

Efficient dynamical low-rank approximation for the Vlasov-Ampère-Fokker-Planck system [☆]

Jack Coughlin, Jingwei Hu ^{*}

Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

ARTICLE INFO

Article history:

Received 22 April 2022
Received in revised form 27 July 2022
Accepted 29 August 2022
Available online 5 September 2022

Keywords:

Dynamical low-rank integrator
Vlasov-Ampère-Fokker-Planck model
High-field limit
Convolution
Implicit-explicit scheme

ABSTRACT

Kinetic equations are difficult to solve numerically due to their high dimensionality. A promising approach for reducing computational cost is the dynamical low-rank algorithm, which decouples the dimensions of the phase space by proposing an ansatz as the sum of separable (rank-1) functions in position and velocity respectively. The fluid asymptotic limit of collisional kinetic equations, obtained in the small-Knudsen number limit, admits a low-rank representation when written as $f = Mg$, where M is the local Maxwellian, and g is low-rank. We apply this decomposition to the Vlasov-Ampère-Fokker-Planck equation of plasma dynamics, considering the asymptotic limit of strong collisions and electric field. We implement our proposed algorithm and demonstrate the expected improvement in computation time by comparison to an implementation that evolves the full solution tensor f . We demonstrate that our algorithm can capture dynamics in the fluid regime with very low rank, thereby efficiently capturing the asymptotic fluid limit. Moreover we find that a modest rank of between 15 and 20 is sufficient to capture kinetic effects on the problems we consider, showing that the approach is applicable and efficient across a range of regimes.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Magnetohydrodynamics and multi-fluid systems of equations provide reasonable descriptions of plasma dynamics across a wide range of parameter regimes. However, in situations where one or more of the particle species' phase space distributions is far from a Maxwellian, fluid models can fail to capture relevant physics. The Vlasov equation, when coupled with Maxwell's equations of electrodynamics, provides a more complete description of plasma dynamics in these regimes [11]. However, the numerical solution of kinetic models is quite costly in 2 or 3 dimensions, since they are posed over 4 or 6 phase space dimensions, respectively. This prompts the search for computational algorithms which can accelerate the solution of kinetic equations.

One promising approach for accelerated kinetic algorithms comes from the recognition that there is low-rank structure in certain solutions of kinetic equations. By low-rank structure, we mean that an approximation of the following form can be successful:

[☆] This work was partially supported by NSF CAREER grant DMS-2153208, NSF CDS&E grant CBET-1854829, and AFOSR grant FA9550-21-1-0358. The authors would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the program "Frontiers in kinetic theory - KineCon 2022" where part of work on this paper was undertaken.

^{*} Corresponding author.

E-mail addresses: johnbc@uw.edu (J. Coughlin), hujw@uw.edu (J. Hu).

$$f(x, v, t) \approx \sum_{ij}^r X_i(x, t) S_{ij}(t) V_j(v, t). \quad (1.1)$$

Such an approximation will be useful only if the numerical rank, r , is small compared to the number of degrees of freedom N_x and N_v . The inspiration for this form of approximation, called a low-rank approximation, comes from linear algebra and the need to deal effectively with extremely large data matrices. The idea is to capture most of the action of a data matrix with a low-rank approximation which can require vastly less storage. Bounds on the quality of the approximation are available in various norms. Perhaps the best-known approximation of this kind is the truncated Singular Value Decomposition, which is known to provide the best rank r approximation to a given matrix in the spectral norm [22].

Our setting is time-dependent kinetic equations, so it is not enough to be able to compress a given phase space distribution. One must also be able to evolve the distribution f in an approximate form. This is made possible by the theory of dynamical low-rank approximation, which has been studied in the matrix and tensor contexts in [14], [15]. This method advances the representation (1.1) by updating the bases X_i, V_j , and the matrix of singular values S_{ij} at each timestep. The point is to never form the full product of all three factors but to evolve the factorized form directly. A crucial innovation in this field that we make use of is the projector-splitting integrator of [17]. This integrator enables a robust dynamical low-rank method which is insensitive to “overapproximation”, or vanishingly small singular values in the approximation. Another such integrator with comparable robustness properties is the recently proposed “unconventional” dynamical low-rank integrator of [3].

In the numerical analysis of kinetic equations, the dynamical low-rank method has recently been applied to many problems. Here we mention a few representative ones: the Vlasov equation [8], [7], linear transport equation [9], [18], [4], Boltzmann equation [12], and BGK equation [6], [10]. In particular, the last contribution [10] is significant for preserving the asymptotic fluid limit of the collisional BGK equation, which inspires our current work.

In this paper we present a dynamical low-rank algorithm for the solution of a model equation for the Vlasov equation with collisions. Collisions with a Coulomb interaction potential can be described by an integro-differential operator with a drift term and a diffusion term, i.e., the Landau operator [11], [1], [20], [23] or simplified Fokker-Planck type operator [5]. We therefore consider the non-magnetic Vlasov equation with a linear Fokker-Planck collision operator, in the so-called “high-field” limit. This scaling was introduced in [19] as a model for the semiconductor Boltzmann equation, was treated numerically in [2], and in [13] with an asymptotic-preserving scheme. It retains key properties of the full Vlasov-Landau-Poisson equation, specifically the diffusive collision operator and nonlinear coupling between f and the electric field. As such, it provides an interesting test case for the dynamical low-rank method applied to collisional plasma equations. In a nondimensionalized form, the equation reads

$$\partial_t f + v \cdot \nabla_x f + \frac{1}{\epsilon} E \cdot \nabla_v f = \frac{1}{\epsilon} P_{\text{FP}}(f), \quad t > 0, \quad x \in \Omega \subset \mathbb{R}^d, \quad v \in \mathbb{R}^d. \quad (1.2)$$

The function $f(x, v, t)$ is the single-particle probability density function defined over phase space of d spatial and d velocity dimensions. The operator P_{FP} is the linear Fokker-Planck operator

$$P_{\text{FP}}(f) = \nabla_v \cdot (vf + \nabla_v f). \quad (1.3)$$

One can imagine that f describes a population of electrons moving under the influence of their own inertia and electric field. The small parameter $\epsilon > 0$ is a scaling parameter which determines the strength of both collisions and the electric field E . It is analogous to the Knudsen number from the theory of hydrodynamic limits of the Boltzmann equation.

The electric field E is determined self-consistently from the phase space distribution f via Ampère’s law:

$$\partial_t E = -J, \quad t > 0, \quad x \in \Omega, \quad (1.4)$$

where the current density J is defined as

$$J(x, t) = \langle v, f \rangle_v. \quad (1.5)$$

Here we have made use of a notation for the L^2 inner product, which we now define as

$$\langle g, h \rangle_x = \int_{\Omega} gh \, dx, \quad \langle g, h \rangle_v = \int_{\mathbb{R}^d} gh \, dv, \quad \langle g, h \rangle_{xv} = \int_{\Omega} \int_{\mathbb{R}^d} gh \, dx \, dv. \quad (1.6)$$

The initial electric field will be specified via a static background charge density $\eta(x)$. To continue the physical picture of an electron fluid, η may represent a density of ions which do not move on the timescale resolved by (1.2). To be physical, the field E should satisfy Gauss’s law with respect to the density ρ :

$$E = -\nabla_x \phi(x), \quad -\nabla_x^2 \phi(x) = \rho(x, t) - \eta(x), \quad (1.7)$$

where

$$\rho(x, t) = \langle 1, f \rangle_v. \quad (1.8)$$

It is easy to show that if $E(x, t)$ satisfies Ampère's equation (1.4) and satisfies Gauss's law (1.7) at time 0, then it will satisfy (1.7) for all time. Numerically, we initialize $E(x, 0)$ using Gauss's law and a specified background density $\eta(x)$, and then timestep E using Ampère's law.

To recapitulate, in this work we are solving the coupled system

$$\begin{cases} \partial_t f + v \cdot \nabla_x f + \frac{1}{\epsilon} E \cdot \nabla_v f = \frac{1}{\epsilon} \nabla_v \cdot (vf + \nabla_v f), & (x, v, t) \in \Omega \times \mathbb{R}^d \times [0, T] \\ \partial_t E = -J, & (x, t) \in \Omega \times [0, T] \\ E = -\nabla_x \phi, \quad -\nabla_x^2 \phi = \rho - \eta, & x \in \Omega, t = 0. \end{cases} \quad (1.9)$$

Remark 1. The standard choice for coupling the Vlasov equation with electrostatics is to solve the Vlasov-Poisson system, in other words to couple the Vlasov equation with Gauss's law, rather than Ampère's law. In this paper we choose to solve Ampère's law in order to strengthen the connection with previous work, namely [10]. In that paper, the moments of the distribution function, ρ and u , are evolved via a time-dependent moment system. Here, we solve for E —which plays an analogous role to u as the average velocity of the equilibrium distribution—in a similar manner, preferring the time-dependent relation to the equivalent elliptic equation.

However, this choice introduces a first-order-in-time violation of Gauss's law, so solving the Vlasov-Poisson system may be preferable for some applications. We note that the framework applied here functions equally well for solving Gauss's law at each timestep. See Remark 3 for details of how to achieve that within our low-rank framework.

1.1. Asymptotic fluid limit

The limit of (1.9) for very small ϵ is a sort of electrostatic “creeping flow”, in which inertial forces are vanishingly small compared to electrostatic forces. To analyze the limit $\epsilon \rightarrow 0$, we introduce a scaled “local Maxwellian” defined as

$$M(x, v, t) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{|v-E(x,t)|^2}{2}}. \quad (1.10)$$

One should note that, in contrast to the Maxwellian equilibrium of the Boltzmann equation, this function has a uniform density. It is isothermal and the flow velocity is equal to E . It is not hard to show that (1.2) is equivalent to

$$\partial_t f + v \cdot \nabla_x f = \frac{1}{\epsilon} \nabla_v \cdot [M \nabla_v (M^{-1} f)]. \quad (1.11)$$

To see this we expand the right hand side of (1.11):

$$\nabla_v (M^{-1} f) = (\nabla_v M^{-1}) f + M^{-1} \nabla_v f = (v - E) M^{-1} f + M^{-1} \nabla_v f = M^{-1} (v - E + \nabla_v) f. \quad (1.12)$$

Therefore,

$$\nabla_v \cdot [M \nabla_v (M^{-1} f)] = \nabla_v \cdot [(v - E + \nabla_v) f] = \underbrace{\nabla_v \cdot (vf + \nabla_v f)}_{P_{FP}(f)} - \underbrace{(\nabla_v \cdot E) f}_{\text{force term}} - \underbrace{E \cdot \nabla_v f}_{\text{force term}}, \quad (1.13)$$

where we have accounted for both $P_{FP}(f)$ and the force term on the left hand side of (1.2).

The form (1.11) reveals the dominant balance structure of (1.2): the linear Fokker-Planck operator and the electrostatic force term are balanced under this scaling; together they drive f to a local equilibrium. Indeed, when $\epsilon \rightarrow 0$ in (1.11), formally $\nabla_v \cdot [M \nabla_v (M^{-1} f)] \rightarrow 0$ which implies $f \rightarrow \rho M$. To see this, first note that

$$\begin{aligned} \int \nabla_v \cdot [M \nabla_v (M^{-1} f)] \log(M^{-1} f) dv &= \int \nabla_v \cdot [f \nabla_v \log(M^{-1} f)] \log(M^{-1} f) dv \\ &= - \int f |\nabla_v \log(M^{-1} f)|^2 dv \leq 0. \end{aligned} \quad (1.14)$$

If equality holds, as it does in the formal limit $\epsilon \rightarrow 0$, then by positivity of f , $|\nabla_v \log(M^{-1} f)| = 0$ and thus $f = cM$, where c is a function independent of v . Consequently $\nabla_v \cdot [M \nabla_v (M^{-1} f)] = 0$ which implies $\int \nabla_v \cdot [M \nabla_v (M^{-1} f)] \log(M^{-1} f) dv = 0$. Therefore, one has the following equivalent statements:

$$\int \nabla_v \cdot [M \nabla_v (M^{-1} f)] \log(M^{-1} f) dv = 0 \iff f = cM \iff \nabla_v \cdot [M \nabla_v (M^{-1} f)] = 0. \quad (1.15)$$

Finally $\rho = \langle 1, f \rangle_v = c \langle 1, M \rangle_v = c$.

