An optimization-based positivity-preserving limiter in semi-implicit discontinuous Galerkin schemes solving Fokker–Planck equations

Chen Liu[a], Jingwei Hu[b], William T. Taitano[c], Xiangxiong Zhang[d,]

[a]*Department of Mathematical Sciences, University of Arkansas, Fayetteville, Arkansas 72701.*
[b]*Department of Applied Mathematics, University of Washington, Seattle, WA 98195.*
[c]*Applied Mathematics and Plasma Physics Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545.*
[d]*Department of Mathematics, Purdue University, West Lafayette, Indiana 47907.*

Abstract

For high-order accurate schemes such as discontinuous Galerkin (DG) methods solving Fokker–Planck equations, it is desired to efficiently enforce positivity without losing conservation and high-order accuracy, especially for implicit time discretizations. We consider an optimization-based positivity-preserving limiter for enforcing positivity of cell averages of DG solutions in a semi-implicit time discretization scheme, so that the point values can be easily enforced to be positive by a simple scaling limiter on the DG polynomial in each cell. The optimization can be efficiently solved by a first-order splitting method with nearly optimal parameters, which has an $O(N)$ computational complexity and is flexible for parallel computation. Numerical tests are shown on some representative examples to demonstrate the performance of the proposed method.

Keywords: Fokker–Planck equations, discontinuous Galerkin, high-order accuracy, positivity-preserving, optimization-based limiter, Douglas–Rachford splitting

1. Introduction

1.1. Motivation and objective

The Fokker–Planck equation is an advection-diffusion equation that describes the transport of probability distribution functions in state space. It has wide applications in modeling stochastic collisional transport of particles in plasmas [1, 2] and photons [3]. In photon transport, it is often used to model Compton scattering against hot electrons, which is typically encountered in high-temperature environments in stars and in high energy density (HED) experiments. In plasmas, the Landau/Rosenbluth Fokker–Planck (L/RFP) operator describes first-principles small-angle Coulomb scattering between charged particles and is considered the gold standard for describing collisional transport processes. As such, the Fokker–Planck model has wide applications in studying fusion plasmas – both magnetic and inertial confinement approaches (ICF) – laboratory HED systems, astrophysical systems, and space systems, among many others. In particular, in ICF applications, the RFP collision operator is used to regularly probe the impacts of long-mean-free path charged-particle effects on important observable performance metrics such as fusion yield and critical neutron spectrum information to infer plasma states [4, 5, 6, 7, 8, 9, 10, 11, 12] in state-of-the-art Vlasov–Rosenbluth–Fokker–Planck codes such as FPion/Fuse [13] and iFP [14]. For physical systems such as in plasmas, the Fokker–Planck operator adheres to rigorous conservation principles such as mass, momentum, and energy while maximizing the entropy of the system and maintaining the positivity of the distribution function. The critical importance of preserving these properties for the Fokker–Planck operator has been

Email addresses: chenl@uark.edu (Chen Liu), hujw@uw.edu (Jingwei Hu), taitano@lanl.gov (William T. Taitano), zhan1966@purdue.edu (Xiangxiong Zhang)

extensively demonstrated for the RFP model for plasmas, as shown in [15, 16]. Of particular relevance to this work is the importance of preserving the positivity of the distribution function. In the RFP model, the diffusion tensor is symmetric-positive-definite (SPD) and is a functional of the distribution. When significant negative distributions are present, the SPD property can be violated, leading to negative diffusion coefficients that lead to numerical instabilities and are demonstrated in [15]. The objective of this paper is to study an optimization-based positivity preserving approach to stabilize a high-order accurate scheme.

1.2. Several existing approaches for preserving bounds

Over the past several decades, researchers have devoted significant efforts to exploring various approaches in the construction of numerical techniques that preserve bounds or positivity when solving partial differential equations (PDEs). Without being exhaustive, let us briefly review several existing approaches. Each of these approaches has broad and successful applications to many different types of PDEs.

One of the most popular approaches to construct a bound-preserving high-order finite volume type method or discontinuous Galerkin (DG) method for conservation laws was introduced by Zhang and Shu in [17, 18], see also [19, 20, 21, 22, 23]. For explicit high order DG methods, a *weak monotonicity* property holds in the sense that a CFL constraint on time step size ensures that the cell average of the discrete solution is a convex combination of monotone first order schemes, thus preserving bounds of cell average [17, 18, 23]. When the cell average (zeroth order moment) of a DG solution are within desired bounds, a simple scaling limiter can be used to modify the high order moments and obtain a bound-preserving DG polynomial without affecting the cell average and high order accuracy. For high-order DG schemes with explicit time-stepping to solve a convection diffusion problem, the weak monotonicity and the Zhang–Shu method can be extended to a third-order accurate direct DG method [24] if using a linear flux and to arbitrarily high-order DG methods if using a nonlinear flux [25, 26].

For an implicit time stepping, such as the backward Euler method, a systemic approach to obtain a sufficient condition of the discrete maximum principle is to show the monotonicity of the system matrix. A matrix is called monotone if all entries of its inverse are nonnegative. Many second-order schemes, such as the classical central finite difference method or continuous finite element method, for discretizing the Laplace operator provide an M-matrix, thus are monotone. The monotonicity of $\mathbb{Q}^1$ interior penalty DG method on multi-dimensional structured meshes has been established in [27]. The monotonicity of the spectral element method with $\mathbb{Q}^2$ and $\mathbb{Q}^3$ elements has been proven in [28, 29, 30]. A monotone $\mathbb{Q}^1$ finite element method for the anisotropic elliptic equations was constructed in [31]. The monotonicity of spectral element method has been used to construct positivity-preserving schemes in first-order implicit time stepping with high-order spatial accuracy for various second-order equations such as Allen–Cahn [32], Keller–Segel [33], Fokker–Planck [34] and also compressible Navier–Stokes [27]. See a recent review in [35] for the provable monotonicity results of the continuous finite element method. However, monotonicity of high-order finite element methods on unstructured meshes does not hold [36]. In addition, for many higher-order implicit time marching strategies, a monotone spatial discretization may not be enough to preserve bounds using a time step like $\Delta t = O(\Delta x)$, e.g., the Crank–Nicolson method with a monotone spatial discretization preserves positivity only if the time step is as small as $\Delta t = O(\Delta x^2)$, see [37, Appendix B] and [38, Section 5.3].

Another approach for constructing bound-preserving schemes is flux limiting. We refer to [39] for an overview of several classical techniques, including the flux limiting, for the finite element method, in solving convection-diffusion equations. Kuzmin et al. in [40] introduced an algebraic flux correction for the finite element method to enforce an M-matrix structure that preserves the discrete maximum principle for solving anisotropic elliptic equations, and numerical experiments showed a second-order accuracy. For continuous finite element and DG methods, flux limiters have been designed and applied to compressible Navier–Stokes [38, 41], Cahn–Hilliard [42], coupled phase-field model [43], and many other equations [44, 45]. In numerical experiments, it is often observed that the application of flux limiters appropriately does not harm the convergence rate, though a rigorous justification is available only for simpler equations [46].

In recent years, optimization-based approaches have become popular. Guba et al. in [47] designed a bound-preserving limiter for spectral element method, implemented by standard quadratic programming solvers. van der Vegt et al. in [48] formulated the positivity constraints in the KKT system for the DG method with implicit time integration, solved by an active set semismooth Newton method. Cheng and

Shen in [49] introduced a Lagrange multiplier approach to preserve bounds for semilinear and quasi-linear parabolic equations, providing a new interpretation for the cut-off method. Ruppenthal and Kuzmin in [50] considered optimization-based flux correction to ensure the positivity of finite element discretization of scalar conservation laws. Kirby and Shapero in [51] enforce bounds for the continuous finite element discretization of convection-diffusion equations using variational inequality constraints, which are subsequently solved by Newton's method via PETSc's reduced space active set solver.

1.3. An optimization based two-stage postprocessing in DG methods

In this paper, we consider the optimization based approach in [52, 53], which is decribed as follows for DG methods.

The DG method is a popular numerical method for solving PDEs. For a comprehensive review of this method, we refer to [54, 55]. The DG method enjoys many attractive properties, such as the ability to easily achieve high-order accuracy, ease of handling complex meshes and hp-adaptivity, highly parallelizable characteristics, and excellent stability and flexibility.

In particular, its flexibility allows easier postprocessing to enforce bounds; e.g., we may first enforce bounds of cell averages and then enforce bounds of point values in each cell. Next, we describe such a two-stage postprocessing to enforce bounds or positivity of a given DG solution without losing global conservation and affecting accuracy. See Figure 1 for a schematic flow chart of this procedure: first, we modify the zeroth-order moment to enforce bounds of cell averages of DG solution polynomials by applying an optimization-based limiter to all cell averages; second, we apply a limiter in each cell to modify the high-order moments to eliminate undershoot and/or overshoot point values.

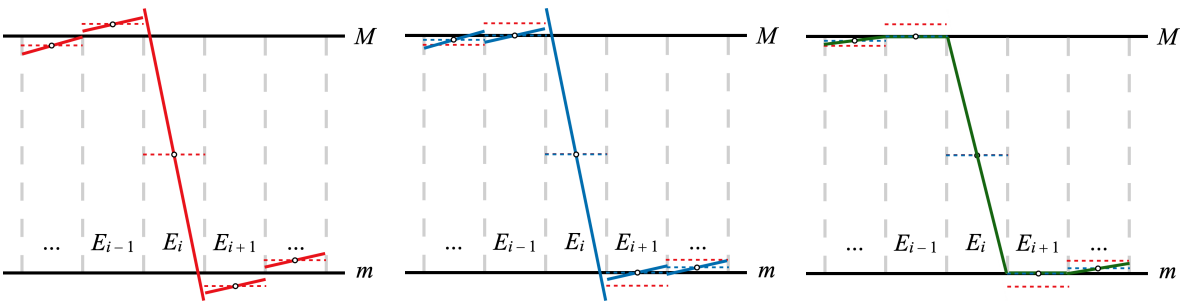

Figure 1: Schematic postprocessing procedure for 1D piecewise linear polynomial. The m and M are desired lower and upper bounds. Left: the original DG solution (red line) with out-of-bound cell averages (red dashed line). Middle: applying an optimization based limiter to modify cell averages. The modified cell averages (blue dashed line) are in $[m, M]$ but the modified DG solution (blue line) may still contains out-of-bound point values. Right: in each cell, apply a limiter to each DG polynomial to eliminate undershoot and/or overshoot, which gives a bound-preserving solution (green line).

To be precise, the two-stage postprocessing is defined and given as follows. Given a DG solution f_h and a desired range interval $[m, M]$, define $\overline{f_h}$ as a piecewise constant representing the cell average of f_h in each cell E, that is, $\overline{f_h}|_E = \frac{1}{|E|}\int_E f_h$. If there exists a cell E such that $\overline{f_h}|_E \notin [m, M]$, then we seek a piecewise constant x_h that minimizes the L^2 distance to $\overline{f_h}$ under the constraints of conserving conservation and bounds:

$$\min_{x_h} \|x_h - \overline{f_h}\|_{L^2}^2 \quad \text{subjects to} \quad \int_\Omega x_h = \int_\Omega f_h \quad \text{and} \quad m \le x_h \le M. \tag{1}$$

Let $\overline{w_h}$ be the minimizer to (1). Then, the postprocessed DG solution

$$\widehat{f_h} = (f_h - \overline{f_h}) + \overline{w_h} \tag{2}$$

preserves global conservation and enforces bounds of cell average. Notice, when only seeking to preserve non-negativity, we set $m = 0$ and $M = +\infty$. See Section 3.2 for a justification of accuracy for (2).

The constrained optimization (1) is equivalent to an unconstrained problem by the indicator function. Let ι_Λ denote an indicator function of a set Λ: $\iota_\Lambda(x) = 0$ for $x \in \Lambda$ and $\iota_\Lambda(x) = +\infty$ for $x \notin \Lambda$. Associated with the conservation constraint and the bound-preserving constraint, for the piecewise constant polynomial x_h, define sets $\Lambda_1 = \{x_h : \int_\Omega x_h = \int_\Omega f_h\}$ and $\Lambda_2 = \{x_h : m \le x_h \le M\}$. Then, we have (1) is equivalent to

$$\min_{x_h} \|x_h - \overline{f_h}\|_{L^2}^2 + \iota_{\Lambda_1}(x_h) + \iota_{\Lambda_2}(x_h). \tag{3}$$

Both sets Λ_1 and Λ_2 are convex and closed, thus the function $\iota_{\Lambda_1} + \iota_{\Lambda_2}$ is a proper closed convex function, and the cost function of (3) is a proper closed strongly convex function, which has a unique minimizer by standard convex optimization theory.

Once the cell averages are in the bounds, one can use the simple scaling limiter in [17, 18, 23] to further enforce bounds of point values in each cell. In the literature, such a limiter is sometimes called the Zhang–Shu limiter, which is often used for preserving conservation and bounds in DG and finite volume schemes solving conservation laws, due to its efficiency and high order accuracy [19].

1.4. Efficient optimization solver

Though there are many existing methods for solving the simple constrained L^2-norm minimization (1), ideally we need an easily implementable efficient algorithm, which is also easily parallelizable, since (1) needs to be solved to machine accuracy in each time step. For numerical simulations in multiple dimensions, the size of variable in (1) to be processed at each time step can be large, thus it is generally preferable to use first-order optimization methods as they scale well with N, which is the number of cells. As shown in [52], the minimizer to (1) can be efficiently computed by applying the Douglas–Rachford splitting method [56] to (3), if using optimal algorithms parameters, which can be derived from the asymptotic linear convergence in [52]. The Douglas–Rachford splitting is equivalent to some other popular methods such as PDHG [57], ADMM [58], dual split Bregman method [59], see also [60] and references therein for the equivalence.

