use f^+ to get the flux. Hence, the second-order accurate, upwinded, flux is

$$\begin{aligned} \bar{F}^v_{i+1/2,j} = U'(x_{i+1/2}) &\left(\frac{5f_{i+1/2,j-1/2} + 5f_{i+1/2,j+1/2} - f_{i+1/2,j-3/2} - f_{i+1/2,j+3/2}}{8}\right) \\ - |U'|(x_{i+1/2}) &\left(\frac{-3f_{i+1/2,j-1/2} + 3f_{i+1/2,j+1/2} + f_{i+1/2,j-3/2} - f_{i+1/2,j+3/2}}{8}\right). \end{aligned} \tag{59}$$

Similarly, we have that

$$\begin{aligned} \bar{F}^x_{i,j+1/2} = K'(v_{j+1/2}) &\left(\frac{5f_{i-1/2,j+1/2} + 5f_{i+1/2,j+1/2} - f_{i-3/2,j+1/2} - f_{i+3/2,j+1/2}}{8}\right) \\ - |K'|(v_{j+1/2}) &\left(\frac{-3f_{i-1/2,j+1/2} + 3f_{i+1/2,j+1/2} + f_{i-3/2,j+1/2} - f_{i+3/2,j+1/2}}{8}\right). \end{aligned} \tag{60}$$

At this point, we have a second-order accurate semi-discrete scheme. Adding artificial diffusion to this scheme (which gives a 2D Fromm-like scheme) is quite cumbersome. Thus, for the time integration, we use RK(2), which is second-order in time. This gives a fully discrete, second-order (in both space and time) numerical scheme. To analyze the numerical scheme, it is convenient define $\bar{F}^x_{i,j+1/2}[f^n]$ and $\bar{F}^v_{i,j+1/2}[f^n]$ as the fluxes given by equation (59) and (60) at time $t = n\Delta t$. This includes that U itself is a function of f^n for the true Vlasov equation. We define

$$
\begin{aligned}
\mathcal{F}_{i+1/2,j+1/2}[f^n] &= -\left(\frac{\bar{F}^x_{i+1,j+1/2}[f^n] - \bar{F}^x_{i,j+1/2}[f^n]}{\Delta x} + \frac{\bar{F}^v_{i+1/2,j+1}[f^n] - \bar{F}^v_{i+1/2,j}[f^n]}{\Delta v} \right) \qquad (61) \\
&= -\frac{K'(v_{j+1/2})}{8\Delta x} \left(-f_{i+5/2,j+1/2} + 6f_{i+3/2,j+1/2} - 6f_{i-1/2,j+1/2} + f_{i-3/2,j+1/2} \right) \\
&- \frac{U'(x_{i+1/2})}{8\Delta v} \left(-f_{i+1/2,j+5/2} + 6f_{i+1/2,j+3/2} - 6f_{i+1/2,j-1/2} + f_{i+1/2,j-3/2} \right) \\
&+ \frac{|K'(v_{j+1/2})|}{8\Delta x} \left(-f_{i+5/2,j+1/2} + 4f_{i+3/2,j+1/2} - 6f_{i+1/2,j+1/2} + 4f_{i-1/2,j+1/2} - f_{i-3/2,j+1/2} \right) \\
&+ \frac{|U'(v_{i+1/2})|}{8\Delta v} \left(-f_{i+1/2,j+5/2} + 4f_{i+1/2,j+3/2} - 6f_{i+1/2,j+1/2} + 4f_{i+1/2,j-1/2} - f_{i+1/2,j-3/2} \right).
\end{aligned}
$$

Then the RK(2) scheme reads

$$f^{n+1}_{i+1/2,j+1/2} = f^n_{i+1/2,j+1/2} + \Delta t \mathcal{F}_{i+1/2,j+1/2}[f^n + \frac{\Delta t}{2}\mathcal{F}[f^n]]. \qquad (62)$$

To show this is second-order in time, we note that the Vlasov equation reads

$$\frac{d}{dt} f(t)_{i+1/2,j+1/2} = \mathcal{F}_{i+1/2,j+1/2}[f^n(t)] + O(h^2). \qquad (63)$$

Ignoring the $O(h^2)$ term (which should stay $O(h^2)$ for any stable method), this is just an ODE for the vector of variables $f_{i+1/2,j+1/2}(t)$. To prove the RK(2) scheme is second-order accurate, we Taylor expand

$$f(t + \frac{\Delta t}{2})_{i+1/2,j+1/2} = f(t)_{i+1/2,j+1/2} + \mathcal{F}_{i+1/2,j+1/2}[f^n(t)]\frac{\Delta t}{2} + O(\Delta t^2). \qquad (64)$$