To derive a macroscopic system of (1.11) when $\epsilon \rightarrow 0$, we first take moments $\langle 1, \cdot \rangle_v$, $\langle v, \cdot \rangle_v$ of (1.11) to obtain

$$\partial_t \rho + \nabla_x \cdot J = 0, \quad (1.16)$$

$$\partial_t J + \nabla_x \cdot \langle v \otimes v, f \rangle_v = \frac{1}{\epsilon} (\rho E - J). \quad (1.17)$$

As $\epsilon \rightarrow 0$, one has $J \rightarrow \rho E$ from (1.17). Then (1.16) becomes

$$\partial_t \rho + \nabla_x \cdot (\rho E) = 0, \quad (1.18)$$

which together with Ampère's law (1.4) constitutes the limiting system:

$$\begin{cases} \partial_t \rho + \nabla_x \cdot (\rho E) = 0, & (x, t) \in \Omega \times [0, T] \\ \partial_t E = -\rho E, & (x, t) \in \Omega \times [0, T] \\ E = -\nabla_x \phi, \quad -\nabla_x^2 \phi = \rho - \eta, & x \in \Omega, t = 0. \end{cases} \quad (1.19)$$

The system (1.19) fully determines the behavior of the kinetic system (1.9) in the asymptotic limit $\epsilon \rightarrow 0$. Our numerical scheme is careful to preserve this asymptotic limit at the discrete level. However, trying to design a low-rank scheme that smoothly approaches this limit quickly runs into a problem: $f^0 = \rho M$ is not a low-rank function in x and v , i.e., we cannot write it in the form (1.1) with small r . Recall the definition (1.10), where the cross term $e^{-v \cdot E(x,t)}$ is not low rank. If we require a high rank to resolve the limiting solution f^0 , then we are, in a sense, wasting effort on a kinetic system whose dominant dynamics are described by the much lower-dimensional system (1.19).

To resolve this problem, we can observe that while $f^0 = \rho M$ is not low rank in x and v , $\rho(x, t)$ certainly is. This motivates us to consider a low-rank approximation to the quotient

$$g(x, v, t) = M^{-1} f(x, v, t), \quad (1.20)$$

which as we have seen has a rank-1 asymptotic limit. We will therefore search for solutions of the form

$$f = Mg = \frac{1}{(2\pi)^{d/2}} e^{-\frac{|v-E(x,t)|^2}{2}} g, \quad (1.21)$$

where g is given a low-rank approximation $\tilde{g}$:

$$g(x, v, t) \approx \tilde{g}(x, v, t) := \sum_{i,j=1}^r X_i(x, t) S_{ij}(t) V_j(v, t). \quad (1.22)$$

The bases X_i and V_j are required to satisfy orthogonality relations,

$$\langle X_i, X_k \rangle_x = \delta_{ik}, \quad \langle V_j, V_l \rangle_v = \delta_{jl}. \quad (1.23)$$

The approximation $\tilde{g}$ is the quantity which we will timestep using the dynamical low-rank method. We will also solve Ampère's law to advance the electric field, which has only a dependence on x . From these, we can reconstruct the approximate solution $\tilde{f} = \tilde{g}M$ as desired.

The rest of this paper is organized as follows. In Section 2 we present the dynamical low-rank algorithm for the evolution of the function $\tilde{g}$ defined in (1.22). This consists of deriving the PDEs satisfied by the low-rank factors. In Section 3 we present a first-order time integration scheme for that system of PDEs. In Section 4, we address the question of discretization in physical (x) and velocity (v) space. Section 5 consists of a brief discussion of the asymptotic limit of the discrete system derived in Sections 2-4, verifying that it recovers the fluid equations of (1.19). Finally, Section 6 includes some numerical results from an implementation of our algorithm and comparison with the full tensor solution of the kinetic equation.

2. Dynamical low-rank algorithm

The dynamical low-rank algorithm works by confining the time derivative of the system to the tangent space of a low-rank manifold. We will write down the time derivative of g imposed by (1.11), and then discuss its projection onto the tangent space. We can derive the dynamics for g by plugging $f = Mg$ into (1.11). This gives

$$\begin{aligned} \partial_t (Mg) + v \cdot \nabla_x (Mg) &= \frac{1}{\epsilon} \nabla_v \cdot [M \nabla_v g] \\ \implies g \partial_t M + M \partial_t g + v \cdot (g \nabla_x M + M \nabla_x g) &= \frac{1}{\epsilon} (\nabla_v M \cdot \nabla_v g) + \frac{1}{\epsilon} M \nabla_v^2 g. \end{aligned} \quad (2.1)$$

Consolidating terms and dividing through by M , we obtain

$$\begin{aligned}
\partial_t g &= -v \cdot \nabla_x g + \frac{1}{\epsilon} [(\nabla_v - v + E) \cdot \nabla_v g] - \frac{1}{M} (\partial_t M + v \cdot \nabla_x M) g \\
&= -v \cdot \nabla_x g + \frac{1}{\epsilon} [(\nabla_v - v + E) \cdot \nabla_v g] - \mathcal{M} g \\
&:= \mathcal{H}[g],
\end{aligned} \tag{2.2}$$

where we have introduced the shorthand

$$\mathcal{M} = \frac{1}{M} (\partial_t M + v \cdot \nabla_x M). \tag{2.3}$$

The time derivative of the low-rank approximation is now given by composing $\mathcal{H}$ with a projection operator. That is,

$$\partial_t \tilde{g} = P(\mathcal{H}[\tilde{g}]), \tag{2.4}$$

where P is the projection onto the tangent space to the manifold of functions with a rank r representation as in (1.22). It can be shown ([14], [17]) that the projection operator takes the form

$$P(h) = \sum_j \langle V_j, h \rangle_v V_j - \sum_{ij} X_i \langle X_i V_j, h \rangle_{xv} + \sum_i X_i \langle X_i, h \rangle_x. \tag{2.5}$$

We now have a three-term expression for the time derivative of our low-rank approximation:

$$\partial_t \tilde{g} = \sum_j \langle V_j, \mathcal{H}[\tilde{g}] \rangle_v V_j - \sum_{ij} X_i \langle X_i V_j, \mathcal{H}[\tilde{g}] \rangle_{xv} + \sum_i X_i \langle X_i, \mathcal{H}[\tilde{g}] \rangle_x. \tag{2.6}$$

This form lends itself to a first-order-in-time Lie-Trotter operator splitting, which we will employ in this paper. Higher-order splitting schemes are possible, for example a second-order scheme based on Strang splitting, although this requires extra care to properly center the electric field [8]. The first-order-in-time scheme splits (2.6) into the three equations

$$\partial_t \tilde{g} = \sum_j \langle V_j, \mathcal{H}[\tilde{g}] \rangle_v V_j, \tag{2.7}$$

$$\partial_t \tilde{g} = - \sum_{ij} X_i \langle X_i V_j, \mathcal{H}[\tilde{g}] \rangle_{xv} V_j, \tag{2.8}$$

$$\partial_t \tilde{g} = \sum_i X_i \langle X_i, \mathcal{H}[\tilde{g}] \rangle_x. \tag{2.9}$$

We actually implement this scheme in terms of a pair auxiliary bases (making use of (1.23)),

$$K_j(x, t) = \langle \tilde{g}, V_j \rangle_v = \sum_i X_i(x, t) S_{ij}(t), \tag{2.10}$$

$$L_i(v, t) = \langle X_i, \tilde{g} \rangle_x = \sum_j S_{ij}(t) V_j(v, t). \tag{2.11}$$

With this notation the splitting scheme is as follows:

- The first step holds the V_j basis constant. Take the inner product of (2.7) with V_j to obtain

$$\partial_t K_j = \langle V_j, \mathcal{H}[\tilde{g}] \rangle_v. \tag{2.12}$$

Integrate this equation for one time step to obtain a new value for K_j . Then perform a QR decomposition of K_j to obtain a new orthogonal basis X_i and coefficients S_{ij} .

- The second step holds both bases constant. Take the inner product of (2.8) with V_j in v , and with X_i in x to obtain

$$\partial_t S_{ij} = - \langle X_i V_j, \mathcal{H}[\tilde{g}] \rangle_{xv}. \tag{2.13}$$

Integrate this equation for one time step to obtain a new matrix S_{ij} .

- The third step holds the X_i basis constant. Take the inner product of (2.9) with X_i to obtain

$$\partial_t L_i = \langle X_i, \mathcal{H}[\tilde{g}] \rangle_x. \tag{2.14}$$

Integrate this equation for one time step to obtain a new value for L_i . Then perform a QR decomposition of L_i to obtain a new orthogonal basis V_j and coefficient matrix S_{ij} .

The above algorithm has the excellent property that it is robust to “overapproximation”, i.e. small singular values in S [17].

The time splitting scheme for $\tilde{g}$ may be straightforwardly coupled with Ampère’s equation (1.4), which can be written in terms of the low-rank components as

$$\partial_t E = -\langle v M \tilde{g} \rangle_v = -\sum_{i,j} X_i S_{ij} \langle v M V_j \rangle_v. \quad (2.15)$$

Remark 2. In this section we have applied the projector-splitting integrator of [17]. Another choice is the “unconventional” integrator proposed in [3]. The unconventional integrator has the advantage of not taking a “backwards” step (viz. the negative sign on the right hand side of (2.13)). On the other hand, it is not known how to extend the unconventional integrator to higher than first order accuracy in time. Fully exploring the tradeoffs between the two integrators in the setting of kinetic equations is a topic for future work.

2.1. Time evolution of low-rank components

In this section we expand the inner products involving $\mathcal{H}[\tilde{g}]$ which appear in equations (2.12), (2.13), (2.14). The result is a self-contained system of r coupled PDEs for K_j and L_i , and a matrix-valued ODE for S_{ij} of size $r \times r$.

Plugging (2.2) and (1.22) into (2.12) gives

$$\begin{aligned} \partial_t K_j &= -\sum_{k,l} \langle V_j, v \cdot (\nabla_x X_k) S_{kl} V_l \rangle_v - \sum_{kl} X_k S_{kl} \langle V_j, V_l \mathcal{M} \rangle_v \\ &\quad + \frac{1}{\epsilon} \left(\sum_{kl} X_k S_{kl} \langle V_j [(\nabla_v - v + E) \cdot \nabla_v V_l] \rangle_v \right) \\ &= -\sum_l (\nabla_x K_l) \cdot \langle v V_j V_l \rangle_v - \sum_l K_l \langle V_j V_l \mathcal{M} \rangle_v \\ &\quad + \frac{1}{\epsilon} \left(\sum_l K_l [\langle V_j (\nabla_v - v) \cdot \nabla_v V_l \rangle_v + E \cdot \langle V_j \nabla_v V_l \rangle_v] \right). \end{aligned} \quad (2.16)$$

Plugging (2.2) and (1.22) into (2.13) gives

$$\begin{aligned} \partial_t S_{ij} &= \sum_{kl} \langle X_i S_{kl} \nabla_x X_k \cdot \langle v V_j V_l \rangle_v \rangle_x + \sum_{kl} \langle X_i X_k S_{kl} V_l V_j \mathcal{M} \rangle_{xv} \\ &\quad - \frac{1}{\epsilon} \left(\sum_{kl} \langle X_i S_{kl} X_k \langle V_j [(\nabla_v - v + E) \cdot \nabla_v V_l] \rangle_v \rangle_x \right) \\ &= \sum_{kl} S_{kl} \langle X_i \nabla_x X_k \rangle_x \cdot \langle v V_j V_l \rangle_v + \sum_{kl} S_{kl} \langle X_i X_k V_l V_j \mathcal{M} \rangle_{xv} \\ &\quad - \frac{1}{\epsilon} \left(\sum_{kl} S_{kl} [\langle X_i X_k \rangle_x \langle V_j (\nabla_v - v) \cdot \nabla_v V_l \rangle_v + \langle X_i X_k E \rangle_x \cdot \langle V_j \nabla_v V_l \rangle_v] \right). \end{aligned} \quad (2.17)$$