Notice that the simple constrained minimization in (1) can also be solved directly via the KKT system of the Lagrangian, which however might be less efficient than the Douglas–Rachford splitting methods for large problems, see a comparison in [53, Appendix]. In addition, the Douglas–Rachford splitting method is extremely simple as the entire scheme can be described in three lines; see (30). This provides a significant practical advantage: a simpler scheme is easier for practitioners to implement and the simplicity makes for a more efficient parallelization. This approach has been shown very efficient for multiphase phase-field flow [52] and compressible Navier–Stokes simulations [53].

1.5. The main results and the organization of this paper

Although bound-preservation or positivity-preservation can be crucial for stability, it is insufficient by itself to produce an accurate numerical solution. In other words, an optimization-based postprocessing described in Section 1.3 should be used on a high-order accurate numerical scheme such as a properly defined DG scheme. The optimization-based postprocessing for enforcing bounds and positivity in Section 1.3 was first applied to DG method solving the Cahn–Hilliard–Navier–Stokes system [52] and compressible Navier–Stokes system [53]. In this paper, we consider a DG method with arbitrarily high-order polynomial basis with a semi-implicit time discretization for solving an equation in the form

$$\partial_t f - \varepsilon^{-1} \boldsymbol{\nabla} \cdot (\mathbf{D} \boldsymbol{\nabla} f) + \varepsilon^{-1} \frac{m}{T} \boldsymbol{\nabla} \cdot \big(\mathbf{D}(\boldsymbol{u} - \boldsymbol{v}) f\big) = 0,$$

where $\mathbf{D}$ is a variable coefficient matrix. We demonstrate that a high-order DG spatial discretization with an implicit time discretization for such an equation can be efficiently stabilized by the optimization based postprocessing described in Section 1.3 for challenging problems, in which loss of positivity often causes instability of DG schemes, and also loss of desired structure such as the SPD property of the diffusion tensor in the nonlinear RFP equation.

The remainder of this paper is organized as follows. In Section 2, we introduce the multispecies Rosenbluth–Fokker–Planck equation and its associated linearized model in convection-diffusion form. In

Section 3, we describe the DG scheme and the optimization-based post-processing to enforce the positivity of the distribution function. In Section 4, numerical tests of a few representative examples are shown to demonstrate the effectiveness of the proposed method. Concluding remarks are given in Section 5.

2. Mathematical model: The Rosenbluth–Fokker–Planck equation

The Landau/Rosenbluth–Fokker–Planck (L/RFP) equation is often considered to be the first principles model to describe the dynamical evolution of plasma particle distribution function subject to Coulomb collision. We start with the RFP equation in its general non-linear form, then derive the simplified linearized form to highlight our positivity-preserving postprocessing algorithm.

Let $\varepsilon > 0$ be the characteristic collision time scale, and let N_s denote the total number of particle species considered. The unknown in the Rosenbluth–Fokker–Planck equation is the distribution function $f = f(t, \boldsymbol{v})$ with independent variables time t and velocity $\boldsymbol{v} = [v_0, \cdots, v_{d-1}]^{\mathrm{T}} \in \mathbb{R}^d$, satisfying:

$$\partial_t f_\alpha = \varepsilon^{-1} \sum_{\beta=1}^{N_s} C(f_\beta, f_\alpha), \tag{4}$$

where the subscripts α and β denote the particle species. The $C(f_\beta, f_\alpha)$ is the Fokker–Planck collision operator of particle species $\alpha \in [1, \cdots, N_s]$ colliding with species $\beta \in [1, \cdots, N_s]$. We use $\partial_i = \frac{\partial}{\partial v_i}$ to denote the partial derivative with respect to v_i, the i^{th} component of velocity. Take the convenience of Einstein notation, i.e., assume the repeated index as summation, we have

$$C(f_\beta, f_\alpha) = \partial_i (J_{\beta\alpha,D,i} - J_{\beta\alpha,A,i}). \tag{5}$$

Let m_α and m_β be the masses of species α and β. The i^{th} component of the diffusion flux and the i^{th} component of the friction flux are given by

$$J_{\beta\alpha,D,i} = D_{\beta,ij}[f_\beta]\, \partial_j f_\alpha \quad \text{and} \quad J_{\beta\alpha,A,i} = \frac{m_\alpha}{m_\beta} A_{\beta,i}[f_\beta]\, f_\alpha. \tag{6}$$

Here, $D_{\beta,ij}[f_\beta](\boldsymbol{v}) = \partial_i \partial_j G_\beta$ and $A_{\beta,i}[f_\beta](\boldsymbol{v}) = \partial_i H_\beta$ are the ij^{th} component of the diffusion coefficient and the i^{th} component of the friction vector, respectively. And $(G_\beta[f_\beta](\boldsymbol{v}), H_\beta[f_\beta](\boldsymbol{v}))$ are the so-called Rosenbluth potentials for species β, evaluated by inverting the coupled Poisson equations [61, 15]

$$\partial_i \partial_i H_\beta = -8\pi f_\beta, \tag{7}$$

$$\partial_i \partial_i G_\beta = H_\beta. \tag{8}$$

In this study, we consider a two-species formulation, where one of the species –test particle– has a vanishing density and imparts negligible impact on the dynamics of itself and the background species. In this limit, a single species distribution function for the test particle is evolved while the background species remains static and provides the collisional transport coefficients to the dynamical species. The limiting linearized form of the RFP equation for the test particle is given as

$$\partial_t f = C(f_b, f) = \epsilon^{-1} \partial_i \left[D_{b,ij} \partial_j f - \frac{m}{m_b} A_{b,i} f \right], \tag{9}$$

where the subscript b denotes the static background species. This limiting form of the RFP equation has applications in modeling slowing and scattering processes of tenuous high-energy particles on massive static background material. This linearized Fokker–Planck collision operator preserves the positivity of the distribution function f and conserves mass, that is, let $\langle \cdot, \cdot \rangle_v$ denote the inner product, then $\langle C(f_\beta, f_\alpha), 1 \rangle_v = 0$, which implies that the integration of f is unchanged with time.

Remark 1. *For a particle species of mass m and density n, the Maxwellian distribution function f^M of independent variable $\boldsymbol{v}$ is defined by*

$$f^M(\boldsymbol{v}; n, \mathbf{u}, T, m) = \frac{n}{(\pi v_{\text{th}}^2)^{d/2}} \exp\Big(- \frac{m}{2T} \|\boldsymbol{v} - \boldsymbol{u}\|_2^2 \Big), \quad \textit{where} \quad v_{\text{th}} = \sqrt{\frac{2T}{m}}. \tag{10}$$

The T represents temperature and $\boldsymbol{u}$ represents velocity, both of which are given. The Maxwellian distribution is the kernel of the collision operator, i.e., the equality $C(f_b^M, f^M) = 0$ holds.

Assuming that the background species is in its own local equilibrium state, i.e., $f_b = f_b^M$ and $C\left(f_b^M, f_b^M\right) = 0 \; \forall \; \boldsymbol{v}$, the first argument of the collision operator in Eq. (9) is a Maxwellian distribution, $f_b = f_b^M$, with a mass of m_b, density n_b, prescribed bulk velocity $\boldsymbol{u}$, and temperature T. Consequently, $D_{b,ij}^M \equiv D_{b,ij}[f_b^M] = \partial_i \partial_j G_b^M$ and $A_{b,i}^M \equiv A_{b,i}[f_b^M] = \partial_i H_b^M$. The H_b^M and G_b^M can be solved analytically utilizing the properties of harmonic and biharmonic functions in spherical coordinates $\boldsymbol{v} = \{v, \theta, \phi\}$, where $v \in \mathbb{R}_+$, $\theta \times \phi \in \mathbb{S}^2$ as:

$$H_b^M(v) = n_b \left[\operatorname{erf}(w_b) / v' \right], \tag{11}$$

and

$$G_b^M(v) = n_b v' \left[\left(1 + \frac{1}{2w_b^2} \right) \operatorname{erf}(w_b) + \frac{e^{-w_b^2}}{\sqrt{\pi} w_b} \right] \tag{12}$$

where $w_b = v' \sqrt{m_b / 2T}$ and $v' = \|\boldsymbol{v} - \boldsymbol{u}\|_2$. Accordingly, the component-wise collisional diffusion tensor in Cartesian coordinates is given as:

$$D_{b,ij}^M = \begin{cases} -n_b v_i' v_j' \left\{ \frac{3e^{-w_b^2}}{w_b v'^3 \sqrt{\pi}} + \left(1 - \frac{3}{2w_b^2} \right) \frac{\operatorname{erf}(w_b)}{v'^3} \right\} & \text{if} \quad i \neq j \\ n_b \left\{ \frac{e^{-w_b^2} \left(v_b'^2 - 3 v_{b,i}'^2 \right)}{v'^3 w_b \sqrt{\pi}} + \left[\frac{3 v_i'^2 - v'^2}{2 w_b^2 v'^3} + \frac{v'^4 - v'^2 v_i'^2}{v'^5} \right] \operatorname{erf}(w_b) \right\} & \text{if} \quad i = j \end{cases} \tag{13}$$

We note that from now on, unless otherwise specified, $n_b = 1$ is used for the rest of the manuscript.

In addition to f^M still being the kernel of the linearized collision operator, we use the fact that the diffusion and friction fluxes cancel out exactly at equilibrium,

$$D_{b,ij}^M \, \partial_j f^M - \frac{m}{m_b} A_{b,i}^M \, f^M = 0.$$

By (10), we have

$$\frac{m}{m_b} A_{b,i}^M = D_{b,ij}^M \, \partial_j \ln f^M = -\frac{m}{T} D_{b,ij}^M (v_j - u_j).$$

Inserting this expression back into (9), we obtain

$$\partial_t f = \varepsilon^{-1} \partial_i \Big[D_{b,ij}^M \Big(\partial_j f + \frac{m}{T} (v_j - u_j) f \Big) \Big]. \tag{14}$$

To this end, let us rewrite (14) in an equivalent convection-diffusion form by using the gradient and divergence operators. We have

$$\partial_t f = \varepsilon^{-1} \partial_i (D_{b,ij}^M \partial_j f) + \varepsilon^{-1} \frac{m}{T} \partial_i \big(D_{b,ij}^M (v_j - u_j) f \big). \tag{15}$$

For simplicity of numerical implementations, we consider a mathematical simplification in two dimensions ($d = 2$) while the derivation of the three-dimensional case is identical. The first term on the right-hand side above becomes the following

$$\partial_i(D^M_{b,ij}\partial_j f) = \sum_{i=0}^{d-1}\partial_i\Big(\sum_{j=0}^{d-1} D^M_{b,ij}\partial_j f\Big) = \nabla\cdot\begin{bmatrix} D^M_{b,00}\partial_0 f + D^M_{b,01}\partial_1 f \\ D^M_{b,10}\partial_0 f + D^M_{b,11}\partial_1 f\end{bmatrix} = \nabla\cdot\left(\overbrace{\begin{bmatrix} D^M_{b,00} & D^M_{b,01}\\ D^M_{b,10} & D^M_{b,11}\end{bmatrix}}^{=\mathbf{D}}\begin{bmatrix}\partial_0 f\\ \partial_1 f\end{bmatrix}\right) = \nabla\cdot(\mathbf{D}\nabla f).$$

The second term on the right-hand side above becomes the following

$$\partial_i\big(D^M_{b,ij}(v_j-u_j)f\big) = \sum_{i=0}^{d-1}\partial_i\Big(\sum_{j=0}^{d-1} D^M_{b,ij}(v_j-u_j)f\Big) = \nabla\cdot\begin{bmatrix} D^M_{b,00}(v_0-u_0)f + D^M_{b,01}(v_1-u_1)f \\ D^M_{b,10}(v_0-u_0)f + D^M_{b,11}(v_1-u_1)f\end{bmatrix} = \nabla\cdot\left(\begin{bmatrix} D^M_{b,00} & D^M_{b,01}\\ D^M_{b,10} & D^M_{b,11}\end{bmatrix}\begin{bmatrix} v_0-u_0\\ v_1-u_1\end{bmatrix} f\right) = \nabla\cdot\big(\mathbf{D}(\boldsymbol{v}-\boldsymbol{u})f\big).$$

Therefore, (14) is equivalent to the following convection-diffusion form.

$$\partial_t f - \varepsilon^{-1}\nabla\cdot(\mathbf{D}\nabla f) + \varepsilon^{-1}\frac{m}{T}\nabla\cdot\big(\mathbf{D}(\boldsymbol{u}-\boldsymbol{v})f\big) = 0. \tag{16}$$

Finally, for the boundary conditions, we supplement $J_{D,i}|_{\partial\Omega_i} = 0$ and $J_{A,i}|_{\partial\Omega_i} = 0$, where $J_{D,i} = D^M_{b,ij}\partial_j f$ and $J_{A,i} = \frac{m}{T}D^M_{b,ij}(v_j-u_j)f$. Consider a rectangular computational domain $\Omega = [a,b]\times[c,d]$ with the outer normal unit $\boldsymbol{n}$ equal to $[\pm1,0]^{\mathrm{T}}$ or $[0,\pm1]^{\mathrm{T}}$ on four sides, respectively. Here, we discuss only one edge in detail. For the remaining three edges, they can proceed in the same way. For example, if $\partial\Omega_i$ is the right side of Ω, then

$$D^M_{b,ij}\partial_j f = \sum_{j=0}^{d-1} D^M_{b,ij}\partial_j f = \left(\begin{bmatrix} D^M_{b,00} & D^M_{b,01}\\ D^M_{b,10} & D^M_{b,11}\end{bmatrix}\begin{bmatrix}\partial_0 f\\ \partial_1 f\end{bmatrix}\right)\cdot\boldsymbol{n}|_{\partial\Omega_i} = (\mathbf{D}\nabla f)\cdot\boldsymbol{n}|_{\partial\Omega_i}.$$

Thus, we obtain the boundary condition $(\mathbf{D}\nabla f)\cdot\boldsymbol{n} = 0$ on $\partial\Omega$. Similarly, we rewrite the boundary condition $J_{A,i}|_{\partial\Omega_i} = 0$ and get $\frac{m}{T}\big(\mathbf{D}(\boldsymbol{u}-\boldsymbol{v})f\big)\cdot\boldsymbol{n} = 0$.