This implies by Taylor series that

$$\mathcal{F}_{i+1/2,j+1/2}[f^n + \frac{\Delta t}{2}\mathcal{F}[f^n]] = \mathcal{F}_{i+1/2,j+1/2}[f(t + \frac{\Delta t}{2})] + O(\Delta t^2). \qquad (65)$$

Taylor series approximating $f(t + \Delta t)$ and $f(t)$ around $t + \Delta t/2$ and plugging this approximation, we get that

$$f_{i+1/2,j+1/2}(t + \Delta t) - f_{i+1/2,j+1/2}(t) = \Delta t F[f(t + \frac{\Delta t}{2})] + O(\Delta t^3) = \mathcal{F}_{i+1/2,j+1/2}[f^n + \frac{\Delta t}{2}\mathcal{F}[f^n]] + O(\Delta t^3), \qquad (66)$$

which is the desired statement that the discrete timestep is second-order accurate. Thus, the scheme we derived is second-order accurate. For any (periodic) function g, we know that $\sum_{ij} \mathcal{F}_{i+1/2,j+1/2}[g] = 0$ (this follows trivially from the definition of $\mathcal{F}$). We thus learn that for our scheme

$$\int dxdv\ f^{n+1} = \sum_{i,j} f^{n+1}_{i+1/2,j+1/2} = \sum_{i,j} f^n_{i+1/2,j+1/2} = \int dxdv\ f^n. \qquad (67)$$

That is, the total particle number is preserved. What is no longer true, which we verify with simulations, is that $f^{n+1}_{i+1/2,j+1/2}$ is always positive. Indeed, we see that our second-order scheme introduces numerical oscillations into the solution of f in the direction of advection.

To check that our method is stable, we are forced to perform Von Neumann stability analysis. We first derive the stability region of RK(2). Indeed, consider the ODE $\dot{u} = au$ with a a constant. Using RK(2) to integrate this ODE, we have that

$$u^{n+1} = u^n(1 + \Delta t(a + \frac{\Delta t}{2}a^2)). \qquad (68)$$

The amplification factor is $\rho = |1 + \hat{a} + \frac{\hat{a}^2}{2}|$ where $\hat{a} = a\Delta t$. Provided $\rho < 1$, the ODE integration is stable. Now we look at the numerical dispersion relation for $\mathcal{F}[f]$. Suppose that $(f_{i+1/2,j+1/2}) = \mathbf{v}_{k,k'} = e^{ik(x_{i+1/2}) + ik'(v_{j+1/2})}$ where

k, k' are multiples of π. Then

$$\begin{aligned} e^{e^{-ik(x_{i+1/2})-ik'(v_{j+1/2})}}\mathcal{F}_{i+1/2,j+1/2}[\mathbf{v}_k] = & -\frac{iK'(v_{j+1/2})}{4\Delta x}\left(6\sin(k\Delta x) - \sin(2k\Delta x)\right) \approx -iK'(v_{j+1/2})\left(k + \frac{k^3\Delta x^2}{12}\right) \\ & - \frac{iU'(x_{i+1/2})}{4\Delta v}\left(6\sin(k'\Delta v) - \sin(2k'\Delta v)\right) \approx -iU'(v_{i+1/2})\left(k' + \frac{k'^3\Delta v^2}{12}\right) \\ & + \frac{|K'(v_{j+1/2})|}{4\Delta x}\left(-\cos(2\Delta xk) + 4\cos(\Delta xk) - 3\right) \approx 0 \\ & + \frac{|U'(v_{i+1/2})|}{4\Delta v}\left(-\cos(2\Delta vk') + 4\cos(\Delta vk') - 3\right) \approx 0, \end{aligned}$$

where the approximations are valid to third order in h. When K' and U' are constant (or at least slowly varying) $\mathbf{v}_k$ is an eigenvector of $\mathcal{F}$. In this case, we have the approximate dispersion relation

$$\frac{1}{\mathbf{v}_k}\mathcal{F}[\mathbf{v}_k] \approx -iK'(k + \frac{k^3\Delta x^2}{12}) - iU'(k + \frac{k^3\Delta v^2}{12}) \tag{69}$$

This implies the modified equation

$$\partial_t f + K'\partial_x U'\partial_x f = \mu\partial_x^3 f + \mu'\partial_v^3 f \tag{70}$$

where the dispersion coefficients are $\mu = \frac{\Delta x^2 K'}{12}$ and $\mu' = \frac{\Delta v^2 U'}{12}$. This verifies that our semi-discrete scheme is second-order, and that our scheme is dispersive. This dispersive character causes waves to propagate too quickly, causing oscillations in front of $f(t)$ when $f(t)$ is numerically integrated. We indeed observe this.