Plugging (2.2) and (1.22) into (2.14) gives

$$\begin{aligned} \partial_t L_i &= -\sum_{kl} v \cdot \langle X_i (\nabla_x X_k) S_{kl} V_l \rangle_x - \sum_{kl} \langle X_i X_k S_{kl} V_l \mathcal{M} \rangle_x \\ &\quad + \frac{1}{\epsilon} \left(\sum_{kl} \langle X_i X_k S_{kl} [(\nabla_v - v + E) \cdot \nabla_v V_l] \rangle_x \right) \\ &= -\sum_k v \cdot \langle X_i (\nabla_x X_k) \rangle_x L_k - \sum_k \langle X_i X_k \mathcal{M} \rangle_x L_k \\ &\quad + \frac{1}{\epsilon} \left(\sum_k [\langle X_i X_k \rangle_x (\nabla_v - v) + \langle X_i X_k E \rangle_x] \cdot \nabla_v L_k \right). \end{aligned} \quad (2.18)$$

We also expand the terms involving $\mathcal{M}$ (defined in (2.3)):

$$\partial_t M = (v - E) \cdot (\partial_t E) M = -M(v - E) \cdot J,$$

$$\mathbf{v} \cdot \nabla_x \mathcal{M} = M \sum_{i,j} (\mathbf{v}_j - E_j) \mathbf{v}_i \partial_{x_i} E_j = M \sum_{i,j} (\mathbf{v}_j \mathbf{v}_i \partial_{x_i} E_j - E_j \mathbf{v}_i \partial_{x_i} E_j) = M(\mathbf{v} \otimes \mathbf{v}) : \nabla_x E - \frac{M}{2} \mathbf{v} \cdot \nabla_x (E^2).$$

Note that $\frac{1}{2} \nabla_x (E^2) \neq E \cdot \nabla_x E$. $(\nabla_x E)_{ij} := \partial_{x_j} E_i$ and $A : B := \sum_{ij} a_{ij} b_{ij}$. Putting these together we obtain

$$\begin{aligned} \mathcal{M} &= \frac{1}{M} (\partial_t M + \mathbf{v} \cdot \nabla_x M) = E \cdot \mathbf{J} - \mathbf{v} \cdot \mathbf{J} - \frac{1}{2} \mathbf{v} \cdot \nabla_x (E^2) + (\mathbf{v} \otimes \mathbf{v}) : \nabla_x E \\ &:= \mathcal{M}_1 + \mathbf{v} \cdot \mathbf{M}_2 + (\mathbf{v} \otimes \mathbf{v}) : \overline{\mathbf{M}}_3, \end{aligned} \quad (2.19)$$

where

$$\mathcal{M}_1 = E \cdot \mathbf{J}, \quad \mathbf{M}_2 = -\mathbf{J} - \frac{1}{2} \nabla_x (E^2), \quad \overline{\mathbf{M}}_3 = \nabla_x E. \quad (2.20)$$

Here we use **boldface** to denote vectors of length d , and sans-serif to denote tensors of size $d \times d$. Both vectors and tensors may also vary in x and v . In all cases the tensor contractions $\cdot, :$ indicate contraction over the length- d dimensions. The terms involving $\mathcal{M}$ then expand to

$$\begin{aligned} \langle V_j V_l \mathcal{M} \rangle_v &= \delta_{jl} \mathcal{M}_1 + \langle v V_j V_l \rangle_v \cdot \mathbf{M}_2 + \langle (\mathbf{v} \otimes \mathbf{v}) V_j V_l \rangle_v : \overline{\mathbf{M}}_3, \\ \langle X_i X_k \mathcal{M} \rangle_x &= \langle X_i X_k \mathcal{M}_1 \rangle_x + \mathbf{v} \cdot \langle X_i X_k \mathbf{M}_2 \rangle_x + (\mathbf{v} \otimes \mathbf{v}) : \langle X_i X_k \overline{\mathbf{M}}_3 \rangle_x, \\ \langle X_i X_k V_j V_l \mathcal{M} \rangle_{xv} &= \delta_{jl} \langle X_i X_k \mathcal{M}_1 \rangle_x + \langle v V_j V_l \rangle_v \cdot \langle X_i X_k \mathbf{M}_2 \rangle_x + \langle (\mathbf{v} \otimes \mathbf{v}) V_j V_l \rangle_v : \langle X_i X_k \overline{\mathbf{M}}_3 \rangle_x. \end{aligned}$$

3. First order in time scheme

The algorithm described up to this point has been fully continuous, except for the projection onto the low-rank manifold. We now present a discretization in time, leaving space continuous for the moment. The time discretization makes use of an implicit-explicit (IMEX) scheme for capturing the fast dynamics of the collision operator in the fluid limit ($\epsilon \ll 1$).

In the following we report rough estimates of the computational complexity of each substep. To avoid complicating the presentation unnecessarily, for these estimates we consider $d \sim 1$, so that we are free to ignore both the dimension and constant factors in our “big-O” notation.

Suppose we have the quantities $(E^n, X_i^n, V_j^n, S_{ij}^n)$ at timestep t^n . Then we calculate $(E^{n+1}, X_i^{n+1}, V_j^{n+1}, S_{ij}^{n+1})$ in the following way.

3.1. Step 1: update E

1. Compute the following integral appearing in (2.15):

$$I_j^n(x) := \langle v V_j^n M^n \rangle_v = \frac{1}{(2\pi)^{d/2}} \int v V_j^n(v) e^{-\frac{|v-E^n(x)|^2}{2}} dv. \quad (3.1)$$

A naive computation of this integral requires $\mathcal{O}(N_x N_v)$ steps, a computational cost that is unacceptably high. However, since the Maxwellian is isothermal, the integral has a convolutional structure, and may be computed using a Fast Fourier Transform (FFT). The required substeps are:

- Compute the convolution

$$\ell_j^n(\zeta) = \left[(v \mapsto v V_j^n(v)) * (v \mapsto e^{-|v|^2/2}) \right](\zeta) \quad (3.2)$$

using an FFT.

Cost: $\mathcal{O}(r N_v \log N_v)$.

- Compute the composition of E^n with ℓ_j^n using any interpolation scheme from the FFT nodes to an arbitrary point $E^n(x)$:

$$I_j^n(x) = \frac{1}{(2\pi)^{d/2}} \ell_j^n(E^n(x)).$$

Cost: $\mathcal{O}(r N_x)$.

Exploiting the convolutional structure of $I_j^n(x)$ with an FFT reduces the total computational cost to $\mathcal{O}(r N_v \log N_v)$, which is acceptable.

2. Compute the current density:

$$J^n(x) = \sum_{ij} X_i^n(x) S_{ij}^n I_j^n(x). \quad (3.3)$$

Cost: $\mathcal{O}(r^2 N_x)$.

3. Perform a Forward Euler step to solve (2.15):

$$E^{n+1}(x) = E^n(x) - \Delta t J^n. \quad (3.4)$$

Cost: $\mathcal{O}(N_x)$.

Remark 3 (Solving Gauss's law). If one chooses to solve the Vlasov-Poisson system instead of the Vlasov-Ampère system considered here, the E update requires the solution of an elliptic equation for the potential ϕ :

$$-\nabla_x^2 \phi^n = \rho^n - \eta, \quad E^{n+1} = -\nabla_x \phi^n. \quad (3.5)$$

The only aspect requiring special treatment in the low-rank method is the computation of the charge density ρ^n appearing on the right-hand side of Gauss's law. This can be computed in much the same manner as the current density J^n in equations (3.1)-(3.3). Briefly, compute r integrals

$$Y_j^n(x) := \left\langle V_j^n M^n \right\rangle_v = \frac{1}{(2\pi)^{d/2}} \int V_j^n(v) e^{-\frac{|v-E^n|^2}{2}} dv. \quad (3.6)$$

This may be done by computing convolutions

$$y_j^n(\zeta) = \left[\left(v \mapsto V_j^n(v) \right) \star \left(v \mapsto e^{-|v|^2/2} \right) \right](\zeta), \quad (3.7)$$

and interpolating in any desired manner to the arbitrary points $E^n(x)$:

$$Y_j^n(x) = \frac{1}{(2\pi)^{d/2}} y_j^n(E^n(x)). \quad (3.8)$$

Then the charge density is given by

$$\rho^n(x) = \sum_{ij} X_i^n(x) S_{ij}^n Y_j^n(x). \quad (3.9)$$

It will be seen that this procedure is nearly identical to that for computing J^n , with the only difference that one computes the zeroth moment of f rather than the first moment. Note that in this formulation, one still needs to calculate J^n via the corresponding convolutions, since it will be used to update K , S , and L as described below.

Remark 4. The use of the FFT to achieve a fast convolutional algorithm is similar to that proposed in [10]. Computationally the approach is nearly identical, with $E^n(x)$ taking the place of $u^n(x)$ as the local drift velocity of the Maxwellian.

3.2. Step 2: update X , S , and V

3.2.1. K step

1. Compute integrals in v . We use **boldface** to denote vector-valued matrices of total size $r \times r \times d$, and **sans-serif** to denote tensor-valued matrices of total size $r \times r \times d \times d$. In both cases the indices running over the length- d dimensions are suppressed. The integrals to compute are:

$$\mathbf{c}_{jl}^1 = \langle v V_j^n V_l^n \rangle_v, \quad \overline{\mathbf{c}}_{jl}^2 = \langle (v \otimes v) V_j^n V_l^n \rangle_v, \quad (3.10)$$

$$d_{jl}^1 = \langle V_j^n (\nabla_v - v) \cdot \nabla_v V_l^n \rangle_v, \quad \mathbf{d}_{jl}^2 = \langle V_j^n \nabla_v V_l^n \rangle_v. \quad (3.11)$$

Cost: $\mathcal{O}(r^2 N_v)$.

2. Compute $\mathcal{M}_1^n$, $\mathbf{M}_2^n$, and $\overline{\mathbf{M}}_3^n$:

$$\mathcal{M}_1^n = E^n \cdot J^n, \quad \mathbf{M}_2^n = -J^n - \frac{1}{2} \nabla_x ((E^n)^2), \quad (3.12)$$

$$\overline{\mathbf{M}}_3^n = \nabla_x E^n. \quad (3.13)$$

Cost: $\mathcal{O}(N_x)$.

3. Compute matrices on the right hand side of (2.16).

$$A_{jl}^1 = \delta_{jl} \mathcal{M}_1^n + \mathbf{c}_{jl}^1 \cdot \mathbf{M}_2^n + \overline{\mathbf{c}}_{jl}^2 : \overline{\mathbf{M}}_3^n, \quad (3.14)$$

$$A_{jl}^2 = d_{jl}^1 + E^n \cdot \mathbf{d}_{jl}^2. \quad (3.15)$$

Cost: $\mathcal{O}(r^2 N_x)$.

4. The evolution equation (2.16) for K may now be written as

$$\partial_t K_j = - \sum_l \mathbf{c}_{jl}^1 \cdot \nabla_x K_l - \sum_l A_{jl}^1 K_l + \frac{1}{\epsilon} \sum_l A_{jl}^2 K_l. \quad (3.16)$$

Advance (3.16) in time, using an IMEX step to handle the stiff term:

$$\sum_l \left[\delta_{jl} - \frac{\Delta t}{\epsilon} A_{jl}^2 \right] K_l^{n+1} = K_j^n - \Delta t \left(\sum_l \mathbf{c}_{jl}^1 \cdot \nabla_x K_l^n + \sum_l A_{jl}^1 K_l^n \right). \quad (3.17)$$

Note that the only differential operator, namely ∇_x , appearing in this equation is treated explicitly. Therefore the linear system appearing in this equation involves no coupling between points in x . When discretized it will consist of N_x separate systems each of size $r \times r$. We can solve this small system at each point in x using any standard dense linear solver—the size is not large enough to warrant any special technique.