As a summary, after linearization, we obtain the Rosenbluth–Fokker–Planck equation in the following convection-diffusion form. Let $\boldsymbol{v}\times t \in \Omega\times[0,t^{\mathrm{end}}] \subset \mathbb{R}^d\times\mathbb{R}_+$ be the simulation domain, Ω the spatial domain, and t^{end} the end time. The $\boldsymbol{n}$ denotes the unit outer normal on boundary $\partial\Omega$. Given parameters: inverse collision-time scale ε^{-1}, mass m, and temperature T, with prescribed coefficient matrix $\mathbf{D}$ and vector $\boldsymbol{u}$, we solve the unknown distribution function f, which satisfies

$$\partial_t f - \varepsilon^{-1}\nabla\cdot(\mathbf{D}\nabla f) + \varepsilon^{-1}\frac{m}{T}\nabla\cdot\big(\mathbf{D}(\boldsymbol{u}-\boldsymbol{v})f\big) = 0 \qquad \text{in } [0,t^{\mathrm{end}}]\times\Omega, \tag{17a}$$

$$f = f^0 \qquad \text{on } \{0\}\times\Omega, \tag{17b}$$

$$(\mathbf{D}\nabla f)\cdot\boldsymbol{n} = 0 \qquad \text{on } [0,t^{\mathrm{end}}]\times\partial\Omega, \tag{17c}$$

$$\frac{m}{T}\big(\mathbf{D}(\boldsymbol{u}-\boldsymbol{v})f\big)\cdot\boldsymbol{n} = 0 \qquad \text{on } [0,t^{\mathrm{end}}]\times\partial\Omega. \tag{17d}$$

To this end, let us construct a high order accurate, conservative, and positivity-preserving numerical scheme to solve (17).

3. The numerical scheme

In this section, we utilize the DG method with a semi-implicit time discretization to solve (17). The positivity of the distribution function is enforced by the approach in Section 1.3 without losing global conservation.

3.1. Semi-implicit DG discretization

Consider a rectangular computational domain $\Omega \subset \mathbb{R}^d$. Let $\mathcal{T}_h = \{E_i\}$ be a uniform partition of Ω by N square cells with element diameter h. Let Γ_h denote the set of interior faces. For each interior face $e \in \Gamma_h$ shared by cells E_{i^-} and E_{i^+}, with $i^- < i^+$, we define a unit normal vector $\boldsymbol{n}_e$ that points from E_{i^-} into E_{i^+}. For a boundary face, $e \subset \partial\Omega$, the normal vector $\boldsymbol{n}_e$ is taken to be the unit outward vector to $\partial\Omega$.

Let $\mathbb{P}^k(E_i)$ be the set of all polynomials of degree at most k on a cell E_i. For any $k \geq 1$, define the broken polynomial space

$$X_h = \{\chi_h \in L^2(\Omega) : \ \chi_h|_{E_i} \in \mathbb{P}^k(E_i), \ \forall E_i \in \mathcal{T}_h\}.$$

The average and jump of $\chi \in X_h$ on a boundary face coincide with its trace; and on interior faces they are defined by

$$\{\!\{\chi\}\!\}|_e = \frac{1}{2}\,\chi|_{E_{i^-}} + \frac{1}{2}\,\chi|_{E_{i^+}}\,, \quad [\![\chi]\!]\,|_e = \chi|_{E_{i^-}} - \chi|_{E_{i^+}}\,, \quad \forall e = \partial E_{i^-} \cap \partial E_{i^+}.$$

For the bases of $\mathbb{P}^k$ spaces, we choose $\hat{E} = [-\frac{1}{2}, \frac{1}{2}]^d$ as the reference element and use Legendre orthonormal polynomials to construct basis functions $\hat{\varphi}_j$ on $\hat{E}$. The bases on each cell $E_i \in \mathcal{T}_h$ are defined by $\varphi_{ij} = \hat{\varphi}_j \circ \boldsymbol{F}_i^{-1}$, where $\boldsymbol{F}_i : \hat{E} \to E_i$ is an invertible mapping from the reference element $\hat{E}$ to cell E_i. For more details on constructing hierarchical modal orthonormal bases, see [62]. For $\mathbb{P}^k$ scheme, we choose the tensor produce of $k+1$ point Gauss quadrature to evaluate numerical integral and denote the set of all quadrature points on a cell E by S_E.

DG forms. Assume the coefficient matrix $\mathbf{D}$ may vary in space, but it is symmetric positive definite and bounded below and above uniformly, e.g., there exist positive constants K_0 and K_1 such that, for all $\boldsymbol{\xi} \in \mathbb{R}^d$,

$$K_0 \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{\xi} \leq \boldsymbol{\xi}^{\mathrm{T}} \mathbf{D} \boldsymbol{\xi} \leq K_1 \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{\xi}. \tag{18}$$

Let vector $\boldsymbol{b} = \mathbf{D}(\boldsymbol{u} - \boldsymbol{v})$. Recall the boundary condition (17d). The Lax–Friedrichs flux of the convection term $-\nabla \cdot \big(\mathbf{D}(\boldsymbol{u} - \boldsymbol{v}) f\big) = -\nabla \cdot (\boldsymbol{b} f)$ is defined by

$$a_{\mathrm{conv}}(f, \chi) = \sum_{E \in \mathcal{T}_h} \int_E f\, \boldsymbol{b} \cdot \nabla \chi - \sum_{E \in \mathcal{T}_h} \int_{\partial E} \widehat{\boldsymbol{b} f \cdot \boldsymbol{n}_E} \chi, \tag{19a}$$

$$\text{where} \quad \widehat{\boldsymbol{b} f \cdot \boldsymbol{n}_E} = \frac{\boldsymbol{b} f^- + \boldsymbol{b} f^+}{2} \cdot \boldsymbol{n}_E - \frac{1}{2} \max_e |\boldsymbol{b} \cdot \boldsymbol{n}_e| (f^+ - f^-). \tag{19b}$$

Here, $\boldsymbol{n}_E$ is the outer normal of cell E. The f^- and f^+ denote the trace of f on the face ∂E that comes from the interior and exterior of E, respectively. We use the non-symmetric interior penalty DG (NIPG) method to discretize the diffusion term $-\nabla \cdot (\mathbf{D} \nabla f)$. Recall the boundary condition (17c). The associated bilinear form a_{diff} is

$$\begin{aligned} a_{\mathrm{diff}}(f, \chi) = & \sum_{E \in \mathcal{T}_h} \int_E (\mathbf{D} \nabla f) \cdot \nabla \chi - \sum_{e \in \Gamma_h} \int_e \{\!\{(\mathbf{D} \nabla f) \cdot \boldsymbol{n}_e\}\!\}\, [\![\chi]\!] \\ & + \sum_{e \in \Gamma_h} \int_e \{\!\{(\mathbf{D} \nabla \chi) \cdot \boldsymbol{n}_e\}\!\}\, [\![f]\!] + \frac{\sigma}{h} \sum_{e \in \Gamma_h} \int_e [\![f]\!]\, [\![\chi]\!]\,. \end{aligned}$$

The above NIPG form a_{diff} contains a penalty parameter σ. It should be noted that, for any $\sigma > 0$, the NIPG form of the diffusion term is coercive [63].

The fully discrete scheme. Let $0 = t^0 < t^1 < \cdots < t^{N_{\rm st}} = t^{\rm end}$ be a uniform partition of the time interval $[0, t^{\rm end}]$. For $1 \le n \le N_{\rm st}$, let $\tau = t^n - t^{n-1}$ be the $n^{\rm th}$ time-step size. The semi-implicit time marching scheme reads: given f^{n-1}, solve for f^n, such that

$$f^n - \tau\varepsilon^{-1}\boldsymbol{\nabla}\cdot(\mathbf{D}\boldsymbol{\nabla} f^n) = f^{n-1} - \tau\varepsilon^{-1}\frac{m}{T}\boldsymbol{\nabla}\cdot\big(\mathbf{D}(\boldsymbol{u}-\boldsymbol{v})f^{n-1}\big). \tag{20}$$

Let $\langle\cdot,\cdot\rangle$ denote the L^2 inner product. Associated with the time discretization (20), our semi-implicit DG scheme is defined as follows: given f_h^{n-1}, solve for f_h^n, such that for all $\chi_h \in X_h$,

$$\langle f_h^n, \chi_h\rangle + \tau\varepsilon^{-1}a_{\rm diff}(f_h^n, \chi_h) = \langle f_h^{n-1}, \chi_h\rangle + \tau\varepsilon^{-1}\frac{m}{T}a_{\rm conv}(f_h^{n-1}, \chi_h). \tag{21}$$

The initial f_h^0 is obtained by applying the L^2 projection on f^0 follows by the Zhang–Shu limiter.

It is easy to verify that the fully discrete scheme (21) conserves the global mass, that is, for any $1 \le n \le N_{\rm st}$, we have $\langle f_h^n, 1\rangle = \langle f^0, 1\rangle$. Due to $a_{\rm diff}(f_h^n, 1) = 0$ and $a_{\rm conv}(f_h^{n-1}, 1) = 0$, by choosing $\chi_h = 1$ in (21), we get $\langle f_h^n, 1\rangle = \langle f_h^{n-1}, 1\rangle$, which implies $\langle f_h^n, 1\rangle = \langle f_h^0, 1\rangle$. Since the Zhang–Shu limiter preserves conservation, the L^2 projection gives $\langle f_h^0, 1\rangle = \langle f^0, 1\rangle$.

3.2. A high order accurate constraint optimization-based postprocessing approach

We describe the approach in Section 1.3 in more detail.

Let $S_h = \cup_i S_{E_i}$ denote the set of all quadrature points. The following two-stage limiting strategy can be used to enforce the positivity of f_h at any quadrature point $\boldsymbol{v}_q \in S_h$ without losing global conservation.

- Stage 1. Use a cell average limiter to enforce the average of the DG polynomial on each cell to be positive, if there exists any cell average out of the bounds.
- Stage 2. Use the Zhang–Shu limiter to eliminate undershoots of the DG polynomial if there exist negative values at any quadrature points in S_h. Select a small number $\epsilon > 0$ as the numerical tolerance of the admissible set. Enforce positivity of the distribution function by

$$\hat{f}_E(\boldsymbol{v}) = \theta_f\big(f_E(\boldsymbol{v}) - \overline{f}_E\big) + \overline{f}_E, \quad \text{where } \theta_f = \min\left\{1,\ \frac{\overline{f}_E - \epsilon}{\overline{f}_E - \min\limits_{\boldsymbol{v}_q\in S_E} f_E(\boldsymbol{v}_q)}\right\}.$$

  In the above, $\overline{f}_E$ denotes the cell average of f_E in the cell E. Notice that $\hat{f}_E$ and f_E have the same cell average, and $\hat{f}_E(\boldsymbol{v}) = f_E(\boldsymbol{v})$ if $\min\limits_{\boldsymbol{v}_q\in S_E} f_E(\boldsymbol{v}_q) \ge \epsilon$.

To this end, let us construct a conservative high-order accurate cell average limiter.

An optimization-based cell average limiter. In the context of the DG scheme, a high order accurate cell average limiter that preserves conservation and bounds can be formulated as seeking a piecewise constant polynomial x_h solving

$$\min_{x_h} \|x_h - \overline{f_h}\|_{L^2}^2 \quad \text{subject to} \quad \int_\Omega x_h = \int_\Omega f_h \ \text{ and } \ m \le x_h \le M. \tag{22}$$

When only seeking to preserve positivity, we treat the upper bound $M = +\infty$. In (22) the $\overline{f_h}$ is a piecewise constant polynomial whose value in each cell E equals the cell average of f_h, namely $\overline{f_h}|_E = \frac{1}{|E|}\int_E f_h$. We denote the solution of the minimization problem (22) by $\overline{w_h}$. Then, the postprocessed polynomial

$$f_h^{\rm lim} = (f_h - \overline{f_h}) + \overline{w_h} \tag{23}$$

preserves conservation and bounds of cell average. Let us show that the modification (23) is of the same approximation order to the exact solution f.

Construct a polynomial $\mathcal{P}_h f$ by taking L^2 projection of the exact solution f then applying the Zhang–Shu limiter. Thus, the polynomial $\mathcal{P}_h f$ is an approximation of f that satisfies $\|\mathcal{P}_h f - f\|_{L^2} \le Ch^{k+1}$ and $m \le \mathcal{P}_h f \le M$. Let $\overline{\mathcal{P}_h f}$ denote a piecewise constant polynomial whose value in each cell is equal to the cell average of $\mathcal{P}_h f$. It is straightforward to verify $m \le \overline{\mathcal{P}_h f} \le M$. Since the L^2 projection, the Zhang–Shu limiter, and the numerical scheme preserve conservation, we have

$$\int_\Omega \overline{\mathcal{P}_h f} = \sum_{E\in\mathcal{T}_h} \int_E \Big(\frac{1}{|E|}\int_E \mathcal{P}_h f\Big) = \int_\Omega \mathcal{P}_h f = \int_\Omega f = \int_\Omega f_h.$$

By (23), the postprocessed polynomial satisfies $f_h^{\lim} - f_h = \overline{w_h} - \overline{f_h}$. Use triangle inequality and (22), we have

$$\|f_h^{\lim} - f\|_{L^2} \le \|\overline{w_h} - \overline{f_h}\|_{L^2} + \|f_h - f\|_{L^2} \le \|\overline{\mathcal{P}_h f} - \overline{f_h}\|_{L^2} + \|f_h - f\|_{L^2}.$$

Thus, if the numerical solution f_h is optimal in the L^2 norm, then we only need to show $\|\overline{\mathcal{P}_h f} - \overline{f_h}\|_{L^2} \le Ch^{k+1}$. Actually, this is guaranteed by Cauchy–Schwarz's inequality. We have

$$\begin{aligned}\|\overline{\mathcal{P}_h f} - \overline{f_h}\|_{L^2} &= \sum_{E\in\mathcal{T}_h} \|\frac{1}{|E|}\int_E (\mathcal{P}_h f - f_h)\|_{L^2(E)} \\ &\le \sum_{E\in\mathcal{T}_h} \frac{1}{|E|}\|\mathcal{P}_h f - f_h\|_{L^2(E)}\|1\|^2_{L^2(E)} = \|\mathcal{P}_h f - f_h\|_{L^2}.\end{aligned}$$

We conclude the proof since $\|\mathcal{P}_h f - f_h\|_{L^2} \le \|\mathcal{P}_h f - f\|_{L^2} + \|f_h - f\|_{L^2} \le Ch^{k+1}$.