While the eigenvalues of $\mathcal{F}$ are difficult to get, the only case we have to worry about for stability analysis is that U' and K' are everywhere equal to their maxima (or minima). Therefore, for the purposes of stability analysis, we assume that $K' = a$ and $U' = b$ are constants. WLOG, we assume $a, b > 0$. Then the eigenvalues of $\mathcal{F}$ can be computed as

$$c_k = \frac{1}{\mathbf{v}_k}\mathcal{F}[\mathbf{v}_k] = \frac{a}{\Delta x}\left(-i6\sin(\theta) + i\sin(2\theta) - \cos(2\theta) + 4\cos(\theta)\right) + \frac{b}{\Delta v}\left(-i6\sin(\phi) + i\sin(2\phi) - \cos(2\phi) + 4\cos(\phi)\right), \tag{71}$$

where we defined $\theta = k\Delta x$ and $\phi = k'\Delta v$. From the stability condition of RK(2), we learn that our numerical scheme is stable only when $|1 + \Delta t c_k + \frac{\Delta t^2 c_k^2}{2}| < 1$ for all k. With some tedious algebra, we see that our numerical scheme is stable iff the Courant number $C = (\frac{\Delta ta}{\Delta x} + \frac{\Delta tb}{\Delta v}) < 1$. We interpret a, b in this expression to be the maximum of $|U'|$ and $|K'|$. Generally, we choose a timestep such that $C = .8$.

We note that this stability analysis is somewhat complicated by the fact that $U(x,t)$ is time-dependent. For the full Vlasov equation, Δt must be chosen at each timestep to make $C < 1$. However, RK(2) uses two forward Euler steps, each with the same Δt. It could hold that $\max(|U^{n+1/2}|)$ is significantly larger than $\max(|U^n|)$. In this case, the RK(2) step is no longer stable since we must necessarily base the timestep on $\max(|U^{n+1/2}|)$, not $\max(|U^n|)$. This issue can be solved by computing $\max(|U^{n+1/2}|)$ and aborting the Runge-Kutta step if $\max(|U^{n+1/2}|)$ exceeds a maximum tolerance. The Runge-Kutta step would then need to be repeated with a smaller timestep. To avoid too many repeated timesteps, dt at each timestep such be chosen such that the Courant factor is not close to unity.

2.4 Slope Limiters and Monotonicity Analysis

Because our second method is not positivity-preserving (hence monotonicity-preserving), we need to implement flux limiters. At first-order, we can approximate (reconstruct) f inside the cell $[x_i, x_{i+1}] \times [v_i, v_{i+1}]$ as a piecewise linear expression $f(x,v) = f_{i+1/2,j+1/2} + (x - x_{i+1/2})\Delta^x f_{i+1/2,j+1/2} + (v - v_{j+1/2})\Delta^v f_{i+1/2,j+1/2}$ where $\Delta^x f$ and $\Delta^v f$ are the slopes of f. Comparing this expression with Equation (57) and its analog for the x direction, our second-order scheme predicts that

$$\left(\Delta^x f_{i+1/2,j+1/2}\right)_C = \frac{f_{i+3/2,j+1/2} - f_{i-1/2,j+1/2}}{2\Delta x}, \tag{72}$$

and

$$\left(\Delta^v f_{i+1/2,j+1/2}\right)_C = \frac{f_{i+1/2,j+3/2} - f_{i+1/2,j-1/2}}{2\Delta v}. \tag{73}$$

These are the central slopes. The other slope choices are

$$\left(\Delta^x f_{i+1/2,j+1/2}\right)_R = \frac{f_{i+3/2,j+1/2} - f_{i+1/2,j+1/2}}{\Delta x}, \tag{74}$$

and

$$\left(\Delta^x f_{i+1/2,j+1/2}\right)_L = \frac{f_{i+1/2,j+1/2} - f_{i-1/2,j+1/2}}{\Delta x}. \tag{75}$$

We have similar expressions for the v slopes. Using some choice of slopes, we can reconstruct $f^{\pm}_{i,j+1/2} = f^{\pm}_{i\pm 1/2,j+1/2} \mp \frac{\Delta x}{2}(\Delta^x f_{i\pm 1/2,j})$ and similarly for $f^{\pm}_{i+1/2,j}$.