Cost: $\mathcal{O}(r^2 N_x)$ for both the right-hand side and the implicit step, due to solving each $r \times r$ system separately.

5. Perform a QR decomposition of K_j^{n+1} to obtain X_i^{n+1} and S_{ij}^1 .

Cost: $\mathcal{O}(r^2 N_x)$.

3.2.2. S step

1. Compute the integrals in x , using the new basis X^{n+1} :

$$\mathbf{c}_{ik}^* = \langle X_i^{n+1} X_k^{n+1} \mathcal{M}_1^n \rangle_x, \quad \mathbf{c}_{ik}^{**} = \langle X_i^{n+1} X_k^{n+1} \mathcal{M}_2^n \rangle_x, \quad \bar{\mathbf{c}}_{ik}^{***} = \langle X_i^{n+1} X_k^{n+1} \bar{\mathcal{M}}_3^n \rangle_x, \quad (3.18)$$

$$\mathbf{d}_{ik}^* = \langle X_i^{n+1} \nabla_x X_k^{n+1} \rangle_x, \quad \mathbf{e}_{ik}^* = \langle X_i^{n+1} X_k^{n+1} E^n \rangle_x. \quad (3.19)$$

Cost: $\mathcal{O}(r^2 N_x)$.

2. Compute

$$\tilde{c}_{ij;kl} = \langle X_i^{n+1} X_k^{n+1} V_j^n V_l^n \mathcal{M}^n \rangle_{xv} = \delta_{jl} \mathbf{c}_{ik}^* + \mathbf{c}_{jl}^1 \cdot \mathbf{c}_{ik}^{**} + \bar{\mathbf{c}}_{jl}^2 : \bar{\mathbf{c}}_{ik}^{***}. \quad (3.20)$$

Cost: $\mathcal{O}(r^4)$.

3. Compute the order-four tensors

$$B_{ij;kl}^1 = \mathbf{d}_{ik}^* \cdot \mathbf{c}_{jl}^1 + \tilde{c}_{ij;kl}, \quad (3.21)$$

$$B_{ij;kl}^2 = \delta_{ik} \mathbf{d}_{jl}^1 + \mathbf{e}_{ik}^* \cdot \mathbf{d}_{jl}^2. \quad (3.22)$$

Cost: $\mathcal{O}(r^4)$.

4. The evolution equation for S may now be written as

$$\partial_t S_{ij} = \sum_{kl} B_{ij;kl}^1 S_{kl} - \frac{1}{\epsilon} \sum_{kl} B_{ij;kl}^2 S_{kl}. \quad (3.23)$$

Perform a Forward Euler step to advance $S_{ij}^1 \rightarrow S_{ij}^2$:

$$S_{ij}^2 = S_{ij}^1 + \Delta t \sum_{kl} S_{kl}^1 B_{ij;kl}^1 - \frac{\Delta t}{\epsilon} \sum_{kl} S_{kl}^1 B_{ij;kl}^2. \quad (3.24)$$

Our use of a Forward Euler step here differs from the presentation in [10], where an IMEX step was used to advance the S equation in the case of the BGK collision operator. Since the structure of the Fokker-Planck operator is more complicated than the BGK type, special care is needed. Heuristically, one can see that something different from the K and L steps may be required, simply because the S equation runs backwards in time. For a more detailed justification and a discussion of how the situation differs from [10], refer to Appendix A.

Cost: $\mathcal{O}(r^4)$.

3.2.3. L step

1. Compute the $r \times r$ matrix

$$\hat{c}_{ik} = \mathbf{c}_{ik}^* + v \cdot \mathbf{c}_{ik}^{**} + (v \otimes v) : \bar{\mathbf{c}}_{ik}^{***}. \quad (3.25)$$

Cost: $\mathcal{O}(r^2)$.

2. The equation (2.18) may now be written as

$$\partial_t L_i = - \sum_k v \cdot \mathbf{d}_{ik}^* L_k - \sum_k \hat{c}_{ik} L_k + \frac{1}{\epsilon} \sum_k \left(\delta_{ik} \nabla_v^2 - \delta_{ik} v \cdot \nabla_v + \mathbf{e}_{ik}^* \cdot \nabla_v \right) L_k. \quad (3.26)$$

Advance (3.26) using an IMEX step by solving the system

$$\sum_k \left[\delta_{ik} - \frac{\Delta t}{\epsilon} (\delta_{ik} \nabla_v^2 - \delta_{ik} v \cdot \nabla_v + \mathbf{e}_{ik}^* \cdot \nabla_v) \right] L_k^{n+1} = L_i^n - \Delta t \sum_k (v \cdot \mathbf{d}_{ik}^* + \hat{c}_{ik}) L_k^n. \quad (3.27)$$

Note that in contrast to (3.17), the left-hand side of this equation does involve differential operators in v , and so the linear system may be discretized by a fully coupled (but sparse) matrix of size $rN_v \times rN_v$. There are $\mathcal{O}(rN_v)$ non-empty entries. Using Krylov subspace methods lets us keep the total cost of solving this system on the order of $\mathcal{O}(rN_v)$, assuming the number of iterations does not grow unboundedly with r or N_v , which is what we observe in practice.

Cost: $\mathcal{O}(r^2 N_v)$.

3. Perform a QR decomposition of L_i^{n+1} to obtain V_j^{n+1} and S_{ij}^{n+1} .

Cost: $\mathcal{O}(r^2 N_v)$.

Adding together all of our computational complexity estimates, we get a total cost of $\mathcal{O}(r^4 + r^2 N_x + r^2 N_v)$ – compare this with the cost of the full tensor method $\mathcal{O}(N_x N_v)$.

4. Fully discrete algorithm

In this section we address the question of physical and velocity space discretization. One of the virtues of the dynamical low-rank method is that it decouples the discretization of the two bases, X_i and V_j , which may be treated more or less independently. The X basis is updated by solving a system of coupled hyperbolic PDEs in (3.17), while the V basis is updated by solving a parabolic system in (3.27). These systems are coupled via the matrix of singular values S , as well as weighted inner products of whichever basis is being held constant (viz. (3.10), (3.11), etc.). We are free to choose whichever discretization is most appropriate for the corresponding evolution equation of each basis. In this work we use second-order finite difference discretizations in both x and v for simplicity. In principle, it is easy to choose, for example, a Fourier spectral method to take advantage of periodicity in the x direction, or even a more involved method such as Discontinuous Galerkin along one or the other basis, without increasing the implementation complexity too greatly.

4.1. Spatial discretization

Our spatial discretization in x is designed to solve the explicit part of the evolution equation for K , which is (3.17). This is a linear hyperbolic PDE with the flux matrix $\mathbf{c}_{jl}^1$, which is a symmetric matrix. We opt for a second-order finite difference discretization with flux limiting, as described in [16], section 16.2. To illustrate, we consider the situation in two spatial dimensions, $d = 2$. The matrix $\mathbf{c}_{jl}^1$ consists of components $\mathbf{c}_{jl}^{1;m}$ for $m \in \{1, 2\}$, acting on the x and y directions respectively. The matrices $\mathbf{c}_{jl}^{1;m}$ are symmetric and real; recall their definition (3.10). Therefore they are unitarily diagonalizable, and we can write

$$(T^m)^T \mathbf{c}_{jl}^{1;m} T^m = \sum_{kl} T_{ij}^m \mathbf{c}_{kl}^{1;m} T_{kl}^m = \lambda_i^m \delta_{ik} = \Lambda^m. \quad (4.1)$$

Left-multiplying (3.17) by $(T^1)^T$, and introducing the eigenbasis $\hat{K}_i^n = [(T^1)^T K^n]_i$, gives the system

$$\sum_{jl} T_{ij}^1 \left[I - \frac{\Delta t}{\epsilon} \mathbf{A}_{jl}^2 \right] K_l^{n+1} = \hat{K}_i^n - \Delta t \lambda_i^1 \partial_x \hat{K}_i^n - \Delta t \sum_{jl} T_{ij}^1 \left(\mathbf{c}_{jl}^{1;2} \partial_y K_l^n + \mathbf{A}_{jl}^1 K_l^n \right). \quad (4.2)$$

At a grid point x_p , the flux-limited finite difference discretization approximates $\lambda_i^1 \partial_x \hat{K}_i^n(x_p)$ by a difference of fluxes at half grid points $x_{p+1/2}, x_{p-1/2}$:

$$\lambda_i^1 \partial_x \hat{K}_i^n(x_p) \approx \frac{F(\lambda_i, \hat{K}_i^n)_{p+1/2} - F(\lambda_i, \hat{K}_i^n)_{p-1/2}}{\Delta x}. \quad (4.3)$$

The flux F is given by the combination of a first-order flux (upwinding) flux,

$$F_L(\lambda, \hat{K})_{p+1/2} = \frac{\lambda}{2} (\hat{K}_{p+1} + \hat{K}_p) - \frac{|\lambda|}{2} (\hat{K}_{p+1} - \hat{K}_p), \quad (4.4)$$

with a second-order Lax-Wendroff flux. The combination is governed by a flux-limiter $\phi(\theta)$, which stabilizes the scheme in the presence of sharp changes in the gradient:

$$F(\lambda, \hat{K})_{p+1/2} = F_L(\lambda, \hat{K})_{p+1/2} + \frac{1}{2}\phi(\theta_{p+1/2}) \left(\text{sgn}(\lambda) - \frac{\lambda \Delta t}{\Delta x} \right) \lambda \delta(\hat{K})_{p+1/2}, \quad (4.5)$$

where $\delta(\hat{K})_{p+1/2} = (\hat{K}_{p+1} - \hat{K}_p)$. The quantity $\theta_{p+1/2}$ measures how quickly the gradient is changing in the vicinity of $x_{p+1/2}$, and itself uses upwinding based on the sign of λ :

$$\theta_{p+1/2} = \frac{\delta(\hat{K})_{p+1/2 - \text{sgn}(\lambda)}}{\delta(\hat{K})_{p+1/2}}. \quad (4.6)$$

The function $\phi: \mathbb{R} \rightarrow [0, 2]$ is called the limiter, and there are many options to choose from. We use the Van Leer limiter,

$$\phi(\theta) = \frac{|\theta| + \theta}{1 + |\theta|}. \quad (4.7)$$

After approximating the term $\lambda_i^1 \partial_x \hat{K}_i^n$, for each eigenvalue λ_i , we can transform back to the original variables by left-multiplying with (T^1) :

$$\sum_l \left[I - \frac{\Delta t}{\epsilon} \mathbf{A}_{jl}^2 \right] K_l^{n+1} = K_j^n - \Delta t \sum_i T_{ij}^1 \delta_x(\lambda_i, \hat{K}^n)_i - \Delta t \sum_l \left(\mathbf{c}_{jl}^{1;2} \partial_y K_l^n + \mathbf{A}_{jl}^1 K_l^n \right). \quad (4.8)$$

The discretization in y is handled similarly, by left-multiplying (4.8) by T^2 . The above scheme is second-order in smooth regions of the solution, and degrades to first order around discontinuities and extrema.