Matrix-vector form. For convenience of solving the minimization problem (22), let us rewrite it into an equivalent non-constraint form. Recall that an indicator function ι_Λ of a set Λ satisfies: $\iota_\Lambda(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \Lambda$ and $\iota_\Lambda(\boldsymbol{x}) = +\infty$ for $\boldsymbol{x} \notin \Lambda$. We define a matrix $\mathbf{A} = [1, 1, \cdots, 1] \in \mathbb{R}^{1\times N}$, where N is the total number of mesh cells. A vector $\boldsymbol{w} \in \mathbb{R}^N$ is introduced to store the cell average of the DG polynomial f_h, that is, the i^{th} entry of $\boldsymbol{w}$ equals $\overline{f_h}|_{E_i}$. We have the following lemma.

Lemma 1. *Define positive constants $\alpha = 2|E|$ and $b = \mathbf{A}\boldsymbol{w}$. Associated with the conservation constraint and the bound-preserving constraint, define sets*

$$\Lambda_1 = \{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} = b\} \quad \textit{and} \quad \Lambda_2 = \{\boldsymbol{x} : m \le x_i \le M,\ \forall i = 0, \cdots, N-1\}.$$

The matrix-vector form of the optimization model (22) *becomes: find a vector $\boldsymbol{x} \in \mathbb{R}^N$ such that it solves*

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{x}) + \iota_{\Lambda_2}(\boldsymbol{x}). \tag{24}$$

Proof. First, notice that the i^{th} entry of the vector $\boldsymbol{w}$ is equal to the cell average of f_h on cell E_i, we have

$$\|x_h - \overline{f_h}\|_{L^2}^2 = \sum_{E\in\mathcal{T}_h}\int_E |x_h - \overline{f_h}|^2 = \sum_{E\in\mathcal{T}_h}\Big|x_h|_E - \overline{f_h}|_E\Big|^2 \int_E 1 = |E|\sum_i |x_i - w_i|^2 = \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2.$$

Thus, seeking a piecewise constant polynomial x_h to minimize the $\|x_h - \overline{f_h}\|_{L^2}^2$ is equivalent to seeking a vector $\boldsymbol{x} \in \mathbb{R}^N$ to minimize $\frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2$.

Next, regarding to the conservation constraint in the minimization problem (22), notice x_h is piecewise constant, we have

$$\int_\Omega x_h = \int_\Omega f_h \quad\Leftrightarrow\quad \sum_{E\in\mathcal{T}_h}\int_E x_h = \sum_{E\in\mathcal{T}_h}\int_E f_h \quad\Leftrightarrow\quad \sum_{E\in\mathcal{T}_h} x_h|_E \int_E 1 = \sum_{E\in\mathcal{T}_h} |E|\Big(\frac{1}{|E|}\int_E f_h\Big).$$

Again, by definition of the vector $\boldsymbol{w}$, we have

$$\sum_{E\in\mathcal{T}_h}|E|(x_h|_E)=\sum_{E\in\mathcal{T}_h}|E|(\overline{f_h}|_E)\quad\Leftrightarrow\quad|E|\sum_i x_i=|E|\sum_i w_i.$$

Recall the matrix $\mathbf{A}=[1,1,\cdots,1]^{\mathrm{T}}$ and the scalar $b=\mathbf{A}\boldsymbol{w}$. We get $\mathbf{A}\boldsymbol{x}=\mathbf{A}\boldsymbol{w}=b$. Thus, finding a piecewise constant polynomial x_h satisfies the conservation constraint in (22) is equivalent to finding a vector $\boldsymbol{x}\in\Lambda_1$.

Finally, by definition of the set Λ_2, it is straightforward to verify that the bound-preserving constraint in (22) is equivalent to seeking a vector $\boldsymbol{x}\in\Lambda_2$. □

The accuracy of (24) is easily understood. Let $\boldsymbol{x}^*$ represent the minimizer of (24). We consider the ℓ^2 distance between $\boldsymbol{x}^*$ and $\overline{\boldsymbol{f}}$, which is a vector in $\mathbb{R}^N$ that stores the cell averages of the exact solution, i.e., the i^{th} entry of $\overline{\boldsymbol{f}}$ equals $\frac{1}{|E_i|}\int_{E_i}f$. By triangle inequality, we have $\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2\le\|\overline{\boldsymbol{f}}-\boldsymbol{w}\|_2+\|\boldsymbol{w}-\boldsymbol{x}^*\|_2\le 2\|\overline{\boldsymbol{f}}-\boldsymbol{w}\|_2$. But we can actually make further improvements. The sets Λ_1 and Λ_2 are convex and closed give $\Lambda_1\cap\Lambda_2$ is a convex closed set. Thus, $\overline{\boldsymbol{f}}$ and $\boldsymbol{x}^*$ belongs to $\Lambda_1\cap\Lambda_2$ implies $\eta\overline{\boldsymbol{f}}+(1-\eta)\boldsymbol{x}^*\in\Lambda_1\cap\Lambda_2$, for any $\eta\in[0,1]$. Let us define

$$\begin{aligned}\phi(\eta)&=\|\boldsymbol{w}-(\eta\overline{\boldsymbol{f}}+(1-\eta)\boldsymbol{x}^*)\|_2^2\\&=\eta^2\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2^2-2\eta(\boldsymbol{w}-\boldsymbol{x}^*)^{\mathrm{T}}(\overline{\boldsymbol{f}}-\boldsymbol{x}^*)+\|\boldsymbol{w}-\boldsymbol{x}^*\|_2^2.\end{aligned}$$

If $\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2=0$, then $\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2\le\|\overline{\boldsymbol{f}}-\boldsymbol{w}\|_2$ automatically holds. Otherwise, it is obvious that $\phi(\eta)$ is a quadratic function with respect to η. From (24), we know $\boldsymbol{x}^*$ minimize $\|\boldsymbol{w}-\boldsymbol{x}\|_2^2$ for all $\boldsymbol{x}\in\Lambda_1\cap\Lambda_2$. Thus, $\phi(\eta)$ achieves its minimum at $\eta=0$, which gives

$$\frac{(\boldsymbol{w}-\boldsymbol{x}^*)^{\mathrm{T}}(\overline{\boldsymbol{f}}-\boldsymbol{x}^*)}{\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2^2}\le 0\quad\Rightarrow\quad(\boldsymbol{x}^*-\boldsymbol{w})^{\mathrm{T}}(\overline{\boldsymbol{f}}-\boldsymbol{x}^*)\ge 0.$$

Therefore, we have

$$\|\overline{\boldsymbol{f}}-\boldsymbol{w}\|_2^2=\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*+\boldsymbol{x}^*-\boldsymbol{w}\|_2^2=\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2^2+2(\boldsymbol{x}^*-\boldsymbol{w})^{\mathrm{T}}(\overline{\boldsymbol{f}}-\boldsymbol{x}^*)+\|\boldsymbol{x}^*-\boldsymbol{w}\|_2^2\ge\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2^2.$$

The $\|\overline{\boldsymbol{f}}-\boldsymbol{x}^*\|_2\le\|\overline{\boldsymbol{f}}-\boldsymbol{w}\|_2$ implies that at each time step after applying the cell average limiter (24), the modified cell average is not worse in the sense of the ℓ^2 distance to the cell average of the exact solution.

3.3. The Douglas–Rachford splitting method for enforcing positivity of cell averages

Splitting algorithms naturally arise and are popular when solving the minimization problem of the form

$$\min_{\boldsymbol{x}}g(\boldsymbol{x})+h(\boldsymbol{x}),\tag{25}$$

where functions g and h are convex closed proper functions, with computable proximal operators.

Let $G=\partial g$ and $H=\partial h$ be the subdifferentials of g and h. Then, a sufficient and necessary condition for $\boldsymbol{x}^*$ being a minimizer of (25) is $\mathbf{0}\in G(\boldsymbol{x}^*)+H(\boldsymbol{x}^*)$. Recall that $\|\cdot\|_2$ denotes the vector 2-norm. The resolvents $\mathtt{prox}_g^\gamma=(\mathrm{I}+\gamma G)^{-1}$ and $\mathtt{prox}_h^\gamma=(\mathrm{I}+\gamma H)^{-1}$ are also called proximal operators, as $\mathtt{prox}_g^\gamma$ maps $\boldsymbol{x}$ to $\operatorname{argmin}_z\gamma g(\boldsymbol{z})+\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{x}\|_2^2$ and $\mathtt{prox}_h^\gamma$ is defined similarly. The reflection operators are defined as $\mathrm{R}_g^\gamma=2\,\mathtt{prox}_g^\gamma-\mathrm{I}$ and $\mathrm{R}_h^\gamma=2\,\mathtt{prox}_h^\gamma-\mathrm{I}$, where I is the identity operator.

The relaxed Douglas–Rachford splitting method for solving the minimization problem (25) can be written as:

$$\begin{cases}\boldsymbol{y}^{k+1}=\lambda\dfrac{\mathrm{R}_g^\gamma\mathrm{R}_h^\gamma+\mathrm{I}}{2}\boldsymbol{y}^k+(1-\lambda)\boldsymbol{y}^k,\\ \boldsymbol{x}^{k+1}=\mathtt{prox}_h^\gamma(\boldsymbol{y}^{k+1}).\end{cases}\tag{26}$$

The vector $\boldsymbol{y}$ is an auxiliary variable, λ belongs to $(0, 2]$ is a parameter, and $\gamma > 0$ is step size. We get the classical Douglas–Rachford splitting when take $\lambda = 1$ in (26). In the limiting case $\lambda = 2$, we obtain the Peaceman–Rachford splitting. For simplicity, we refer to the relaxed Douglas–Rachford splitting method with $\lambda \in (0, 2)$ as the Douglas–Rachford splitting. For two convex closed proper functions $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$, the (26) converges for any positive step size γ and any fixed $\lambda \in (0, 2)$, see [56]. If one function is strongly convex, then $\lambda = 2$ also leads to converges. Using the definition of reflection operators, the (26) can be expressed as follows:

$$\begin{cases} \boldsymbol{y}^{k+1} = \lambda\, \mathtt{prox}_g^{\gamma}(2\boldsymbol{x}^k - \boldsymbol{y}^k) + \boldsymbol{y}^k - \lambda \boldsymbol{x}^k, \\ \boldsymbol{x}^{k+1} = \mathtt{prox}_h^{\gamma}(\boldsymbol{y}^{k+1}). \end{cases} \tag{27}$$

In order to construct a bound-preserving cell average limiter, let us split the objective function in (24) into two parts, where

$$g(\boldsymbol{x}) = \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{x}) \quad \text{and} \quad h(\boldsymbol{x}) = \iota_{\Lambda_2}(\boldsymbol{x}).$$

The two sets Λ_1 and Λ_2 are convex and closed, thus both g and h are convex closed proper functions. Moreover, the function g is strongly convex thus (27) converges to the unique minimizer. After applying (27) solving the minimization to machine precision, the positivity constraint is strictly satisfied and the conservation constraint is enforced up to the round-off error. It is convenient to employ the norm $\|\cdot\|_{2h} = h^{d/2}\|\cdot\|_2$ to measure the conservation error. To this end, let us list the subdifferentials and the associated resolvents as follows:

- The subdifferential of function g is

$$\partial g(\boldsymbol{x}) = \alpha(\boldsymbol{x} - \boldsymbol{w}) + \mathcal{R}(\mathbf{A}^{\mathrm{T}}),$$

  where $\mathcal{R}(\mathbf{A}^{\mathrm{T}})$ denotes the range of the matrix $\mathbf{A}^{\mathrm{T}}$.

- The subdifferential of function h is

$$[\partial h(\boldsymbol{x})]_i = \begin{cases} [0, +\infty], & \text{if } x_i = M, \\ 0, & \text{if } x_i \in (m, M), \\ [-\infty, 0], & \text{if } x_i = m. \end{cases}$$

- For the function $g(\boldsymbol{x}) = \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{w}\|_2^2 + \iota_{\Lambda_1}(\boldsymbol{x})$, the associated resolvent is

$$\mathtt{prox}_g^{\gamma}(\boldsymbol{x}) = \frac{1}{\gamma\alpha + 1}\left(\mathbf{A}^{+}(b - \mathbf{A}\boldsymbol{x}) + \boldsymbol{x}\right) + \frac{\gamma\alpha}{\gamma\alpha + 1}\boldsymbol{w}, \tag{28}$$

  where $\mathbf{A}^{+} = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}$ denotes the pseudo inverse of the matrix $\mathbf{A}$.