To understand why using the central slopes everywhere is problematic, suppose that $f_{i+3/2,j+1/2} > f_{i+1/2,j+1/2} > f_{i-1/2,j+1/2}$. Then it typically holds that $f_{i-1/2,j+1/2} < f^-_{i+1,j+1/2} < f_{i+1/2,j+1/2} < f^+_{i+1,j+1/2} < f_{i+3/2,j+1/2}$. This amounts to the statement that $\Delta^x f_{i+1/2,j+1/2} < \min(2(\Delta^x f_{i+1/2,j+1/2})_L, 2(\Delta^x f_{i+1/2,j+1/2})_R)$. If this condition is violated, then either we are at a smooth extremum of f (in the x direction), or we have overestimated the slope of f. This latter case will cause spurious oscillations in our code unless addressed. For example, momentarily dropping the v dependence of f, suppose that $f(x, t = 0)$ is a Heaviside function such that $f_{i-1/2} = f_{i-3/2} = \ldots = 0$ and $f_{i+1/2} = f_{i+3/2} = \ldots = 1$. Then we know that $\Delta^x f = 0$ for all cells. However, our central difference scheme for the slopes would predict that $\Delta^x f_{i-1/2} = 1/2$, hence that $f^+_i = \frac{1}{4}$. If $K' > 0$, then for any timestep Δt, $f_{i-1/2}(\Delta t) < 0$. This is clearly a problem since we wish for our numerical scheme to preserve positivity.

To solve this problem, we define the moncen limited slope

$$\left(\Delta^x f_{i+1/2,j+1/2}\right)_{\text{moncen}} = \min\left(\left(\Delta^x f_{i+1/2,j+1/2}\right)_C, 2\left(\Delta^x f_{i+1/2,j+1/2}\right)_R, 2\left(\Delta^x f_{i+1/2,j+1/2}\right)_L\right), \tag{76}$$

where the minimum means the term with the smallest absolute value. We also define $\left(\Delta^x f_{i+1/2,j+1/2}\right)_{\text{moncen}} = 0$ if $\left(\Delta^x f_{i+1/2,j+1/2}\right)_R$ and $\left(\Delta^x f_{i+1/2,j+1/2}\right)_L$ have different signs. We also define an analogous moncen slope for $\Delta^v f$. By definition, as long as $f_{i+1/2,j+1/2} \geq 0$, all edge values of f are nonnegative. Thus, the linear reconstruction of f always creates a positive function.

Typically, one implements a smooth extremum detector to turn off the flux limiting at smoothly varying extremum. This prevents the numerical solution to f from being clipped near smooth extreme points. We choose not to implement such a detector, but one could be implemented in the future.

We implement the moncen slope limiter as a replacement to the central slope we used for the second-order scheme. In particular, the time integration is still performed with RK(2) using the flux limiter on both the predictor and the corrector step. Notice that if $f_{i+1/2,j+1/2}$ is locally smooth, then moncen slope limiter reduces to our prior, second-order scheme. Thus away from any sharp changes and extremum the moncen slope limiter combined with our $RK(2)$ scheme remains second-order. Further, because the edge values of f can only be reduced, the scheme remains stable for $C < 1$.

The moncen limiter combined with RK(2) is not designed to preserve positivity, although it does for $C < 1/2$ (we will show this). In practice, moncen with RK(2) seems to work very well at preserving positivity. A slope limiter designed to preserve positivity is minmod, whereby the slopes of f are given by

$$\left(\Delta^x f_{i+1/2,j+1/2}\right)_{\text{minmod}} = \min\left(\left(\Delta^x f_{i+1/2,j+1/2}\right)_R, \left(\Delta^x f_{i+1/2,j+1/2}\right)_L\right), \tag{77}$$

and similarly for v. It is trivial to show for the minmod slope limiter that if f is initially positive, then f remains positive provided that $C < 1$. Thus, minmod with RK(2) is a stable, positivity preserving scheme. In smooth regions, minmod is second-order accurate. We implement the minmod limiter and compare it to the moncen limiter with RK(2). We find that minmod has a strong tendency to clip to extreme values of f, leading to less accurate solutions.

An alternative scheme, which is similar to a MUSCL-Hancock scheme without smooth extremum detection or artificial diffusion, modifies the corrector step in RK(2). In this scheme, we use the limited slopes in the predictor step, while in the corrector step we take $f^{n+1/2,\pm}_{i,j+1/2} = f^{n+1/2}_{i\pm,j+1/2}$. That is, the corrector step uses a first-order approximation of the edge value. This renders the method first-order overall. This method is positivity preserving (and hence stable) as long as $C < 1$. We show positivity preservation in the nontrivial case when a moncen flux limiter is used. This also shows that our RK(2) scheme with the moncen limiter is positivity preserving for $C < 1/2$.