4.2. Velocity discretization

Our discretization in v is designed to effectively solve (3.27), which is a parabolic system (strictly speaking, a convection-diffusion type equation). We recall the linear system to be solved here:

$$\sum_k \left[\delta_{ik} - \frac{\Delta t}{\epsilon} (\delta_{ik} \nabla_v^2 - \delta_{ik} v \cdot \nabla_v + \mathbf{e}_{ik}^* \cdot \nabla_v) \right] L_k = RHS. \quad (4.9)$$

It is convenient to discretize this operator by splitting the left hand side into a diagonal ($i = k$) term and an offdiagonal term. In the case when $i = k$, we have

$$\begin{aligned} \left(\delta_{ik} \nabla_v^2 - \delta_{ik} v \cdot \nabla_v + \mathbf{e}_{kk}^* \cdot \nabla_v \right) L_k &= \left(\nabla_v^2 - v \cdot \nabla_v + \mathbf{e}_{kk}^* \cdot \nabla_v \right) L_k \\ &= \left[((\nabla_v - v) + \mathbf{e}_{kk}^*) \cdot \nabla_v \right] L_k \\ &= \frac{1}{M^k} \nabla_v \cdot (M^k \nabla_v L_k) \\ &:= \mathbb{T}_k(\mathbf{e}_{kk}^*) L_k, \end{aligned} \quad (4.10)$$

where M^k is the local Maxwellian

$$M^k = e^{-\frac{|v - \mathbf{e}_{kk}^*|^2}{2}}. \quad (4.11)$$

In one dimension, a second-order-accurate central difference discretization of (4.10) is

$$(\mathbb{T}_k(\mathbf{e}_{kk}^*) L)_p \approx \frac{1}{M_p^k \Delta v} \left(M_{p+1/2}^k \frac{L_{p+1} - L_p}{\Delta v} - M_{p-1/2}^k \frac{L_{p-1} - L_p}{\Delta v} \right) \quad (4.12)$$

$$= \frac{M_{p+1/2}^k L_{p+1} - (M_{p+1/2}^k + M_{p-1/2}^k) L_p + M_{p-1/2}^k L_{p-1}}{M_p^k \Delta v^2}. \quad (4.13)$$

The off-diagonal terms are simply

$$\mathbf{e}_{ik}^* \cdot \nabla_v L_k := \mathbb{U}_{ik} L_k. \quad (4.14)$$

For simplicity we discretize this using a second-order centered difference operator. Stability is not a concern, since it will be coupled to a Backwards Euler timestepping scheme. With these discretizations in hand the implicit step for L takes the form

$$\sum_k \left[\delta_{ik} - \frac{\Delta t}{\epsilon} (\delta_{ik} \mathbb{T}_k(\mathbf{e}_{kk}^*) + \mathbb{U}_{ik}) \right] L_k^{n+1} = L_i^n - \Delta t \sum_k (v \cdot \mathbf{d}_{ik}^* + \hat{c}_{ik}) L_k^n. \quad (4.15)$$

Despite being of size $rN_v \times rN_v$, this linear system is quite sparse, having roughly $\mathcal{O}(r^2 N_v)$ nonzero entries. It is therefore amenable to fast solution by iterative solvers. Since it is not symmetric, we use the Restarted GMRES [21] iterative algorithm. We find good results by preconditioning with the constant matrix $(\delta_{ik} - \delta_{ik} \frac{\Delta t}{\epsilon} \mathbb{T}_k(\mathbf{0}))^{-1}$. Timings of our code indicate that this step takes on the same order of magnitude as the other components of the algorithm, up to the largest problems we consider here.

5. Asymptotic behavior of the discrete scheme

In this section we demonstrate that the discrete scheme described in the preceding sections preserves the asymptotic limit (1.19) as $\epsilon \rightarrow 0$.

We consider the limit of the discrete system at the level of g , which is advanced via the K , S , and L steps with an accuracy that is first-order in time and second-order in space:

$$\frac{g^{n+1} - g^n}{\Delta t} = -v \cdot \nabla_x g^n - \mathcal{M}^n g^n + \frac{1}{\epsilon} (M^n)^{-1} \nabla_v \cdot (M^n \nabla_v g^{n+1}) + \mathcal{O}(\Delta t + \Delta x^2). \quad (5.1)$$

Furthermore, the electric field is advanced by

$$\frac{E^{n+1} - E^n}{\Delta t} = -J^n. \quad (5.2)$$

From (5.1), we can see that

$$(M^n)^{-1} \nabla_v \cdot (M^n \nabla_v g^{n+1}) = \mathcal{O}(\epsilon) \implies g^{n+1} = c + \mathcal{O}(\epsilon), \quad (5.3)$$

that is to say, after one time step, we expect the solution g to be close to a constant function in v when ϵ is small. To see this, just note the following

$$\int \nabla_v \cdot [M \nabla_v g] \log g \, dv = \int \nabla_v \cdot [M g \nabla_v \log g] \log g \, dv = - \int M g |\nabla_v \log g|^2 \, dv \leq 0, \quad (5.4)$$

where the equality holds if and only if g is a function independent of v . Moreover, (5.3) implies

$$\rho^{n+1} = \int M^{n+1} g^{n+1} \, dv = c \int M^{n+1} \, dv + \mathcal{O}(\epsilon) = c + \mathcal{O}(\epsilon), \quad (5.5)$$

$$J^{n+1} = \int v M^{n+1} g^{n+1} \, dv = c \int v M^{n+1} \, dv + \mathcal{O}(\epsilon) = c E^{n+1} + \mathcal{O}(\epsilon) = \rho^{n+1} E^{n+1} + \mathcal{O}(\epsilon). \quad (5.6)$$

On the other hand, we can multiply (5.1) by M^n and integrate in v to obtain

$$\begin{aligned} \frac{\int M^n g^{n+1} \, dv - \rho^n}{\Delta t} &= - \int v \cdot (\nabla_x g^n) M^n \, dv - \int (\partial_t M^n + v \cdot \nabla_x M^n) g^n \, dv \\ &\quad + \frac{1}{\epsilon} \int \nabla_v \cdot (M^n \nabla_v g^{n+1}) \, dv + \mathcal{O}(\Delta t + \Delta x^2) \\ &= - \int [v \cdot (\nabla_x g^n) M^n + (v \cdot \nabla_x M^n) g^n] \, dv - \partial_t E^n \cdot \int (v - E^n) M^n g^n \, dv \\ &\quad + \mathcal{O}(\Delta t + \Delta x^2) \\ &= - \nabla_x \cdot J^n - c \partial_t E^n \cdot \int (v - E^n) M^n \, dv + \mathcal{O}(\epsilon + \Delta t + \Delta x^2), \end{aligned} \quad (5.7)$$

where we used $g^n = c + \mathcal{O}(\epsilon)$ for $n \geq 1$.

Finally, noticing that

$$\rho^{n+1} - \int M^n g^{n+1} \, dv = \int (M^{n+1} - M^n) g^{n+1} \, dv = c \int (M^{n+1} - M^n) \, dv + \mathcal{O}(\epsilon) = \mathcal{O}(\epsilon), \quad (5.8)$$

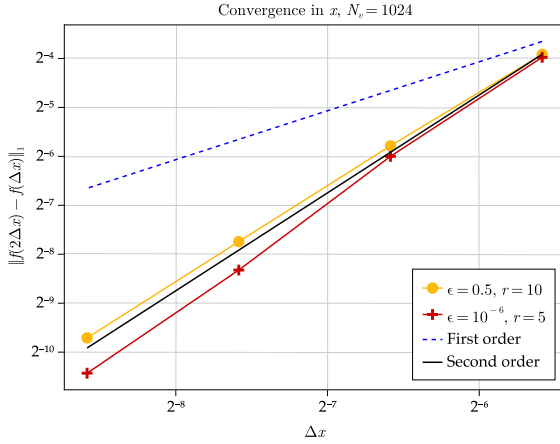
and using $J^n = \rho^n E^n + \mathcal{O}(\epsilon)$ for $n \geq 1$ (5.7) becomes

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = - \nabla_x \cdot (\rho^n E^n) + \mathcal{O}\left(\epsilon + \Delta t + \Delta x^2 + \frac{\epsilon}{\Delta t}\right). \quad (5.9)$$

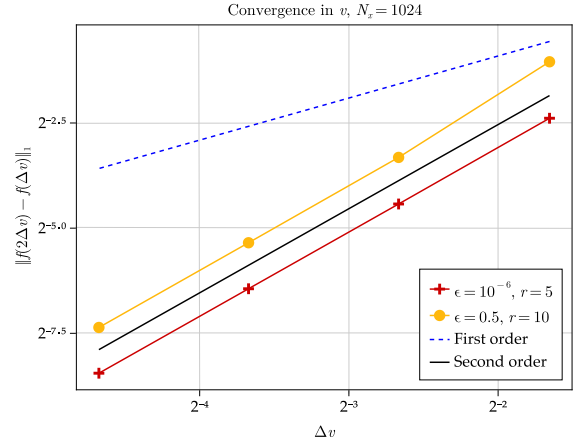
(5.2) becomes

$$\frac{E^{n+1} - E^n}{\Delta t} = - \rho^n E^n + \mathcal{O}(\epsilon). \quad (5.10)$$

Equations (5.9) and (5.10) form a first-order in time discretization of (1.19) as $\epsilon \rightarrow 0$, as expected.



(a) Second order convergence in x . $N_v = 1024$, N_x between 24 and 384.



(b) Second order convergence in v . $N_x = 1024$, N_v between 32 and 512.

Fig. 1. Convergence of the solution as a function of the grid spacing in both x and v . We achieve second-order convergence in v , and very nearly second-order convergence in x , as expected.

6. Numerical results

In this section, we present extensive numerical results in 1D1V and 2D2V to illustrate the accuracy and efficiency of the proposed low-rank algorithm. We will see that the algorithm becomes quite low rank in the asymptotic limit $\epsilon \rightarrow 0$ by our design. On the other hand, in the kinetic and transition regimes, the numerical rank needed appears higher but still relatively small compared to N_x or N_v . Therefore, the proposed algorithm presents as a very effective method for the Vlasov-Ampère-Fokker-Planck system over a wide range of problems.

6.1. 1D1V examples

6.1.1. Convergence study

To verify second order convergence of our scheme in physical space and velocity space, we perform convergence studies comparing the relative errors in f as the grid is refined. In order to evaluate the performance of the discrete Fokker-Planck collision operator, we use a nonequilibrium initial condition. The initial distribution consists of two counterstreaming beams moving at velocities ± 1.5 .

$$f(x, v, 0) = \frac{\rho_0(x)}{2\sqrt{2\pi}} \left[e^{-\frac{|v-1.5|^2}{2}} + e^{-\frac{|v+1.5|^2}{2}} \right], \quad (6.1)$$

$$\rho_0(x) = \sqrt{2\pi} (2 + \cos(2\pi x)). \quad (6.2)$$

The initial electric field is determined by Poisson's equation,

$$E_0 = -\nabla_x \phi(x), \quad -\nabla_x^2 \phi(x) = \rho_0(x) - \eta(x), \quad (6.3)$$

$$\eta(x) = \frac{2\sqrt{2\pi}}{1.2661} e^{\cos(2\pi x)}. \quad (6.4)$$

We evaluate the convergence in both the kinetic and fluid regimes, with $\epsilon = 0.5$ and $\epsilon = 10^{-6}$, respectively. The fluid regime is adequately resolved with $r = 5$, while the kinetic regime requires a higher rank of $r = 10$. The spatial domain is periodic on the interval $[0, 1]$, and the velocity domain is the interval $[-10, 10]$. Convergence is verified by holding one of N_x, N_v fixed, while the other is varied. The timestep Δt is chosen to give a CFL number of 0.25 at the finest grid, for which we use $\Delta t = \frac{1.0}{(1024)(4v_{\max})} = 2.44 \times 10^{-5}$. This is found to be sufficient for the spatial discretization error to dominate. The L^1 norm of the difference between f at subsequent levels of approximation is computed by exactly restricting the solution at the finer grid onto the coarser grid. Our chosen discretization in the x direction is a high-resolution flux-limited scheme as described in Section 4.1, for which we expect between first and second-order convergence depending on the smoothness of the solution. Indeed we observe very nearly second-order convergence on this particular problem (Fig. 1a). For convergence in the v direction, we find that the method is second-order in the grid spacing Δv as expected (Fig. 1b).