- For the function $h(\boldsymbol{x}) = \iota_{\Lambda_2}(\boldsymbol{x})$, the associated resolvent is $\mathtt{prox}_h^{\gamma}(\boldsymbol{x}) = \mathrm{S}(\boldsymbol{x})$, where S is a cut-off operator defined by

$$[\mathrm{S}(\boldsymbol{x})]_i = \min\left(\max\left(x_i, m\right), M\right), \quad \forall i = 0, \cdots, N-1. \tag{29}$$

Define the parameter $c = \frac{1}{\gamma\alpha+1}$, which gives $\frac{\gamma\alpha}{\gamma\alpha+1} = 1 - c$. Using the expressions of resolvents in (28) and (29), we obtain the Douglas–Rachford splitting method for solving the minimization problem (24) in matrix-vector form:

$$\begin{cases} \boldsymbol{z}^k = 2\boldsymbol{x}^k - \boldsymbol{y}^k, \\ \boldsymbol{y}^{k+1} = \lambda c\left(\mathbf{A}^{+}(b - \mathbf{A}\boldsymbol{z}^k) + \boldsymbol{z}^k\right) + \lambda(1 - c)\boldsymbol{w} + \boldsymbol{y}^k - \lambda\boldsymbol{x}^k, \\ \boldsymbol{x}^{k+1} = \mathrm{S}(\boldsymbol{y}^{k+1}). \end{cases} \tag{30}$$

Notice that the pseudo-inverse $\mathbf{A}^{+}$ in (30) can be precomputed before code implementation, namely, the matrix $\mathbf{A} = [1, 1, \cdots, 1]^{\mathrm{T}}$ gives $\mathbf{A}^{+} = \frac{1}{N}\mathbf{1}$, where $\mathbf{1}$ is a constant one vector of size N.

Implementation. To this end, let us briefly summarize our optimization solver. After obtaining the DG polynomial f_h from solving (21), calculate the cell averages to generate vector $\boldsymbol{w}$, where the i^{th} entry of $\boldsymbol{w}$ equals $\overline{f_h}|_{E_i} = \frac{1}{|E_i|}\int_{E_i} f_h$, then our cell average limiter can be implemented as follows.

Algorithm DR. To start the Douglas–Rachford iteration, set $\boldsymbol{y}^0 = \boldsymbol{w}$, $\boldsymbol{x}^0 = \mathrm{S}(\boldsymbol{w})$, and $k = 0$. Compute the parameters c and λ using the formula in Remark 2. And select a small ϵ_{tol} for numerical tolerance of the conservation error.

Step 1. Compute the intermediate variable $\boldsymbol{z}^k = 2\boldsymbol{x}^k - \boldsymbol{y}^k$.

Step 2. Compute the auxiliary variable $\boldsymbol{y}^{k+1} = \lambda c\big(\boldsymbol{z}^k + \frac{1}{N}(b - \sum_i z_i^k)\mathbf{1}\big) + \lambda(1-c)\boldsymbol{w} + \boldsymbol{y}^k - \lambda\boldsymbol{x}^k$.

Step 3. Compute $\boldsymbol{x}^{k+1} = \mathrm{S}(\boldsymbol{y}^{k+1})$.

Step 4. If the stopping criterion $\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\|_{2h} < \epsilon_{\text{tol}}$ is satisfied, then terminate and output $\boldsymbol{x}^* = \boldsymbol{x}^{k+1}$, otherwise set $k \leftarrow k+1$ and go to Step 1.

In the above algorithm, $2\boldsymbol{x}^k$ is a trivial process, which is equivalent to a shift left by one bit; the $\lambda(1-c)\boldsymbol{w}$ remains unchanged during iteration; and each entry of $\boldsymbol{z}^k + \frac{1}{N}(b - \sum_i z_i^k)\mathbf{1}$ can be computed by $z_i^k + \frac{1}{N}(b - \sum_i z_i^k)$, therefore if only counting the number of multiplication operations and taking the maximum, the computational complexity of each iteration is $3N$. To preserve positivity, the upper bound M is set to $+\infty$, we only need to take the maximum in operator S.

Remark 2. *The analysis in [52] proves the asymptotic linear convergence and suggests a simple choice of almost optimal parameters c and λ in* (30)*. Let $\hat{r}$ be the number of bad cells $\overline{f_h} \notin [m, M]$ and let $\hat{\theta} = \cos^{-1}\sqrt{\frac{\hat{r}}{N}}$, then we have:*

$$\begin{cases} c = \frac{1}{2}, \lambda = \frac{4}{2-\cos(2\hat{\theta})}, & \text{if } \hat{\theta} \in (\frac{3}{8}\pi, \frac{1}{2}\pi], \\ c = \frac{1}{(\cos\hat{\theta}+\sin\hat{\theta})^2}, \lambda = \frac{2}{1+\frac{1}{1+\cot\hat{\theta}}-\frac{1}{(\cos\hat{\theta}+\sin\hat{\theta})^2}}, & \text{if } \hat{\theta} \in (\frac{1}{4}\pi, \frac{3}{8}\pi], \\ c = \frac{1}{(\cos\hat{\theta}+\sin\hat{\theta})^2}, \lambda = 2, & \text{if } \hat{\theta} \in (0, \frac{1}{4}\pi]. \end{cases}$$

Remark 3. *The scaling factor $h^{d/2}$ in $\|\cdot\|_{2h}$ is from the discrete approximation of the L^2 norm. Notice that our optimization-based cell average limiter postprocesses the piecewise-constant DG polynomial. Given two piecewise constant DG polynomials x_h and f_h, we consider using $\|x_h - f_h\|_{L^2}$ to measure the error between x_h and f_h. In d-dimensional space, we have*

$$\|x_h - f_h\|_{L^2}^2 = \sum_{E\in\mathcal{T}_h}\int_E |x_h - f_h|^2 = \sum_{E\in\mathcal{T}_h} |x_h|_E - f_h|_E|^2 \int_E 1 = |E|\sum_i |x_i - f_i|^2$$

Here, the i-th entry of the vector $\boldsymbol{x}$ is the cell average of the piecewise constant DG polynomial x_h on cell E_i, namely $x_i = \frac{1}{|E_i|}\int_{E_i} x_h|_{E_i} = x_h|_{E_i}$ and similar to the vector $\boldsymbol{f}$. Therefore, we utilize $h^{d/2}\|\boldsymbol{x} - \boldsymbol{f}\|$ to measure $\|x_h - f_h\|_{L^2}$.

Remark 4. *For efficiency purposes, an ad hoc technique is to first detect trouble cells, i.e., mark a part of the computational domain as good cells, where the cell averages will not be modified. Define K be a strict subset of $\{0, 1, \cdots, N-1\}$ that contains all trouble cells and also some good cells, as follows*

$$K = \{i : \ \text{either } \ \overline{f_h} \notin [m, M] \ \ \text{or } \ \overline{f_h} \geq 10^{-8}\}.$$

We only do the postprocessing (22) *on a subset of the domain Ω, where the cells E_i have indexes $i \in K$.*

4. Numerical experiments

In this section, we verify our numerical scheme through accuracy tests. The simulation of a challenging reduced 2D model shows that our proposed method enjoys mass conservation and positivity-preserving properties. To preserve positivity, in all simulations, we set the upper bound $M = +\infty$ and the lower bound $m = 10^{-13}$ in (22) and take the tolerance $\epsilon_{\text{tol}} = 10^{-13}$ in Algorithm DR. In addition, we employ the bad cell detection defined in Remark 4 for all physical tests.

4.1. Accuracy test

We utilize the manufactured solution method to verify the accuracy of our algorithm. Let $\Omega = [-10, 10]^2$ be the computational domain. Set the simulation to start at time $t = 1$ and terminate at time $t^{\text{end}} = 20$. Consider a simple two-dimensional linear Fokker–Planck equation, associated with $\mathbf{D} = \mathbf{I}$, $\boldsymbol{u} = \mathbf{0}$ and all other coefficients equals to one in (17a), as follows

$$\partial_t f = \Delta f + \boldsymbol{\nabla} \cdot (f \, \boldsymbol{\nabla} \mathcal{W}) \quad \text{in} \quad [1, t^{\text{end}}] \times \Omega, \tag{31}$$

where $f(t, \boldsymbol{v})$ denotes the unknown and $\mathcal{W} = \frac{1}{2}\|\boldsymbol{v}\|_2^2$. The initial and boundary conditions are imposed by the following exact solution

$$f(t, \boldsymbol{v}) = \frac{1}{2\pi(1 - e^{-2t})} \, e^{-\frac{1}{2(1-e^{-2t})}\|\boldsymbol{v}\|_2^2}. \tag{32}$$

We refer to [33, section 5.2] for a derivation of the steady state solution (32) of (31).

We employ NIPG methods in $\mathbb{P}^2$ and $\mathbb{P}^3$ spaces with penalty parameter $\sigma = 1$. Let $\texttt{err}_{\Delta x}$ denote the error on a grid associated with the mesh resolution Δx. To be specific, the discrete L_h^2 and L_h^∞ errors are defined by

$$\begin{aligned} L_h^2 \text{ error:} \qquad & \|f_h^n - f(t^n)\|_{L_h^2}^2 = \Delta x^2 \sum_i \sum_v \omega_v \Big| \sum_j f_{ij}^n \, \varphi_{ij}(\boldsymbol{q}_v) - f(t^n, \boldsymbol{q}_v) \Big|^2, \\ L_h^\infty \text{ error:} \qquad & \|f_h^n - f(t^n)\|_{L_h^\infty} = \max_i \max_v \Big| \sum_j f_{ij}^n \, \varphi_{ij}(\boldsymbol{q}_v) - f(t^n, \boldsymbol{q}_v) \Big|, \end{aligned}$$

where ω_v and $\boldsymbol{q}_v$ are quadrature weights and points. Then, the rate is defined by $\ln(\texttt{err}_{\Delta x}/\texttt{err}_{\Delta x/2})/\ln(2)$. Notice, in even-order spaces, the NIPG methods are sub-optimal; and in odd-order spaces, the NIPG methods are optimal [63]. Specifically, the convergence rates are second order for $\mathbb{P}^2$ scheme and fourth order for $\mathbb{P}^3$ scheme. We obtain expected convergence rates, see Table 1. The Figure 2 shows snapshots of the distribution function at the simulation final time $t^{\text{end}} = 20$, which is large enough to approximate a steady-state solution. The cell average limiter is triggered. Our scheme preserve the positivity.

k	τ	Δx	$\|f_h^{N_{\text{st}}} - f(t^{\text{end}})\|_{L_h^2}$	rate	$\|f_h^{N_{\text{st}}} - f(t^{\text{end}})\|_{L_h^\infty}$	rate
2	$2^2 \cdot 10^{-2}$	$20 \cdot 2^{-6}$	$1.932 \cdot 10^{-3}$	—	$1.666 \cdot 10^{-3}$	—
	$2^0 \cdot 10^{-2}$	$20 \cdot 2^{-7}$	$5.186 \cdot 10^{-4}$	1.898	$4.626 \cdot 10^{-4}$	1.849
	$2^{-2} \cdot 10^{-2}$	$20 \cdot 2^{-8}$	$1.330 \cdot 10^{-4}$	1.963	$1.194 \cdot 10^{-4}$	1.954
3	$2^4 \cdot 10^{-2}$	$20 \cdot 2^{-6}$	$1.821 \cdot 10^{-5}$	—	$1.877 \cdot 10^{-5}$	—
	$2^0 \cdot 10^{-2}$	$20 \cdot 2^{-7}$	$1.188 \cdot 10^{-6}$	3.938	$1.264 \cdot 10^{-6}$	3.893
	$2^{-4} \cdot 10^{-2}$	$20 \cdot 2^{-8}$	$7.555 \cdot 10^{-8}$	3.975	$8.079 \cdot 10^{-8}$	3.968

Table 1: Errors and convergence rates for $\mathbb{P}^2$ and $\mathbb{P}^3$ schemes.

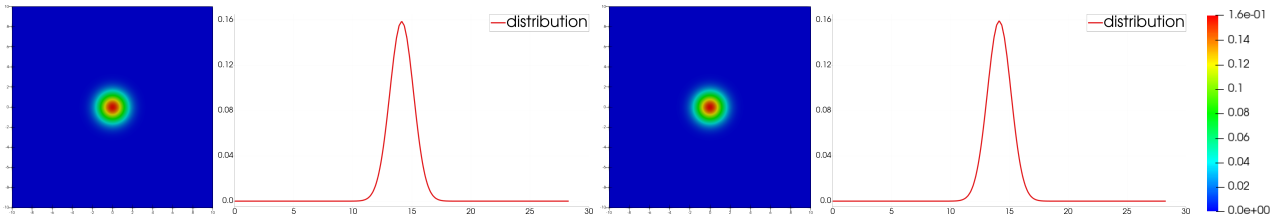

Figure 2: Plot the distribution function f and its value along the diagonal $\{x = y\}$ at time $t^{\text{end}} = 20$. From left to right: simulation results associated with $\mathbb{P}^2$ and $\mathbb{P}^3$ scheme of mesh resolution $\Delta x = 1/128$.

4.2. Anisotropic initial condition with non-identity diffusion matrix

Let the computational domain $\Omega = [-10, 10]^2$ and select the simulation end time $t^{\text{end}} = 2$. Similar to the numerical test in [64, Example 5.3], we consider the following Fokker–Planck equation

$$\partial_t f = \nabla \cdot (\mathbf{D}\nabla f) - \nabla \cdot (\boldsymbol{b} f). \tag{33}$$

Let δ_{ij} denote the Kronecker delta, namely if $i = j$ then $\delta_{ij} = 1$, otherwise it equals to 0. The matrix $\mathbf{D}$ and vector $\boldsymbol{b}$ in (33) are given by ($d = 2$ is the dimension):

$$\boldsymbol{b} = -(d-1)\boldsymbol{v}, \quad \mathbf{D}_{ij} = \delta_{ij}(|\boldsymbol{v}|^2 + 2E) - v_i v_j - \Sigma_{ij}(t), \tag{34a}$$

$$\Sigma_{ij}(t) = \Sigma_{ij}(\infty) - (\Sigma_{ij}(\infty) - \Sigma_{ij}(0))\exp(-4dt), \tag{34b}$$

$$\Sigma_{ij}(0) = \int v_i v_j f^0(\boldsymbol{v})\mathrm{d}\boldsymbol{v}, \quad \Sigma_{ij}(\infty) = \frac{2E}{d}\delta_{ij}, \quad 2E = \mathrm{tr}(\Sigma(0)). \tag{34c}$$

For this equation, one can consider the initial condition f^0 as the normal distribution with mean 0 and covariance matrix $\Sigma_{ij}(0) = \sigma_i \delta_{ij}$, where $\sigma_1 = 1.8$ and $\sigma_2 = 0.2$. Then the covariance of the solution $\Sigma_{ij}(t) = \int_{\mathbb{R}^2} v_i v_j f(t, \boldsymbol{v})\,\mathrm{d}\boldsymbol{v}$ at a later time is given by the formula (34b).