Consider the advection equation with the moncen slope limiter

$$\partial_t f_{i+1/2} + \frac{a}{\Delta x}\left(f^+_{i+1} - f^+_i\right) = 0. \tag{78}$$

where WLOG, we take $a > 0$. By definition, $f^-_i = f_{i+1/2} + \frac{1}{2}(\Delta f_{i+1/2})_{\text{moncen}}$. Suppose that $f_{i-5/2} < f_{i-3/2} < f_{i+1/2} < f_{i+3/2}$. Then trivially we have that $f^-_{i+1/2} > f^-_i > f_{i+1/2}$ for all the i indices. By moncen limiting, we also have that $f^{-}_{i+1} < 2f_{i+1/2} - f_{i-1/2} < 2f_{i+1/2}$. Consider the predictor step in RK(2). We have that (for $C < 1$)

$$f^n_{i+1/2} \geq f^{n+1/2}_{i+1/2} = f^n_{i+1/2} - \frac{C}{2}f^{-,n}_{i+1} + \frac{C}{2}f^{-,n}_{i} \geq f^n_{i+1/2}(1-C) + Cf^n_{i-1/2} \geq f^n_{i-1/2}, \tag{79}$$

$$(1-C)f^n_{i-1/2} + Cf^n_{i-3/2} \leq f^{n+1/2}_{i-1/2} = f^n_{i-1/2} - \frac{C}{2}f^{-,n}_{i} + \frac{C}{2}f^{-,n}_{i-1} \leq f^n_{i-1/2}(1-\frac{C}{2}) + \frac{C}{2}f^n_{i-3/2} \leq f^n_{i-1/2}. \tag{80}$$

Hence we see that the forward Euler step with RK(2) is montonicity and positivity preserving since $f^{n+1/2}_{i+1/2} > f^{n+1/2}_{i-1/2} \geq 0$. For $C < 1/2$, these inequalities also show the corrector step in $RK(2)$ leaves f positive since $f^{n+1/2}_{i+1/2} \leq f^n_{i+1/2}$.

For the corrector step in the MUSCL-Hancock-like scheme,

$$f^{n+1}_{i+1/2} = f^n_{i+1/2} - Cf^{n+1/2,-}_{i+1} + Cf^{n+1/2,-}_{i} = f^n_{i+1/2} - Cf^{n+1/2}_{i+1/2} + Cf^{n+1/2}_{i-1/2} \geq (1-C)f^n_{i+1/2}, \tag{81}$$

which is positive for $C < 1$. Thus, the MUSCL-Hancock-like scheme is positivity preserving.

We implement the MUSCL-Hancock-like scheme using the moncen flux limiter. We then compare this to the RK(2) scheme with minmod and moncen flux limiting. We find that our slope-limited RK(2) methods greatly outperform the MUSCL-Hancock like scheme, which is comparable to our first-order method.

2.5 The Program

We implement our programs in a Jupyter notebook named "Vlasov Solver Notebook". Using our various methods, we simulate the free-Hamiltonian system and the simple harmonic oscillator. We plot the solution, and the L^2 norm of the numerical solution, which is a proxy for the numerical error. We then use the true Vlasov solver to simulate the two-stream instability and plasma oscillations. All of our methods work as expected, with all our examples having the right physics. We use a Jupyter notebook to run the simulations and add commentary on what the reader is seeing. This is contained in the GitHub. The notebook is already compiled, but can be modified by downloading the file and using any .ipynb interpreter. Adroit is capable of running such files for example.

2.6 Numerical Experiments

The code for this section is in a Jupyter notebook named "Numerical Experiments."

We have developed a first-order method, a second-order method, and a second-order method with slope-limiting. Of the slope limiters, moncen works the best, so we will always assume the slope-limited method is moncen-limited.

There are no exact solutions to the true nonlinear Vlasov equation, but there are solutions to the Vlasov equation with a prescribed Hamiltonian. Let us analyze the error in the three methods presented. We always assume that $N_x = N_v = N$. We also assume that the Courant number $C = dx/2$ for all runs is 0.8.

We use the SHO Hamiltonian $H = v^2/2 + x^2/2$, which has the known solution $f(x, v, t) = f(x\cos(t) - v\sin(t), v\cos(t) + x\sin(t), 0)$. The error $\epsilon(t)$ is defined by $\epsilon(t) = ||f_{num} - f_{true}||_{L^1}$ where f_{num} is the numerical solution and f_{true} is the true solution. We use the initial condition $f(x, v, t = 0) = e^{-20(v-.3)^2-20x^2}$. We run the SHO test for $N \in \{32, 64, 128, 256, 512\}$ with $t \in [0, 2\pi]$ and record $\epsilon(t)$ for each of the methods. We also record the total computation time for each run. We then plot $\epsilon(2\pi)$ against $h = dx = dv = 2dt/C = 2/N$. We also plot the compute time, call is $C(2\pi)$, defined to be the time it takes to integrate the Vlasov equation up to time 2π, against N.

For all the methods, there are $O(N^2)$ operations per timestep, and $O(N)$ timesteps to integrate the Vlasov equation. Thus we expect that $C(2\pi)$ scales as N^3 for all the methods. With N fixed, $C(2\pi)$ should be at least twice as big for the second-order method as for the first, since there are two Euler solves per timestep. Adding a slope limiter adds another factor of 1.5 to the computational complexity because there are $O(N^2)$ comparisons per timestep needed for the limiting.