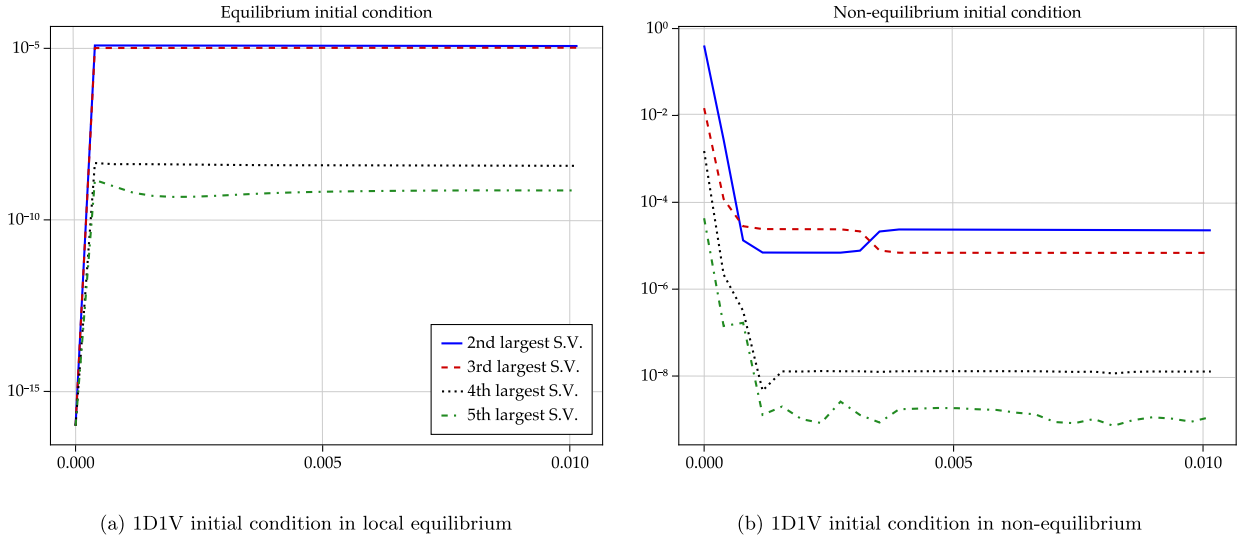


Fig. 2. Ratio of the second- through fifth-largest singular values to the largest singular value in the fluid regime, $\epsilon = 10^{-6}$. There is a clear separation of five orders of magnitude between the largest and second largest singular values. This holds whether the solution begins in local equilibrium (left) or in the non-equilibrium initial condition (6.1).

6.1.2. Verification that the asymptotic limit is rank-1

In order to verify that our method efficiently captures the asymptotic limit (1.19), we examine the evolution of the singular values of the low rank solution for very small ϵ ($\epsilon = 10^{-6}$). The singular values of the low rank solution are simply the diagonal entries of S . We consider a solution beginning in local equilibrium,

$$f(x, v, 0) = \frac{\rho_0(x)}{\sqrt{2\pi}} e^{-\frac{|v-E_0|^2}{2}}, \quad (6.5)$$

$$\rho_0(x) = \frac{\sqrt{2\pi}}{2} (2 + \cos(2\pi x)), \quad (6.6)$$

where E_0 satisfies (6.3) with

$$\eta(x) = \frac{\sqrt{2\pi}}{1.2661} e^{\cos(2\pi x)}. \quad (6.7)$$

We evolve the initial condition with a rank of 5, until time $t = 0.01$, which is enough to demonstrate that the asymptotic limit is captured. $N_x = N_v = 128$ grid points are used in each direction. The timestep chosen is $\Delta t = 3.9 \times 10^{-4}$. The evolution of singular values in Fig. 2a shows that the solution maintains a clear separation between the first singular value and the rest. Fig. 2b demonstrates the same behavior, but for a solution beginning in the counterstreaming beams initial condition, (6.1), evolved with rank 10. The solution takes slightly longer to “settle down”, but after three time steps it shows the same rank separation as in the equilibrium case.

6.1.3. Comparison of fluid and kinetic regimes

Conversely to the clear singular value separation observed in the fluid regime for very small ϵ , solutions in the kinetic regime exhibit slower singular value decay. To demonstrate, we consider a “bump-on-tail” initial condition evolving in both the fluid and kinetic regimes. In the fluid regime, the bump disappears within a single timestep, and the slow time scale dynamics of the limiting fluid equation take over. In the kinetic regime, we observe the shearing behavior characteristic of low-collision phase space flows. Our method is designed to capture the low-rank structure inherent in the fluid equation, and so it is not surprising that the kinetic solution requires a higher rank to capture effectively.

The “bump-on-tail” distribution is defined by

$$f_0(x, v) = \frac{\rho_0(x)}{(2\pi)^{1/2}} \left(e^{-\frac{|v|^2}{2}} + e^{-\frac{|v-1.5|^2}{2T_{\text{cold}}}} \right), \quad (6.8)$$

where the temperature of the perturbation is $T_{\text{cold}} = .005$. The density $\rho_0(x)$ is initialized to a Gaussian pulse centered at $x = 0.3$. To induce dynamics in the limiting fluid equation, we initialize the background charge density η with a potential well centered at $x = 0.6$:

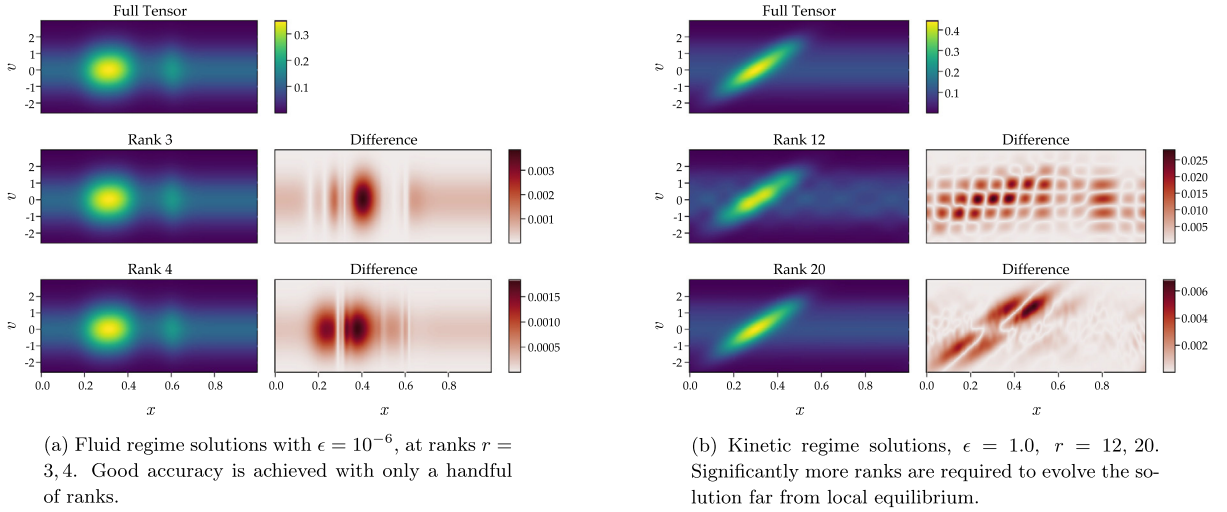


Fig. 3. Comparison of the rank r required for a solution in the fluid (left) and kinetic (right) regime. The fluid solution with $\epsilon = 10^{-6}$ requires significantly lower rank, demonstrating the ability of our approach to capture the fluid limit without excessive computation, by seeking a low-rank approximation to $g = f/M$ instead of f itself. Solutions are computed with $N_v = 128$ and $N_x = 256$. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

$$\rho_0(x) = 0.3 + e^{-\frac{|x-0.3|^2}{0.01}}, \quad \eta(x) = 0.3 + e^{-\frac{|x-0.6|^2}{0.01}}. \quad (6.9)$$

The electric field is initialized via the solution to Poisson's equation (6.3), as above.

The solution is computed with $N_x = 256$ grid points in the x coordinate, and $N_v = 128$ in the v coordinate. We use $\epsilon = 10^{-6}$ to demonstrate the fluid regime behavior, and in the kinetic regime we use $\epsilon = 1$. Both of our low-rank solutions are compared to a full tensor solution of (1.11) with the same discretization parameters, using the scheme proposed in [13]. The solution is run until $t = 0.1$ in the fluid regime, and $t = 0.5$ in the kinetic regime, with time steps of $\Delta t = 3.9 \times 10^{-4}$ in all cases. Results are shown in Fig. 3. We find that as expected, only a handful of ranks are required to obtain good accuracy in the fluid regime. On the other hand, the kinetic solution requires around $r = 20$ for this problem.

6.2. 2D2V examples

6.2.1. Climbing an electrostatic potential hill

To demonstrate that our method can handle nontrivial dynamics in the kinetic regime, we consider the problem of a density “pulse” climbing an electrostatic potential hill. We initialize the background density η uniform everywhere except for a band through the center of the domain where it is set to zero. This creates a region of negative charge density through which the electron fluid cannot pass, unless it has enough inertia to do so. Since in the fluid limit inertial forces vanish, this will only occur in the kinetic regime. We use an elongated Gaussian initial density centered to the left of the potential hill and oriented obliquely to the grid:

$$\rho_0(x, t) = 0.1 + \frac{0.0003}{2\pi|\Sigma|} e^{-\frac{(x-x_0)^T \Sigma^{-1} (x-x_0)}{2}}, \quad x \in [0, 1]^2, \quad (6.10)$$

where $x_0 = [0.3, 0.3]$, $\Sigma = \mathbf{R} \Lambda \mathbf{R}^{-1}$, Λ is a diagonal matrix with entries $[0.006, 0.03]$, and $\mathbf{R}$ a rotation matrix through an angle of $\pi/4$. We initialize a uniform Maxwellian velocity distribution throughout the domain so that the pulse is traveling along the direction of its major axis:

$$f_0(x, v) = \frac{\rho_0(x)}{2\pi T} e^{-\frac{|v-u_0|^2}{2T}}, \quad v \in [-5, 5]^2, \quad (6.11)$$

where $T = 0.01$ and $u_0 = [0.5, 0.5]^T$. The background density η is initialized constant on its support, which is the entire domain excluding a band between $x_l = 0.55$ and $x_r = 0.7$. This creates the potential hill which a kinetic distribution is able to pass over, while the fluid solution remains on the left side, where it starts.

To illustrate both regimes, we use the values $\epsilon = 1.0$ and $\epsilon = 0.01$, and a computational domain with $N = 72$ grid points in each of the four coordinates. We use a fixed time step of $\Delta t = 6.9 \times 10^{-4}$. The results at time $T = 0.35$ are shown in Fig. 4. As expected, the kinetic solution retains a significant flow velocity throughout the domain, and its inertia carries it over the $x = 0.7$ line. On the other hand the fluid solution is pushed out of the region of negative charge density by electrostatic forces. The presence of the potential hill in the interval $x \in [0.55, 0.7]$ is clearly visible in the density plots for the fluid regime.