We utilize NIPG methods in $\mathbb{P}^2$ and $\mathbb{P}^3$ spaces with penalty parameter $\sigma = 1$. Since the true solution is unknown, we cannot plot the L^2 error. However, the exact form $\Sigma_{ij}(t)$ for the second moments of the true solution allows us to compare it with the second moments of the numerical solution. We use Σ_{ij}^h to denote the second moments of a numerical solution associated to mesh resolution Δx. Table 2 shows the convergence rate and Figure 3 shows the covariance trajectories. The cell average limiter is triggered and our scheme preserves the positivity.

k	τ	Δx	$\lvert\Sigma_{11}^h(t^{\text{end}}) - \Sigma_{11}(\infty)\rvert$	rate	$\lvert\Sigma_{22}^h(t^{\text{end}}) - \Sigma_{22}(\infty)\rvert$	rate
2	$2^2 \cdot 10^{-4}$	$20 \cdot 2^{-6}$	$1.529 \cdot 10^{-2}$	—	$1.529 \cdot 10^{-2}$	—
	$2^0 \cdot 10^{-4}$	$20 \cdot 2^{-7}$	$3.874 \cdot 10^{-3}$	1.981	$3.874 \cdot 10^{-3}$	1.981
	$2^{-2} \cdot 10^{-4}$	$20 \cdot 2^{-8}$	$9.802 \cdot 10^{-4}$	1.983	$9.800 \cdot 10^{-4}$	1.983
3	$2^4 \cdot 10^{-3}$	$20 \cdot 2^{-5}$	$3.702 \cdot 10^{-4}$	—	$3.717 \cdot 10^{-4}$	—
	$2^0 \cdot 10^{-3}$	$20 \cdot 2^{-6}$	$2.412 \cdot 10^{-5}$	3.940	$2.437 \cdot 10^{-5}$	3.931
	$2^{-4} \cdot 10^{-3}$	$20 \cdot 2^{-7}$	$1.466 \cdot 10^{-6}$	4.041	$1.651 \cdot 10^{-6}$	3.884

Table 2: Non-identity diffusion test. Errors and convergence rates for $\mathbb{P}^2$ and $\mathbb{P}^3$ schemes.

4.3. Reduced 2D RFP model with analytical expression of the diffusion tensor

Let us consider a 2D reduced dimensional RFP model so that the unknown distribution function f depends on independent variables $t \in [0, t^{\text{end}}] \subset \mathbb{R}_+$ and $\boldsymbol{v} \in \Omega \subset \mathbb{R}^2$. In this example, the computational domain Ω is selected to be $[-10, 10]^2$ with the simulation end time $t^{\text{end}} = 1$. We choose mass $m_b = 2000$

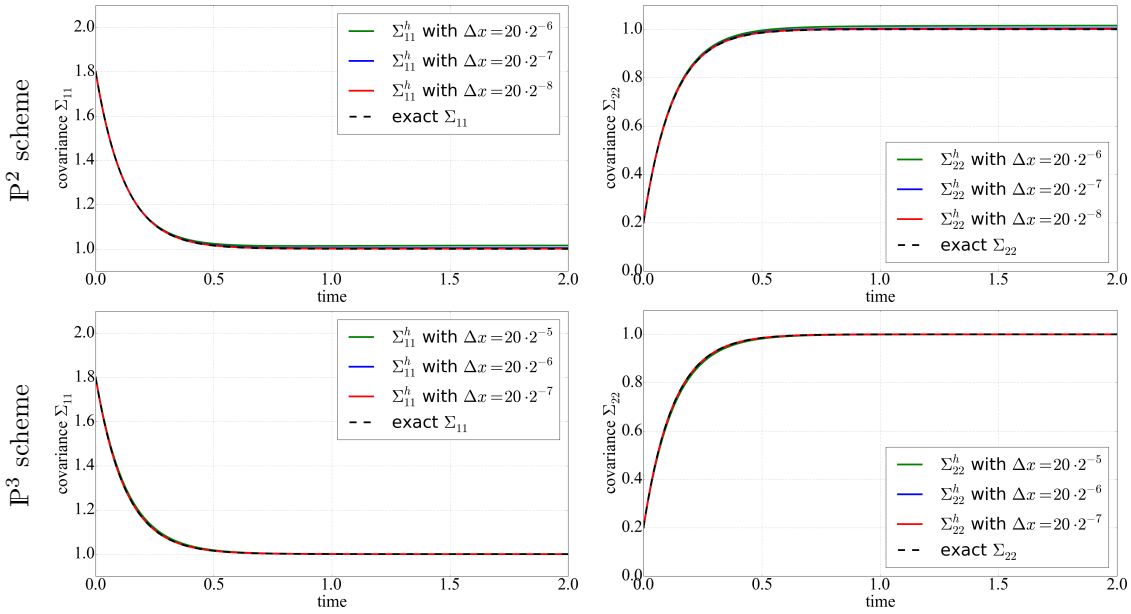

Figure 3: Non-identity diffusion test. Snapshot of covariances trajectories. From top to bottom: simulation results associated with $\mathbb{P}^2$ and $\mathbb{P}^3$ schemes. The black dashed line denotes the analytic solution.

to model an electron colliding against a background of heavy ions, density $n_b = 1$, temperature $T = 1$, and velocity $\boldsymbol{u} = [u_0, u_1]^{\mathrm{T}} = [2.5, 0]^{\mathrm{T}}$. See Figure 4 for displaying the coefficients $D_{b,ij}^M$ in (13) on a 128-by-128 grid.

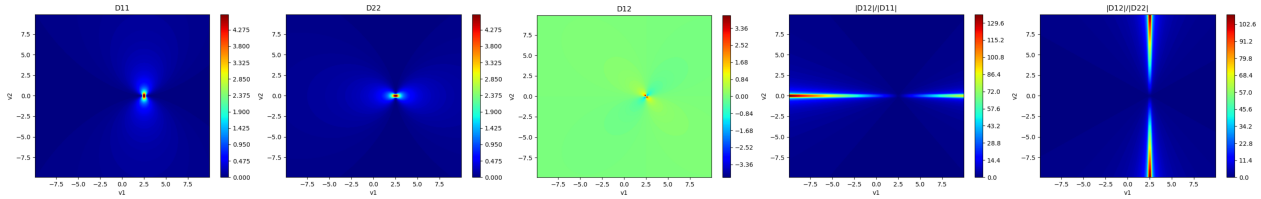

Figure 4: Plotting the entries in the coefficient matrix $\mathbf{D}$ and their ratios on a 128-by-128 grid. The first three sub-figures from left to right: the $D_{b,00}^M$, $D_{b,11}^M$, and $D_{b,01}^M$. The last two sub-figures from left to right: the ratio of $D_{b,01}^M$ to $D_{b,00}^M$ and $D_{b,11}^M$.

Define $\mathcal{V} = \frac{m}{2T}\|\boldsymbol{v} - \boldsymbol{u}\|_2^2$ and $\mathcal{M} = \exp(-\mathcal{V})$. Then, a steady state solution of (17) is $f^\infty = c\mathcal{M}$, where the constant c is determined by initial condition of using mass conservation

$$\int_\Omega f^0 = \int_\Omega f^\infty \quad \Rightarrow \quad c = \frac{\int_\Omega f^0}{\int_\Omega \mathcal{M}}.$$

To see this, let us rewrite (17a) in an equivalent form. After multiplying ε on both sides of (17a) and moving the spatial derivatives to the right-hand side, we have the following.

$$\varepsilon \partial_t f = \nabla \cdot (\mathbf{D}\nabla f) - \frac{m}{T}\nabla \cdot \big(\mathbf{D}(\boldsymbol{u} - \boldsymbol{v})f\big) = \nabla \cdot \big(\mathbf{D}\nabla f + \frac{m}{T}\mathbf{D}(\boldsymbol{v} - \boldsymbol{u})f\big).$$

Notice, $\nabla\mathcal{V} = \frac{m}{T}(\boldsymbol{v} - \boldsymbol{u})$, we have

$$\varepsilon \partial_t f = \nabla \cdot \big(\mathbf{D}\nabla f + (\mathbf{D}\nabla\mathcal{V})f\big).$$

The function $\mathcal{M} = \exp(-\mathcal{V})$ gives $\mathcal{V} = -\ln \mathcal{M}$. Taking the gradient, we get $\nabla \mathcal{V} = -\frac{\nabla \mathcal{M}}{\mathcal{M}}$, which implies

$$\varepsilon \partial_t f = \nabla \cdot \left(\mathbf{D} \nabla f - \mathbf{D} \frac{\nabla \mathcal{M}}{\mathcal{M}} f \right) = \nabla \cdot \left(\mathbf{D} \mathcal{M} \frac{\mathcal{M} \nabla f - f \nabla \mathcal{M}}{\mathcal{M}^2} \right) = \nabla \cdot \left(\mathbf{D} \mathcal{M} \nabla \frac{f}{\mathcal{M}} \right).$$

From hereon, by the definition of $\mathcal{M}$, as f^∞ is time independent, it is straightforward to see that $f^\infty = c\mathcal{M}$ is a steady-state solution.

The function $\mathcal{M}$ is exponentially decaying, which presents significant challenges for numerical simulations. As an example, using the initial $f^0 = \frac{1}{400}$ with parameters $m = 10$ and $T = 1$ results in $\int_\Omega f^0 = 1$ and $\int_\Omega \mathcal{M} \approx 0.628$, leading to the constant $c \approx 1.6$. This indicates that the steady-state solution is very close to zero in a wide range of the computational domain. In this case, the distance between $\boldsymbol{v}$ and $\boldsymbol{u}$ is greater than 2.715 already gives $\mathcal{M} < 10^{-16}$. Notice that the computational domain $\Omega = [-10, 10]^2$ is much larger than a ball of radius 2.715. This causes the function f^∞ to be very close to zero over a wide region of the domain Ω, making the simulation very challenging as it approaches the steady state.

Let us take the initial as a uniform distribution $f(0, \boldsymbol{v}) = \frac{1}{400}$ and set the simulation end time $t^{\text{end}} = 20$. Fix time step size $\tau = 5 \cdot 10^{-4}$. We employ the NIPG method in $\mathbb{P}^2$ space with penalty parameter $\sigma = 1$. The tensor product of the 3-point Gauss quadrature is utilized for computing volume and face integrations. See Figure 5 for simulation results at the final time $t^{\text{end}} = 20$ with inverse collision time-scale $\varepsilon^{-1} = 10^1$, 10^2, and 10^3. Increasing the value of ε^{-1} will result in the system reaching a steady state at a faster rate. For $\varepsilon^{-1} = 10^1$ and 10^2, the positivity-preserving cell average limiter is not triggered, as the solution has not yet come close to steady state. For $\varepsilon^{-1} = 10^3$, the system has quickly reached steady state and the cell average limiter is triggered and our scheme preserves the positivity. The left sub-figure in Figure 5 shows the number of iterations for the Douglas–Rachford algorithm to converge at each step. The asymptotic linear convergence rate matches the theoretical result, see the middle and right sub-figures in Figure 5.

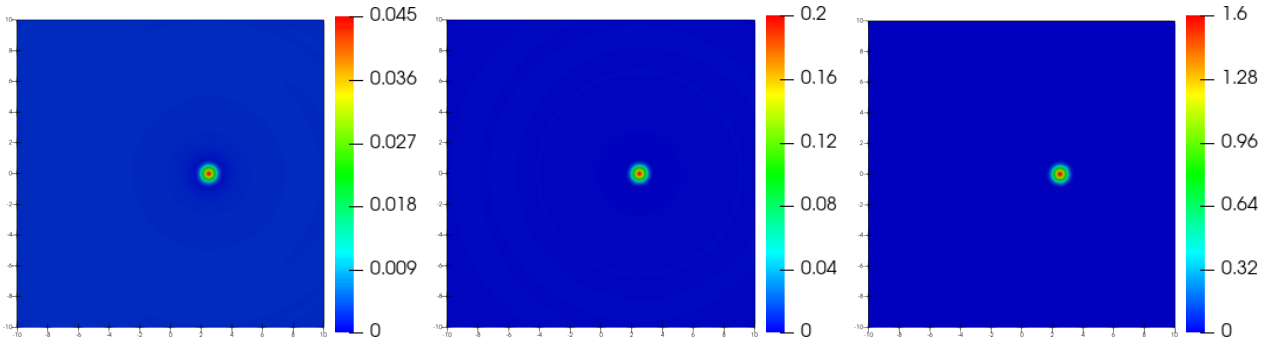

Figure 5: The $\mathbb{P}^2$ scheme. Snapshot of the discrete distribution function at time $t^{\text{end}} = 20$. From left to right: simulation results associated with the inverse collision time-scale $\varepsilon^{-1} = 10^1$, 10^2, and 10^3.