For the first-order method. we expect that $\epsilon(2\pi) \sim h$ for h sufficiently small. For the second-order method, $\epsilon(2\pi)$ should scale like h^2. When moncen slope limiting is introduced to the second-order method, we should similarly have that $\epsilon(2\pi) \sim h^2$. The limiting adds another factor of about 1.5 to the computational complexity because of the $O(N^2)$ comparisons needed at each step.

We see clearly from Figure 1 that the second-order methods are very comparable in accuracy. The first-order method has its error dominated by numerical diffusion, and it shows from the monotonic increase in the error. The second-order method error is a superposition of a linear diffusion error and "wobble," which causes the error to oscillate. The

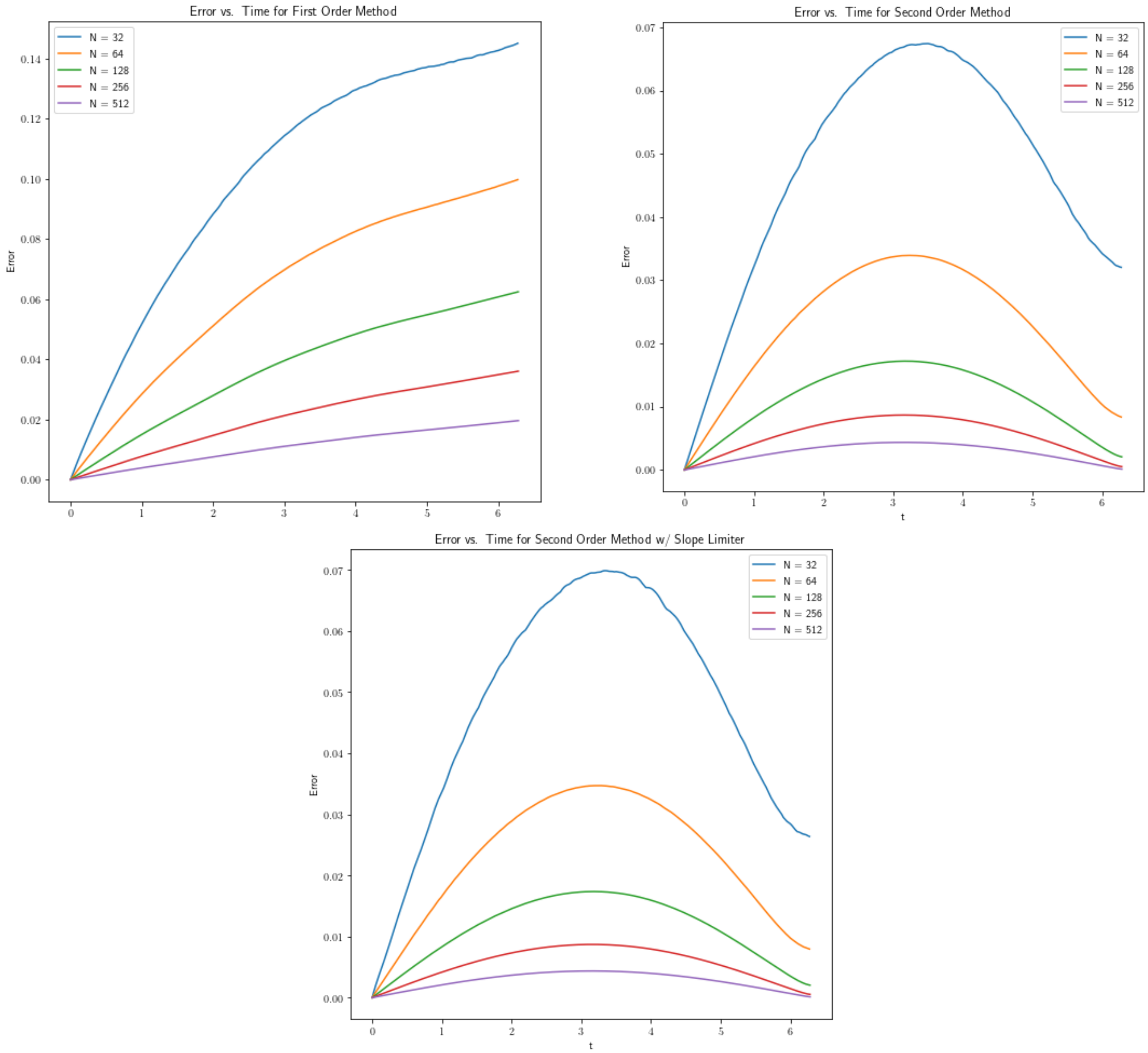

Figure 1: Error vs. Time for The First-Order and Second-Order Methods

”wobble” comes about because the numerical solution over and undershoots the true solution. The best gauge of the accuracy of the second-order solution is at the period time $t = 2\pi$.