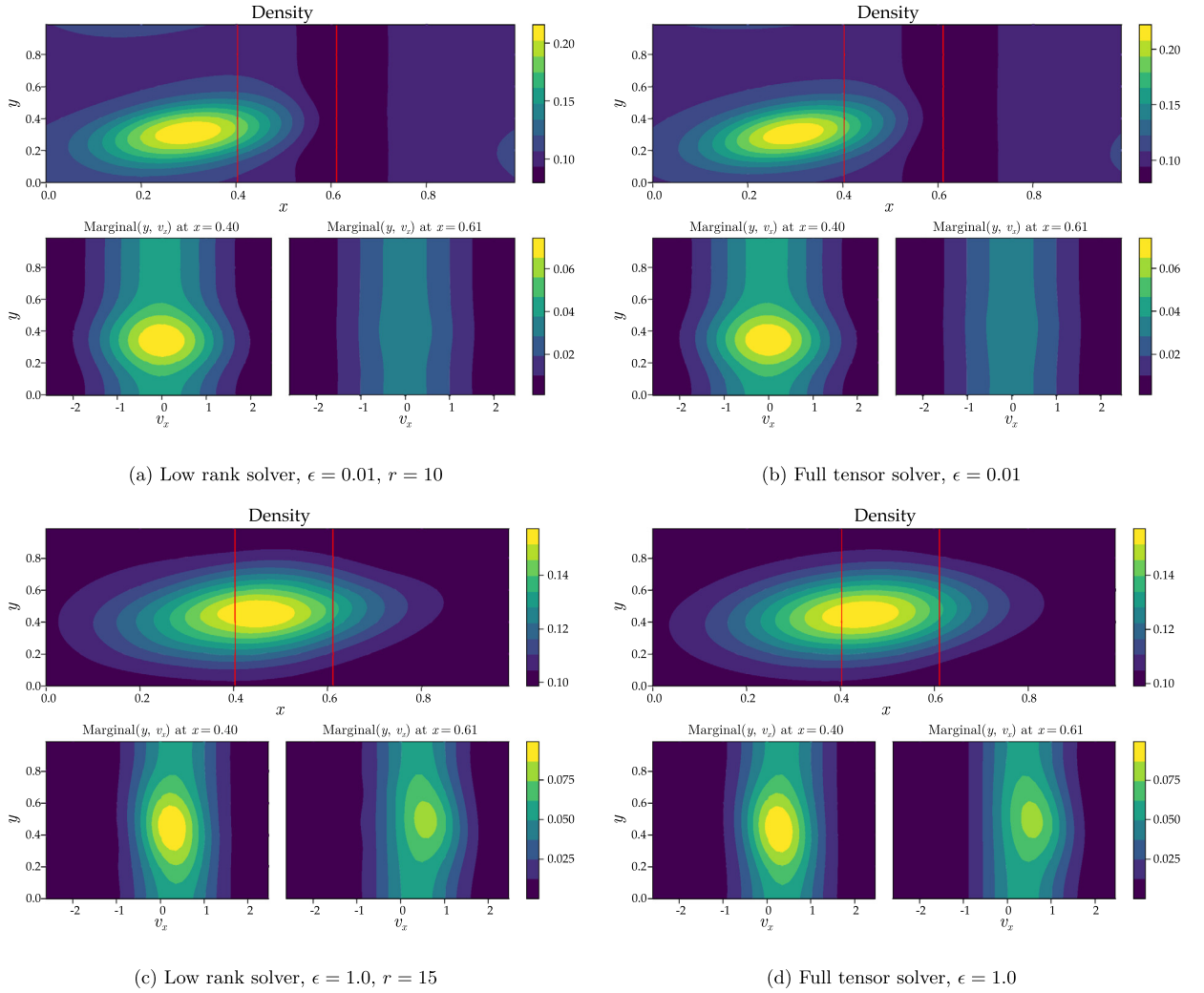


Fig. 4. Results of the potential hill problem at $T = 0.35$, comparing low rank (left) and full tensor (right) solutions. Marginal distributions of f as a function of y, v_x are obtained by integrating with respect to v_y while holding x fixed. The marginals are calculated along two lines at $x \approx 0.4, x \approx 0.6$, indicated in red. The marginal distributions are shown in the bottom pair of plots for each case. We see good agreement with the full tensor solution and with our heuristic predictions for what each flow should do.

Our dynamical low-rank method shows its computational advantages on this 2D2V problem, even for small problem sizes. Timings for a single timestep are detailed in Table 1. We observe that the computational cost of the algorithm is $\mathcal{O}(r^2 N_x + r^2 N_v)$, compared to the $\mathcal{O}(N_x N_v)$ of the full tensor algorithm. The constant factors are small enough to already be dominated at $N_x = N_v = 24^2$.

6.2.2. Relaxation of a cold beam

To demonstrate the relaxation of the solution towards the local Maxwellian in 2 velocity dimensions, we consider a “cold beam” initial condition:

$$f(x, v, 0) = e^{-\frac{|v-u|^2}{0.5}}, \quad (6.12)$$

$$u = [4, 2]^T. \quad (6.13)$$

This is discretized on a doubly spatially periodic unit domain $[0, 1]^2 \times [-10, 10]^2$, with $N_x = 32^2$ spatial grid points dimension, and $N_v = 128^2$ velocity grid points. We choose $\Delta t \approx 0.0008$, and evolve the distribution until $T = 0.3$. The difference between the solution f and the local equilibrium distribution M is plotted for three intermediate points in time, along with the history of the L^1 norm of the difference. It can be seen that the deviation from local equilibrium decays exponentially. The results are shown in Fig. 5.

Table 1

Comparison of single-timestep runtimes on the 2D2V potential hill problem of Section 6.2.1. N is the number of grid points in each dimension, so that $N_x = N_y = N^2$. To facilitate comparison, runtimes are normalized to the runtime of the $r = 5$, $N = 24$ case, which was 16.3 ms. For example, the runtime of the full tensor step for $N = 24$ was 13.8 times longer than that of the rank 5 low-rank step. The bolded numbers are the empirical exponent of N . We see agreement with the expected asymptotic complexity of 2 for the low-rank case, compared with 4 for the full tensor solver.

N	Low rank			Full tensor
	$r = 5$	$r = 10$	$r = 15$	
24	1.00 / –	2.14 / –	3.89 / –	13.8 / –
48	3.21 / 1.7	7.94 / 1.9	14.4 / 1.9	256 / 4.2
72	10.5 / 2.1	23.6 / 2.2	46.7 / 2.3	1.5×10^3 / 4.3
96	18.6 / 2.1	38.3 / 2.1	54.7 / 1.9	5.3×10^3 / 4.3
120	29.2 / 2.1	52.2 / 2.0	88.9 / 1.9	1.23×10^4 / 4.2

Timings were measured on an 8-core Apple M1 Pro, using Julia 1.7.2. Code is available upon request.

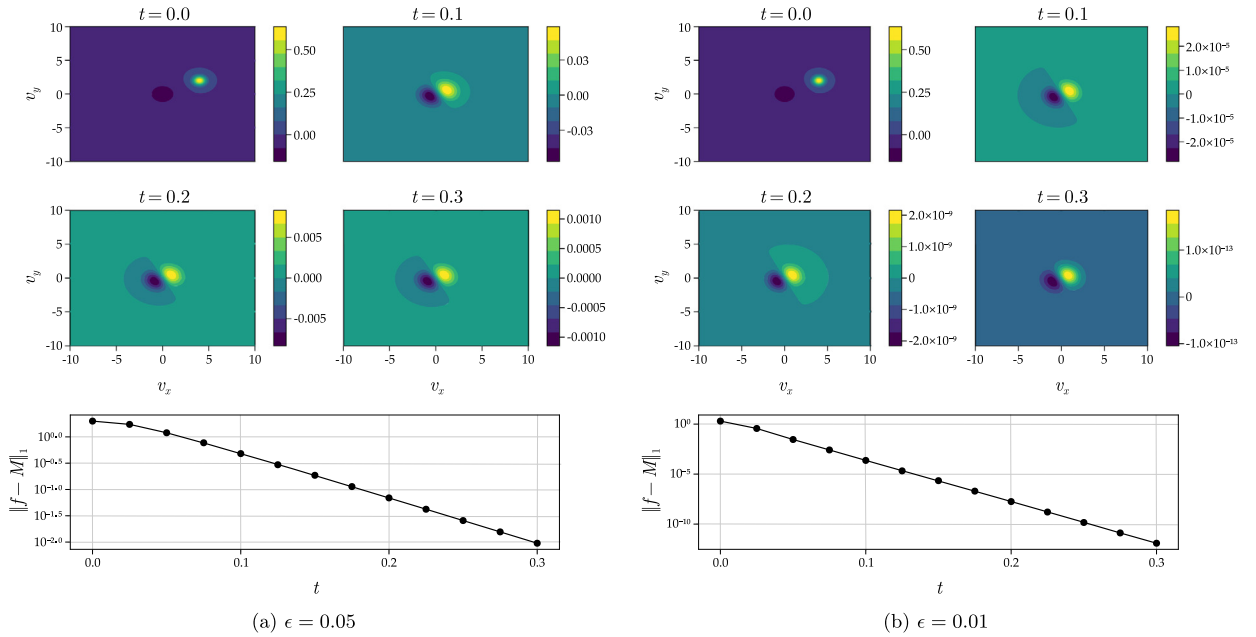


Fig. 5. Comparison of the characteristic relaxation rates of a cold beam in two velocity dimensions. Heatmaps are the raw difference between f and f^0 , demonstrating convergence to the Maxwellian.

7. Conclusion

We have proposed and implemented an efficient algorithm for the electrostatic Vlasov equation with linear Fokker-Planck collision operator. By dividing by the Maxwellian, we are able to represent the quotient with a low-rank approximation, thereby capturing the fluid limit with very little computational effort. Moreover, our method is also efficient when far from the fluid limit, owing to the great reduction in computational complexity afforded by the dynamical low-rank method. In order to get an efficient overall algorithm, we used the fact that the Maxwellian limit of our equation is isothermal to quickly compute a convolution with the Fast Fourier Transform. Our implementation is found to be multiple orders of magnitude faster than a full-tensor numerical solution, with better asymptotic scaling and constant coefficients for moderately sized problems.

CRedit authorship contribution statement

JC and JH designed the research. JC implemented the method. JC and JH wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A. Timestepping for the backwards-in-time S step

In this section we motivate the choice of a Forward Euler timestep for the stiff part of the S flow in (3.24). Recall that the S step evolves just the singular values of the solution backwards in time, and comes in between the K and L steps:

$$\begin{aligned} K \text{ step: } & g^n \xrightarrow{X^n \rightarrow X^{n+1}, S^n \rightarrow S^1} g^1 \\ S \text{ step: } & g^1 \xrightarrow{S^1 \rightarrow S^2} g^2 \\ L \text{ step: } & g^2 \xrightarrow{V^n \rightarrow V^{n+1}, S^2 \rightarrow S^{n+1}} g^{n+1} \end{aligned} \quad (\text{A.1})$$

Because the S step is backwards in time, strictly speaking it is an ill-posed ODE. However, empirically we find that an IMEX step for K followed by a Forwards Euler step for S is stable. To motivate this with a heuristic argument, consider an initial condition that is uniform in space, with vanishing electric field and current. We also assume that the solution begins in local equilibrium. In terms of our low-rank method, we take

$$g(x, v, t_0) = 1, \quad E(x) = J(x) = 0. \quad (\text{A.2})$$

The low-rank decomposition of g gives us $S_{ij} = \delta_{i1}\delta_{j1}$, i.e. S_{11} is the only nonzero entry of S . With no spatial dependence, all but the collisional term of (3.16) drop out, and we are left with

$$\partial_t K_j = \frac{1}{\epsilon} \sum_l d_{jl}^1 K_l, \quad (\text{A.3})$$

where d_{jl}^1 is defined in (3.11). The time evolution equation (3.23) for S also simplifies:

$$\partial_t S_{ij} = -\frac{1}{\epsilon} \sum_{kl} \delta_{ik} d_{jl}^1 S_{kl} = -\frac{1}{\epsilon} \sum_l d_{jl}^1 S_{il}. \quad (\text{A.4})$$

It is useful to rewrite these equations in matrix form. Define the matrices $\mathbf{X} \in \mathbb{R}^{N_x \times r}$, $\mathbf{S} \in \mathbb{R}^{r \times r}$, $\mathbf{V} \in \mathbb{R}^{N_v \times r}$. Then $g = \mathbf{X}\mathbf{S}\mathbf{V}^T = \mathbf{K}\mathbf{V}^T$. Further define $\mathbf{D} = \{d_{jl}^1\}$. The K flow and S flow are given by

$$\partial_t \mathbf{K} = \frac{1}{\epsilon} \mathbf{K} \mathbf{D}^T, \quad (\text{A.5})$$

$$\partial_t \mathbf{S} = -\frac{1}{\epsilon} \mathbf{S} \mathbf{D}^T. \quad (\text{A.6})$$