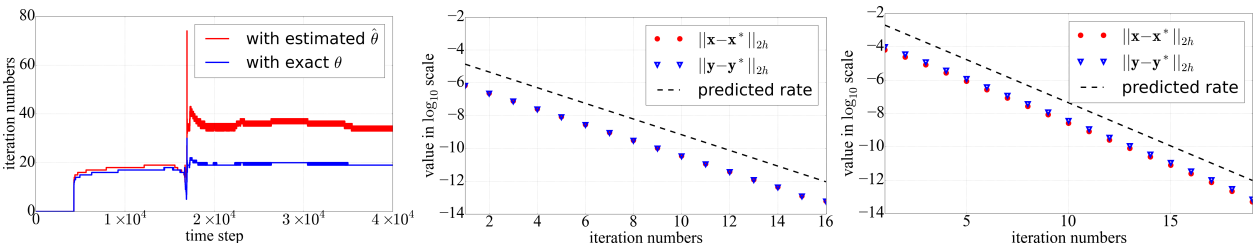

Figure 6: The convergence behavior of the cell average limiter associated with the simulation of using parameter $\varepsilon^{-1} = 10^3$. Left: the number of Douglas–Rachford iterations at each time step. Middle and right: the actual asymptotic linear convergence rate of the Douglas–Rachford splitting algorithm with optimal parameters at time step 10000 and 30000.

4.4. Beam relaxation

Let the computational domain $\Omega = [-10, 10]^2$ with the final simulation time $t^{\text{end}} = 200$ be set to be large enough so that the system reaches a nearly steady state. The initial condition is given by a Maxwellian parameterized by $n_0 = 1$, $m = 1$, $\boldsymbol{u}_0 = [7, 0]^{\mathrm{T}}$, and $T_0 = 0.25$.

$$f^0(\boldsymbol{v};\ n_0, m, \boldsymbol{u}_0, T_0) = \frac{n_0}{2\pi T_0/m} \exp\Big(- \frac{m}{2T_0} \|\boldsymbol{v} - \boldsymbol{u}_0\|_2^2 \Big).$$

We choose the parameters $\varepsilon^{-1} = 10^2$, $m = 1$, and $T = 1$ in (17). The coefficient matrix $\mathbf{D}$ is computed by setting $m_b = 100$, $T = 1$, and $\boldsymbol{u} = [0, 0]^{\mathrm{T}}$.

We use the NIPG method in $\mathbb{P}^2$ space with penalty parameter $\sigma = 1$. The mesh partition has been chosen as a 128-by-128 structure grid and the time step size is 5×10^{-4}. See Figure 7 for selected snapshots of the simulation result. Similarly to the fully nonlinear calculations performed in [65], we first observe the correct qualitative behavior of isotropization, where a ring structure forms, followed by a slower energy relaxation to the Maxwellian equilibrium. The cell average limiter is triggered and our algorithm successfully enforces the positivity of the distribution function at all times.

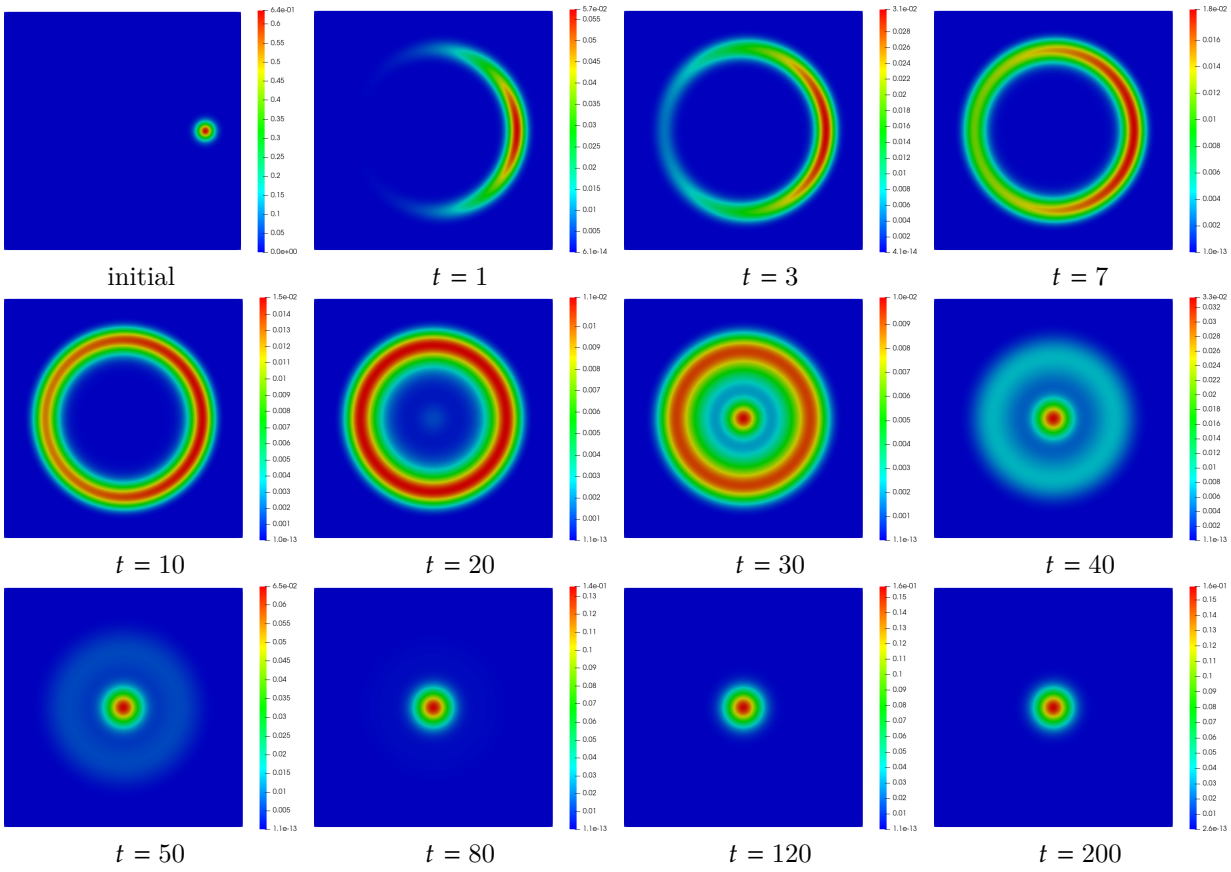

Figure 7: Beam relaxation test. Snapshots taken at initial ($t = 0$) and selected time ($t = 1, 3, \cdots, 200$). A ring structure forms in a relatively quick manner, then the system slowly approaches to the steady state – a "fat" Maxwellian distribution.

4.5. Importance of positivity preservation

Finally, we stress the importance of preserving the positivity of the distribution function by considering a relaxation of an initially non-equilibrium solution to a numerical equilibrium. Using the same setup as in

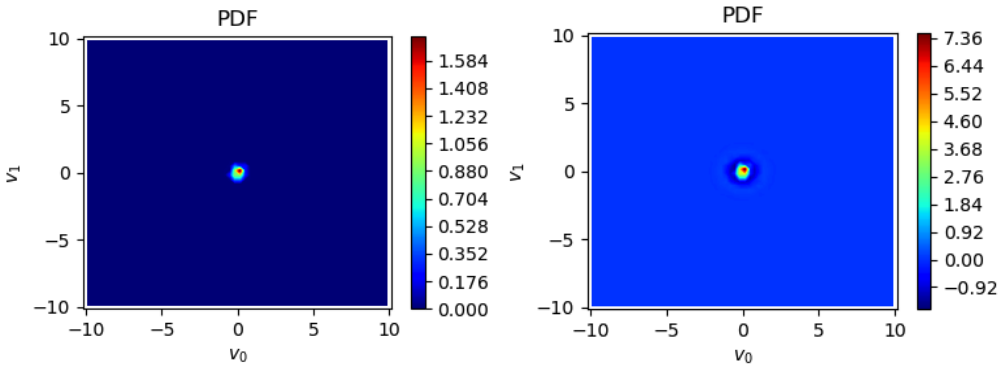

Figure 8: Importance of positivity: The distribution function at numerical equilibrium with (left) and without (right) positivity postprocessing.

the previous section, but with $m = 30$ and $\boldsymbol{u} = [0, 0]^{\mathrm{T}}$ the simulation begins from

$$f^0(\boldsymbol{v}) = \frac{1}{2\pi} \exp\Big(-\frac{1}{2}\|\boldsymbol{v}\|_2^2 \Big).$$

We run the simulation with a sufficiently large end time $t^{\mathrm{end}} = 20$ and let the system evolve to numerical equilibrium. In Figure 8 the numerical equilibrium is shown with and without the positivity enforcement. As can be seen, a large negative distribution function is present without the positivity postprocessing. In a linearized test particle relaxation model where the test particle species do not feedback to itself or the background species, these negative values are benign. However, in fully self-consistent non-linear settings, where the transport coefficients are functionals of the solution, they lead to the loss of SPD property of the diffusion tensor, and negative diffusion coefficients in the diagonal entries could emerge, leading to numerical instabilities; see Figure 9. As shown in the figures, the diagonal components of the diffusion matrix for the test particle species are obtained by self-consistently solving the Poisson equations in Eqs. (7) and (8) with far-field boundary conditions similarly defined in [66], result in large negative values when the positivity of the distribution function is not ensured. In contrast, they remain positive everywhere when the positivity of the solution is maintained. These results reflect the biharmonic nature of G and the loss of convexity caused by a significantly negative distribution function underscores the critical importance of maintaining the positivity of the solution in the RFP equations.

4.6. Performance of the cell average limiter

We aim to validate that the complexity of the Algorithm DR is $\mathcal{O}(N)$. To achieve this, we compare the running time on a single CPU, while varying the total number of degrees of freedom (DOF). For an $\mathcal{O}(N)$ scheme, refining the mesh resolution by a factor of two in the two-dimensional space results in a fourfold increase in the total number of DOF. Therefore, in an ideal case, we expect that the running time on a single CPU will also increase by a factor of four.

Let us utilize the same synthetic problem as described in [53, Appendix A] to generate synthetic data. Define $\boldsymbol{w}$ in (24) as the point values of the following function on a uniform grid of resolution Δx on the computational domain $[0, 1]^2$:

$$f(x, y) = \begin{cases} -0.25, & -\frac{\delta}{4} + 0.25 \le x \le \frac{\delta}{4} + 0.25 \\ -0.25, & -\frac{\delta}{4} + 0.75 \le x \le \frac{\delta}{4} + 0.75 \\ \cos^8(2\pi x) + 10^{-13}, & \text{otherwise} \end{cases}, \tag{35}$$

where $\delta > 0$ is a parameter that controls the ratio of negative point values. We choose the value of δ such that the ratio of negative point values is 5%.

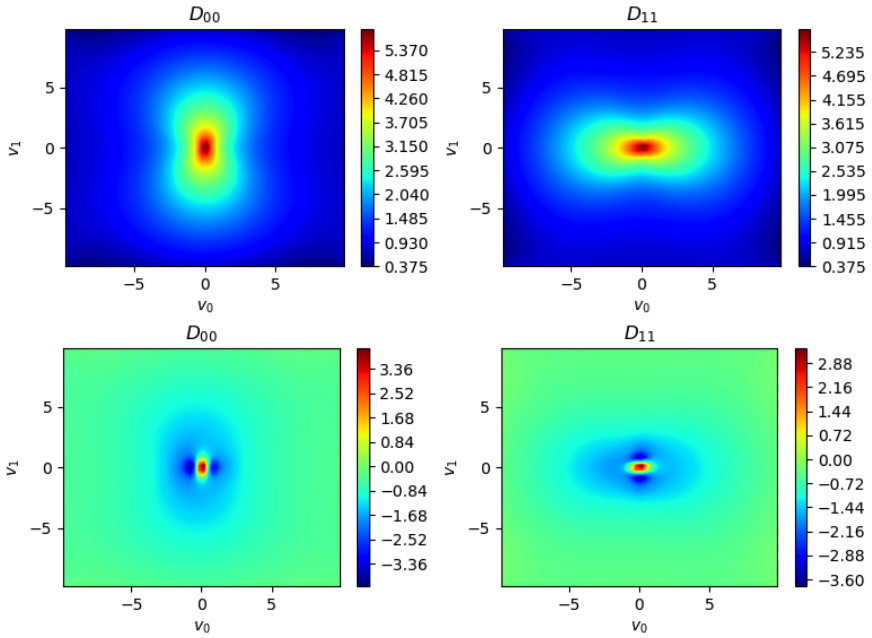

Figure 9: Importance of positivity: The diagonal entries of the diffusion coefficients for the test particle species, computed self consistently at numerical equilibrium with (top) and without (bottom) positivity postprocessing.

We solve the constrained minimization problem (24) to machine precision 100 times on a single CPU to limit cell averages. The processor utilized is the Intel Xeon CPU E5-2660 v3 2.60 GHz. Table 3 displays the average CPU time for a single run with mesh resolution $\Delta x = 2^{-7}, 2^{-8}, \cdots, 2^{-11}$ and Figure 3 plots the CPU time versus the total number of DOF on a $\log_2$–$\log_2$ scale. The performance of Algorithm DR is as expected; see the dashed line of slope one in Figure 3.

Δx	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}
time [s]	9.048×10^{-3}	3.068×10^{-2}	1.029×10^{-1}	5.979×10^{-1}	2.705×10^{0}

Table 3: Average CPU time for Algorithm DR, calculated over 100 repetitions with continuous mesh refinement.

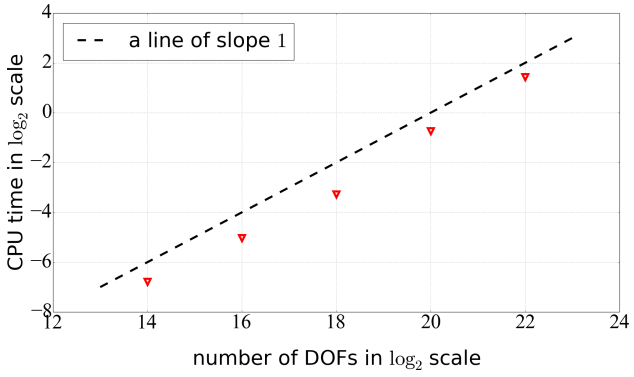

Figure 10: Plot the average CPU time on a $\log_2$–$\log_2$ scale, calculated over 100 repetitions with continuous mesh refinement.