In Figure 2, we plot $\epsilon(2\pi)$ vs. h in a loglog plot. We see for the first-order method, $\epsilon(2\pi)(h)$ goes like h as h is made small. As h becomes larger, $\epsilon(2\pi)(h)$ deviates from the ideal fit. This is expected since the error only scales as h for h sufficiently small. Nonetheless, the first-order method is indeed first order in h.

For both of the second-order methods, $\epsilon(2\pi)(h)$ perfectly scales as h^2. For the unlimited second-order method, this is expected since the method is second-order in both $\Delta x, \Delta v$, and Δt. For the moncen limited run, second-order behavior is still expected since the initial condition is smooth and thus limiting is mostly turned off.

Figure 3 shows the compute time $C(2\pi)$ vs N as a loglog plot. We see that roughly $C(2\pi) \sim N^3$, but the fits are far from exact. I suspect the deviation comes from memory management on my device, but it is hard to say. We end up seeing that the second-order method is about 3 slower than the first-order method. This makes sense since the slopes for the second-order method require more operations to compute compared to the first-order method, and there are two Euler steps for a timestep in the second-order method. The slope-limited method is about 1.5 times slower than the unlimited method.

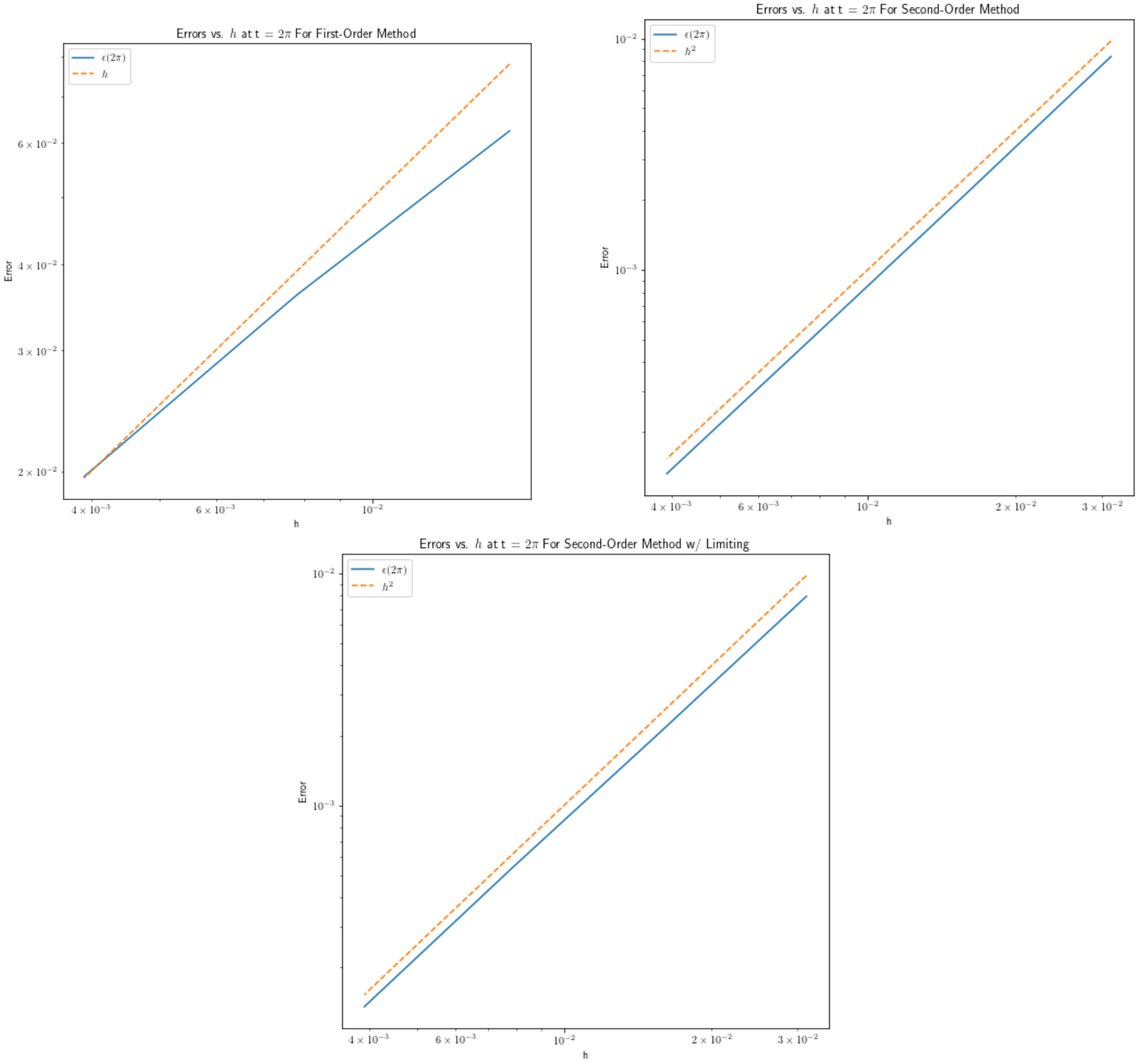

Figure 2: $\epsilon(2\pi)$ vs. h for The First-Order and Second-Order Methods with ideal fit as dashed lines. Ideal fit for the first order method is h, and h^2 for the second order methods.