During the K step and the S step, the respective time derivatives of g are equal and opposite:

$$K \text{ step: } \quad \partial_t g = (\partial_t \mathbf{K}) \mathbf{V}^T = \frac{1}{\epsilon} \mathbf{K} \mathbf{D}^T \mathbf{V}^T = \frac{1}{\epsilon} \mathbf{X} \mathbf{S} \mathbf{D}^T \mathbf{V}^T, \quad (\text{A.7})$$

$$S \text{ step: } \quad \partial_t g = \mathbf{X} (\partial_t \mathbf{S}) \mathbf{V}^T = -\frac{1}{\epsilon} \mathbf{X} \mathbf{S} \mathbf{D}^T \mathbf{V}^T. \quad (\text{A.8})$$

It follows that at the continuous level, our low-rank approximation has the property that $g^n = g^2$ for spatially homogeneous starting point g^n . We choose our time discretization to preserve this invariant. Using the backward Euler for the K flow results in

$$\mathbf{K}^{n+1} = \mathbf{K}^n \left(\mathbf{I} - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right)^{-1}, \quad (\text{A.9})$$

while using the forward Euler for the S flow results in

$$\mathbf{S}^2 = \mathbf{S}^1 \left(\mathbf{I} - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right). \quad (\text{A.10})$$

Therefore,

$$\begin{aligned}
 g^2 &= \mathbf{X}^{n+1} \mathbf{S}^2 (\mathbf{V}^n)^T \\
 &= \mathbf{X}^{n+1} \mathbf{S}^1 \left(I - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right) (\mathbf{V}^n)^T \\
 &= \mathbf{K}^{n+1} \left(I - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right) (\mathbf{V}^n)^T \\
 &= \mathbf{K}^n \left(I - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right)^{-1} \left(I - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right) (\mathbf{V}^n)^T \\
 &= g^n.
 \end{aligned}$$

While we have exact cancellation of the K and S flows for this spatially homogeneous equilibrium, each of these substeps is quite large when taken individually. The stiffness of the ϵ^{-1} term means that it is quite important to preserve this cancellation at the numerical level, otherwise the method is unable to hold even a spatially homogeneous equilibrium. For example, if one uses an IMEX step for the S flow, one finds

$$\begin{aligned}
 g^2 &= \mathbf{K}^n \left(I - \frac{\Delta t}{\epsilon} \mathbf{D}^T \right)^{-1} \left(I + \frac{\Delta t}{\epsilon} \mathbf{D}^T \right)^{-1} (\mathbf{V}^n)^T \\
 &= \mathbf{K}^n \left(I - \left(\frac{\Delta t}{\epsilon} \right)^2 (\mathbf{D}^T)^2 \right)^{-1} (\mathbf{V}^n)^T,
 \end{aligned}$$

which is very far from the identity indeed. The preceding argument is not a rigorous justification of our choice of timestepping scheme for general solutions g . We simply wish to highlight one subtle numerical aspect of the projector-splitting approach which implementors should be aware of.

A.1. Comparison to the BGK operator

In [10], the authors successfully used an IMEX step to advance the S flow. The issues raised above do not arise for the BGK-type operators considered there, as we demonstrate here with a simple example. A spatially homogeneous equation with BGK-type collision operator is

$$\partial_t f = \frac{1}{\epsilon} (M - f),$$

or, using the fact that M is constant for a spatially homogeneous problem,

$$\partial_t g = \frac{1}{\epsilon} (1 - g).$$

Projecting this onto the low-rank approximation gives the following subflows for K and S :

$$\partial_t K_j = \frac{1}{\epsilon} (\langle V_j \rangle_v - K_j), \quad \partial_t S_{ij} = -\frac{1}{\epsilon} (\langle X_i V_j \rangle_{xv} - S_{ij}).$$

An IMEX (backwards Euler) step for each of these subflows will give

$$\begin{aligned}
 \mathbf{K}^{n+1} &= \left(1 + \frac{\Delta t}{\epsilon} \right)^{-1} \left(\mathbf{K}^n + \frac{\Delta t}{\epsilon} \langle (\mathbf{V}^n)^T \rangle_v \right), \\
 \mathbf{S}^2 &= \left(1 - \frac{\Delta t}{\epsilon} \right)^{-1} \left(\mathbf{S}^1 - \frac{\Delta t}{\epsilon} \langle \mathbf{X}^{n+1} (\mathbf{V}^n)^T \rangle_{xv} \right).
 \end{aligned}$$

Plugging these into the expression for g^2 , we find

$$\begin{aligned}
 g^2 &= \mathbf{X}^{n+1} \mathbf{S}^2 (\mathbf{V}^n)^T \\
 &= \mathbf{X}^{n+1} \left(\mathbf{S}^1 - \frac{\Delta t}{\epsilon} \langle \mathbf{X}^{n+1} (\mathbf{V}^n)^T \rangle_{xv} \right) \left(1 - \frac{\Delta t}{\epsilon} \right)^{-1} (\mathbf{V}^n)^T \\
 &= \mathbf{K}^{n+1} \left(1 - \frac{\Delta t}{\epsilon} \right)^{-1} (\mathbf{V}^n)^T - \mathbf{X}^{n+1} \left(\frac{\epsilon}{\Delta t} - 1 \right)^{-1} \langle \mathbf{X}^{n+1} (\mathbf{V}^n)^T \rangle_{xv} (\mathbf{V}^n)^T \\
 &= \mathbf{X}^{n+1} \langle 1, \mathbf{X}^{n+1} (\mathbf{V}^n)^T \rangle_{xv} (\mathbf{V}^n)^T + O\left(\frac{\epsilon}{\Delta t}\right).
 \end{aligned}$$

Note that we have dropped terms of order $\epsilon/\Delta t$, to illustrate that g is driven to within $\epsilon/\Delta t$ of its equilibrium value, which is 1 (projected onto the low-rank bases). Because the BGK operator on g is affine rather than linear there is no cancellation, but the IMEX approach for both flows poses no problems in the $\epsilon \rightarrow 0$ limit.

References

- [1] Carlo Cercignani, *The Boltzmann Equation and Its Applications*, Applied Mathematical Sciences, vol. 67, Springer New York, New York, NY, 1988, ISBNs 978-1-4612-6995-3, 978-1-4612-1039-9.
- [2] Carlo Cercignani, Irene M. Gamba, Joseph W. Jerome, Chi-Wang Shu, Device benchmark comparisons via kinetic, hydrodynamic, and high-hield models, *Comput. Methods Appl. Mech. Eng.* (ISSN 0045-7825) 181 (4) (January 2000) 381–392, [https://doi.org/10.1016/S0045-7825\(99\)00186-3](https://doi.org/10.1016/S0045-7825(99)00186-3).
- [3] Gianluca Ceruti, Christian Lubich, An unconventional robust integrator for dynamical low-rank approximation, *BIT Numer. Math.* 62 (1) (March 2022) 23–44, <https://doi.org/10.1007/s10543-021-00873-0>, ISSN 0006-3835, 1572-9125.
- [4] Zhiyan Ding, Lukas Einkemmer, Qin Li, Dynamical low-rank integrator for the linear Boltzmann equation: error analysis in the diffusion limit, *SIAM J. Numer. Anal.* 59 (4) (January 2021) 2254–2285, <https://doi.org/10.1137/20M1380788>, ISSN 0036-1429, 1095-7170.
- [5] J.P. Dougherty, Model Fokker-Planck equation for a plasma and its solution, *Phys. Fluids* (ISSN 0031-9171) 7 (11) (1964) 1788, <https://doi.org/10.1063/1.2746779>.
- [6] Lukas Einkemmer, A low-rank algorithm for weakly compressible flow, *SIAM J. Sci. Comput.* 41 (5) (January 2019) A2795–A2814, <https://doi.org/10.1137/18M1185417>, ISSN 1064-8275, 1095-7197.
- [7] Lukas Einkemmer, Ilon Joseph, A mass, momentum, and energy conservative dynamical low-rank scheme for the Vlasov equation, *J. Comput. Phys.* (ISSN 0021-9991) 443 (October 2021) 110495, <https://doi.org/10.1016/j.jcp.2021.110495>.
- [8] Lukas Einkemmer, Christian Lubich, A low-rank projector-splitting integrator for the Vlasov–Poisson equation, *SIAM J. Sci. Comput.* 40 (5) (January 2018) B1330–B1360, <https://doi.org/10.1137/18M116383X>, ISSN 1064-8275, 1095-7197.
- [9] Lukas Einkemmer, Jingwei Hu, Yubo Wang, An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation, *J. Comput. Phys.* (ISSN 0021-9991) 439 (August 2021) 110353, <https://doi.org/10.1016/j.jcp.2021.110353>.
- [10] Lukas Einkemmer, Jingwei Hu, Lexing Ying, An efficient dynamical low-rank algorithm for the Boltzmann-BGK equation close to the compressible viscous flow regime, *SIAM J. Sci. Comput.* 43 (5) (January 2021) B1057–B1080, <https://doi.org/10.1137/21M1392772>, ISSN 1064-8275, 1095-7197.
- [11] R.J. Goldston, P.H. Rutherford, *Introduction to Plasma Physics*, Institute of Physics Pub, Bristol, UK; Philadelphia, ISBN 978-0-7503-0183-1, 1995.
- [12] Jingwei Hu, Yubo Wang, An adaptive dynamical low rank method for the nonlinear Boltzmann equation, *J. Sci. Comput.* 92 (2) (August 2022) 75, <https://doi.org/10.1007/s10915-022-01934-4>, ISSN 0885-7474, 1573-7691.
- [13] Shi Jin, Li Wang, An asymptotic preserving scheme for the Vlasov-Poisson-Fokker-Planck system in the high field regime, *Acta Math. Sci.* (ISSN 0252-9602) 31 (6) (November 2011) 2219–2232, [https://doi.org/10.1016/S0252-9602\(11\)60395-0](https://doi.org/10.1016/S0252-9602(11)60395-0).
- [14] Othmar Koch, Christian Lubich, Dynamical low-rank approximation, *SIAM J. Matrix Anal. Appl.* 29 (2) (January 2007) 434–454, <https://doi.org/10.1137/050639703>, ISSN 0895-4798, 1095-7162.
- [15] Othmar Koch, Christian Lubich, Dynamical tensor approximation, *SIAM J. Matrix Anal. Appl.* 31 (5) (January 2010) 2360–2375, <https://doi.org/10.1137/09076578X>, ISSN 0895-4798, 1095-7162.
- [16] Randall J. LeVeque, *Numerical Methods for Conservation Laws*, 2nd edition, Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel; Boston, 1992, ISBNs 978-3-7643-2723-1, 978-0-8176-2723-2.
- [17] Christian Lubich, Ivan V. Oseledets, A projector-splitting integrator for dynamical low-rank approximation, *BIT Numer. Math.* 54 (1) (March 2014) 171–188, <https://doi.org/10.1007/s10543-013-0454-0>, ISSN 0006-3835, 1572-9125.
- [18] Zhuogang Peng, Ryan G. McClarren, Martin Frank, A low-rank method for two-dimensional time-dependent radiation transport calculations, *J. Comput. Phys.* (ISSN 0021-9991) 421 (November 2020) 109735, <https://doi.org/10.1016/j.jcp.2020.109735>.
- [19] F. Poupaud, Runaway phenomena and fluid approximation under high fields in semiconductor kinetic theory, *Z. Angew. Math. Mech.* 72 (8) (1992) 359–372, <https://doi.org/10.1002/zamm.19920720813>, ISSN 0044-2267, 1521-4001.
- [20] Marshall N. Rosenbluth, William M. MacDonald, David L. Judd, Fokker-Planck equation for an inverse-square force, *Phys. Rev.* (ISSN 0031-899X) 107 (1) (July 1957) 1–6, <https://doi.org/10.1103/PhysRev.107.1>.
- [21] Youcef Saad, Martin H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (3) (July 1986) 856–869, <https://doi.org/10.1137/0907058>, ISSN 0196-5204, 2168-3417.
- [22] Lloyd N. Trefethen, David Bau, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, ISBN 978-0-89871-361-9, 1997.
- [23] Cédric Villani, A review of mathematical topics in collisional kinetic theory, in: *Handbook of Mathematical Fluid Dynamics*, vol. 1, Elsevier, ISBN 978-0-444-50330-5, 2002, pp. 71–74.