5. Concluding remarks

In this paper, we have considered an optimization-based positivity-preserving method for a semi-implicit DG scheme that solves the Fokker–Planck equations. The DG cell averages are enforced to be non-negative without affecting accuracy and global conservation by a constrained minimization, solved by the Douglas–Rachford splitting method with nearly optimal parameters. With non-nonnegative cell averages, the Zhang–Shu limiter is used to eliminate undershoot point values in the DG polynomials. The practical advantages of this approach include high accuracy, efficiency, and ease of implementation. Numerical tests suggest that this approach can efficiently improve the robustness of semi-implicit high-order DG schemes.

Acknowledgments

J.H. is partially supported under the NSF grant DMS-2409858 and DOE grant DE-SC0023164.

W.T.T was partially supported by Triad National Security, LLC under contract 89233218CNA000001 and DOE Office of Applied Scientific Computing Research (ASCR) through the Mathematical Multifaceted Integrated Capability Centers (MMICCs) program.

X.Z. is suppored by NSF DMS-2208518.

References

[1] L. D. Landau, The kinetic equation in the case of Coulomb interaction, Zh. Eksper. i Theoret. Fiz. 7 (2) (1937).
[2] M. N. Rosenbluth, W. M. Macdonald, D. L. Judd, Fokker-Planck equation for an inverse-square force, Phys. Rev. Lett. 107 (1) (1957) 1–6.
[3] G. Cooper, Compton Fokker-Planck equation for hot plasmas, Physical Review D 3 (10) (1970) 2312–2316.
[4] M. Gatu-Johnson, P. J. Adrian, B. D. Appelbe, A. J. Crilly, e. a. Forrest, C J, Impact of mid-z gas fill on dynamics and performance of shock-driven implosions at the Omega laser, Physical Review E 109 (2024) 065201.
[5] W. T. Taitano, A. N. Simakov, L. Chacón, B. D. Keenan, Yield degradation in inertial-confinementfusion implosions due to shock-driven kinetic fuel-species stratification and viscous heating, Physics of Plasmas 25 (2018) 056310.
[6] B. D. Keenan, A. N. Simakov, W. T. Taitano, L. Chacón, Ion species stratification within strong shocks in two-ion plasmas, Physics of Plasmas 25 (2018) 032103.
[7] O. Larroche, H. G. Rinderknecht, M. J. Rosenberg, Nuclear yield reduction in inertial confinement fusion exploding-pusher targets explained by fuel-pusher mixing through hybrid kinetic-fluid modeling, Physical Review E 98 (2018) 031201 (R).
[8] E. A. Anderson, L. Chacón, W. T. Taitano, A. N. Simakov, B. D. Keenan, Fully kinetic simulations of strong steady-state collisional planar plasma shocks, Physical Review E 104 (5) (2021) 055205.
[9] O. Mannion, W. Taitano, B. Appelbe, A. Crilly, C. Forrest, V. Glebov, J. Knauer, P. McKenty, Z. Mohamed, C. Stoeckl, B. Keenan, J. Cittenden, P. Adrian, J. Frenje, N. Kabadi, M. G. Johnson, S. Regan, Evidence of non-Maxwellian ion velocity distributions in spherical shock-driven implosions, Phys. Rev. E 108 (3) (2023) 035201.
[10] O. Larroche, Kinetic simulations of fuel ion transport in ICF target implosions, European Physical Journal D 27 (2) (2003) 131–146.
[11] O. Larroche, Ion fokker-planck simulation of d-3he gas target implosions, Physics of Plasmas 19 (12) (2012).
[12] A. Inglebert, B. Canaud, , O. Larroche, Species separation and modification of neutron diagnostics in inertial-confinement fusion, Erophysics Letters 107 (6) (2014) 65003.
[13] B. E. Peigney, O. Larroche, V. Tikhonchuk, Fokker–planckkinetic modeling of suprathermal α-particles in a fusion plasma, Journal of Computational Physics 278 (2014) 416–444.
[14] W. T. Taitano, B. D. Keenan, L. Chacón, S. E. Anderson, H. R. Hammer, A. N. Simakov, An Eulerian Vlasov-Fokker–Planck algorithm for spherical implosion simulations of inertial confinement fusion capsule, Computer Physics Communications 263 (2021) 107861.
[15] W. T. Taitano, L. Chacon, A. N. Simakov, An adaptive, conservative 0D-2V multispecies Rosenbluth–Fokker–Planck solver for arbitrarily disparate mass and temperature regimes, Journal of Computational Physics 318 (2016) 391–420.
[16] W. T. Taitano, L. Chacón, S. A. N, An equilibrium-preserving discretization for the nonlinear rosenbluth–fokker–planck operator in arbitrary multi-dimensional geometry, Journal of Computational Physics 339 (2017) 453–460.
[17] X. Zhang, C.-W. Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, Journal of Computational Physics 229 (9) (2010) 3091–3120.
[18] X. Zhang, C.-W. Shu, On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes, Journal of Computational Physics 229 (23) (2010) 8918–8934.
[19] Z. Xu, X. Zhang, Bound-preserving high-order schemes, in: Handbook of numerical analysis, Vol. 18, Elsevier, 2017, pp. 81–102.
[20] X. Zhang, C.-W. Shu, A minimum entropy principle of high order schemes for gas dynamics equations, Numerische Mathematik 121 (3) (2012) 545–563.

[21] X. Zhang, C.-W. Shu, Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms, Journal of Computational Physics 230 (4) (2011) 1238–1248.
[22] Y. Xing, X. Zhang, C.-W. Shu, Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations, Advances in Water Resources 33 (12) (2010) 1476–1493.
[23] X. Zhang, On positivity-preserving high order discontinuous Galerkin schemes for compressible Navier–Stokes equations, Journal of Computational Physics 328 (2017) 301–343.
[24] Z. Chen, H. Huang, J. Yan, Third order maximum-principle-satisfying direct discontinuous Galerkin methods for time dependent convection diffusion equations on unstructured triangular meshes, Journal of Computational Physics 308 (2016) 198–217.
[25] Z. Sun, J. A. Carrillo, C.-W. Shu, A discontinuous Galerkin method for nonlinear parabolic equations and gradient flow problems with interaction potentials, Journal of Computational Physics 352 (2018) 76–104.
[26] S. Srinivasan, J. Poggie, X. Zhang, A positivity-preserving high order discontinuous Galerkin scheme for convection–diffusion equations, Journal of Computational Physics 366 (2018) 120–143.
[27] C. Liu, X. Zhang, A positivity-preserving implicit-explicit scheme with high order polynomial basis for compressible Navier–Stokes equations, Journal of Computational Physics 493 (2023) 112496.
[28] H. Li, X. Zhang, On the monotonicity and discrete maximum principle of the finite difference implementation of $C^0 - Q^2$ finite element method, Numerische Mathematik 145 (2020) 437–472.
[29] L. Cross, X. Zhang, On the monotonicity of Q^2 spectral element method for Laplacian on quasi-uniform rectangular meshes, Communications in Computational Physics 35 (2024) 160–180.
[30] L. Cross, X. Zhang, Monotonicity of Q^3 spectral element method for discrete Laplacian, Annals of Applied Mathematics (to appear) (2024).
[31] H. Li, X. Zhang, A monotone Q^1 finite element method for anisotropic elliptic equations, to appear in the Beijing Journal of Pure and Applied Mathematics (2024).
[32] J. Shen, X. Zhang, Discrete maximum principle of a high order finite difference scheme for a generalized Allen–Cahn equation, Communications in Mathematical Sciences 20 (5) (2022) 1409–1436.
[33] J. Hu, X. Zhang, Positivity-preserving and energy-dissipative finite difference schemes for the Fokker–Planck and Keller–Segel equations, IMA Journal of Numerical Analysis 43 (3) (2023) 1450–1484.
[34] C. Liu, Y. Gao, X. Zhang, Structure preserving schemes for Fokker–Planck equations of irreversible processes, Journal of Scientific Computing 98 (1) (2024) 4.
[35] X. Zhang, Recent progress on Q^k spectral element method: Accuracy, monotonicity and applications, to appear in ICOSAHOM 2023 Conference Proceedings, Lecture Notes in Computational Science and Engineering, Springer Nature Switzerland AG (2024).
[36] W. Höhn, H. D. Mittelmann, Some remarks on the discrete maximum-principle for finite elements of higher order, Computing 27 (2) (1981) 145–154.
[37] H. Li, X. Zhang, A high order accurate bound-preserving compact finite difference scheme for two-dimensional incompressible flow, Communications on Applied Mathematics and Computation 6 (1) (2024) 113–141.
[38] J.-L. Guermond, M. Maier, B. Popov, I. Tomas, Second-order invariant domain preserving approximation of the compressible Navier–Stokes equations, Computer Methods in Applied Mechanics and Engineering 375 (2021) 113608.
[39] G. R. Barrenechea, V. John, P. Knobloch, Finite element methods respecting the discrete maximum principle for convection-diffusion equations, SIAM Review 66 (1) (2024) 3–88.
[40] D. Kuzmin, M. J. Shashkov, D. Svyatskiy, A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems, Journal of Computational Physics 228 (9) (2009) 3448–3463.
[41] J.-L. Guermond, B. Popov, I. Tomas, Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems, Computer Methods in Applied Mechanics and Engineering 347 (2019) 143–175.
[42] F. Frank, A. Rupp, D. Kuzmin, Bound-preserving flux limiting schemes for DG discretizations of conservation laws with applications to the Cahn–Hilliard equation, Computer Methods in Applied Mechanics and Engineering 359 (2020) 112665.
[43] C. Liu, D. Ray, C. Thiele, L. Lin, B. Riviere, A pressure-correction and bound-preserving discretization of the phase-field method for variable density two-phase flows, Journal of Computational Physics 449 (2022) 110769.
[44] M. Sarraf Joshaghani, B. Riviere, Bound-preserving discontinuous Galerkin methods for compressible two-phase flows in porous media, International Journal for Numerical Methods in Engineering 125 (4) (2024) e7396.
[45] M. Joshaghani, B. Riviere, M. Sekachev, Maximum-principle-satisfying discontinuous Galerkin methods for incompressible two-phase immiscible flow, Computer Methods in Applied Mechanics and Engineering 391 (2022) 114550.
[46] Z. Xu, Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem, Mathematics of Computation 83 (289) (2014) 2213–2238.
[47] O. Guba, M. Taylor, A. St-Cyr, Optimization-based limiters for the spectral element method, Journal of Computational Physics 267 (2014) 176–195.
[48] J. J. van der Vegt, Y. Xia, Y. Xu, Positivity preserving limiters for time-implicit higher order accurate discontinuous Galerkin discretizations, SIAM journal on scientific computing 41 (3) (2019) A2037–A2063.
[49] Q. Cheng, J. Shen, A new Lagrange multiplier approach for constructing structure preserving schemes, II. Bound preserving, SIAM Journal on Numerical Analysis 60 (3) (2022) 970–998.
[50] F. Ruppenthal, D. Kuzmin, Optimal control using flux potentials: A way to construct bound-preserving finite element schemes for conservation laws, Journal of Computational and Applied Mathematics 434 (2023) 115351.
[51] R. C. Kirby, D. Shapero, High-order bounds-satisfying approximation of partial differential equations via finite element variational inequalities, Numerische Mathematik (2024) 1–21.
[52] C. Liu, B. Riviere, J. Shen, X. Zhang, A simple and efficient convex optimization based bound-preserving high order

accurate limiter for Cahn–Hilliard–Navier–Stokes system, SIAM Journal on Scientific Computing 46 (3) (2024) A1923–A1948.
[53] C. Liu, G. T. Buzzard, X. Zhang, An optimization based limiter for enforcing positivity in a semi-implicit discontinuous Galerkin scheme for compressible Navier–Stokes equations, Journal of Computational Physics 519 (2024) 113440.
[54] B. Cockburn, G. E. Karniadakis, C.-W. Shu, Discontinuous Galerkin methods: theory, computation and applications, Vol. 11, Springer Science & Business Media, 2012.
[55] C.-W. Shu, Discontinuous Galerkin method for time-dependent problems: survey and recent developments, Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations: 2012 John H Barrett Memorial Lectures (2014) 25–62.
[56] P.-L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM Journal on Numerical Analysis 16 (6) (1979) 964–979.
[57] A. Chambolle, T. Pock, An introduction to continuous optimization for imaging, Acta Numerica 25 (2016) 161–319.
[58] M. Fortin, R. Glowinski, Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems, Elsevier, 2000.
[59] T. Goldstein, S. Osher, The split Bregman method for L1-regularized problems, SIAM journal on imaging sciences 2 (2) (2009) 323–343.
[60] L. Demanet, X. Zhang, Eventual linear convergence of the Douglas–Rachford iteration for basis pursuit, Mathematics of Computation 85 (297) (2016) 209–238.
[61] W. T. Taitano, L. Chacon, A. Simakov, K. Molvig, A mass, momentum, and energy conserving, fully implicit, scalable algorithm for the multi-dimensional, multi-species Rosenbluth–Fokker–Planck equation, Journal of Computational Physics 297 (2015) 357–380.
[62] F. Frank, C. Liu, F. O. Alpak, B. Riviere, A finite volume/discontinuous Galerkin method for the advective Cahn–Hilliard equation with degenerate mobility on porous domains stemming from micro-CT imaging, Computational Geosciences 22 (2018) 543–563.
[63] B. Riviere, Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation, SIAM, 2008.
[64] V. Ilin, J. Hu, Z. Wang, Transport based particle methods for the Fokker–Planck–Landau equation, arXiv preprint arXiv:2405.10392 (2024).
[65] W. T. Taitano, L. Chacón, A. N. Simakov, S. E. Anderson, A conservative phase-space moving-grid strategy for a 1D-2V Vlasov-Fokker-Planck solver, Computer Physics Communications 258 (2021) 107547.
[66] L. Chac'on, D. Barnes, D. Knoll, G. Miley, An implicit energy-conservative 2D Fokker-Planck algorithm, Journal of Computational Physics 157 (2000) 618–653.