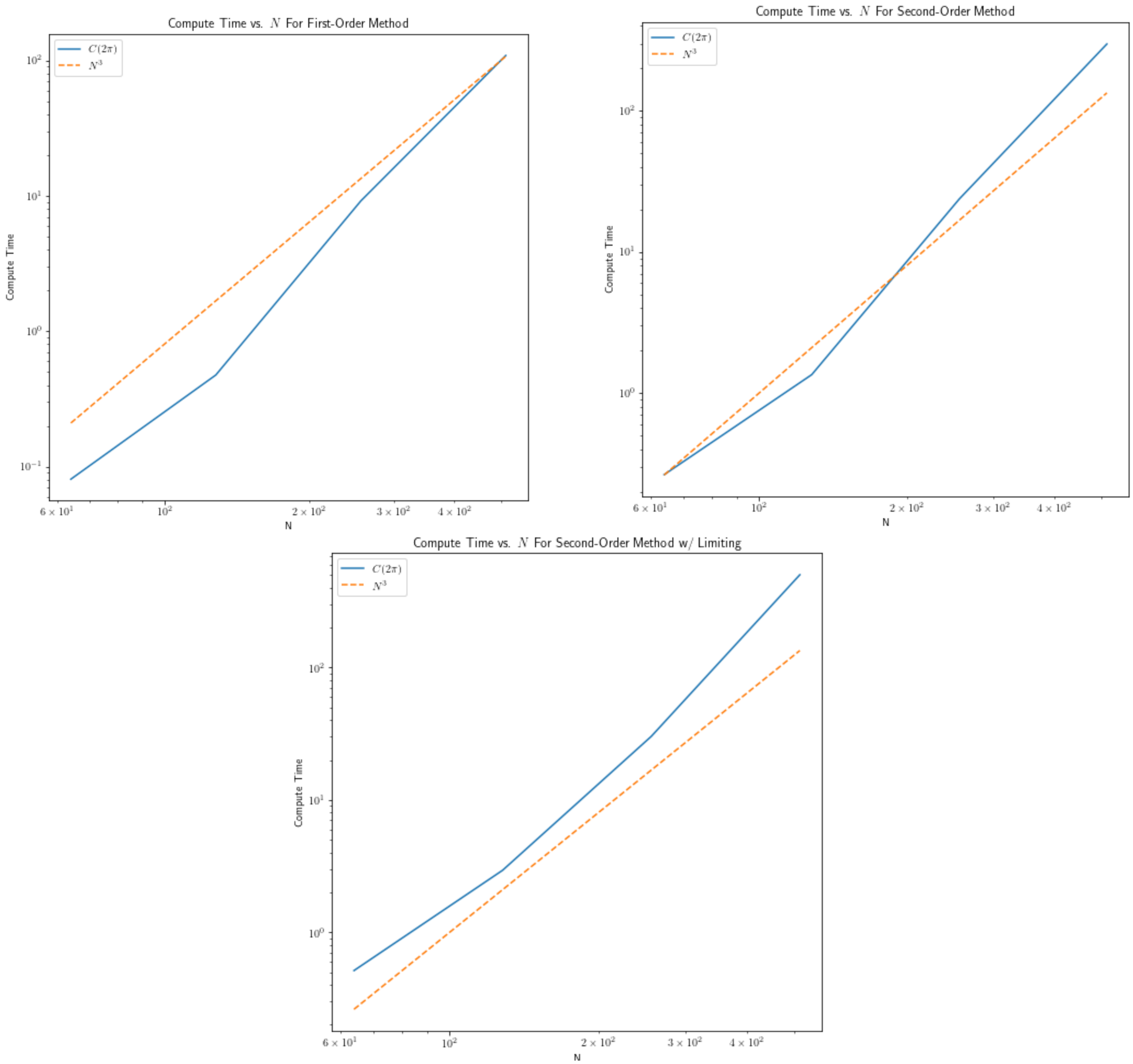

Figure 3: $C(2\pi)$ vs. N for The First-Order and Second-Order Methods with ideal fit of N^3 as a dashed line.