Version of Record: https://www.sciencedirect.com/science/article/pii/S0021999124000330
Manuscript_c00e437b044b434cf387ec6b93d32ff4

A Collision-Based Hybrid Method for the BGK Equation

Minwoo Shin[a], Cory D. Hauck[b,*], Ryan G. McClarren[c,*]

[a] *Yonsei University, School of Mathematics and Computing, Seoul, 03722, Republic of Korea*
[b] *Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, 37831, TN, USA*
[c] *Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, 46556, IN, USA*

Abstract

We apply the collision-based hybrid method introduced by Hauck and McClarren [1] to the Boltzmann equation with the BGK operator and a hyperbolic scaling. An implicit treatment of the source term is used to handle stiffness associated with the BGK operator. Although it helps the numerical scheme become stable with a large time step size, it is still not obvious to achieve the desired order of accuracy due to the relationship between the size of the spatial cell and the mean free path. Without asymptotic preserving property, a very restricted grid size is required to resolve the mean free path, which is not practical. Our approaches are based on the noncollision-collision decomposition of the BGK equation. We introduce the arbitrary order of nodal discontinuous Galerkin (DG) discretization in space with a semi-implicit time-stepping method; we employ the backward Euler time integration for the uncollided equation and the 2nd order predictor-corrector scheme for the collided equation, i.e., both source terms in uncollided and collided equations are treated implicitly and only streaming term in the collided equation is solved explicitly. This improves the computational efficiency without the complexity of the numerical implementation. Numerical results are presented for various Knudsen numbers to present the effectiveness and accuracy of our hybrid method. Also, we compare the solutions of the hybrid and non-hybrid schemes.

Keywords: BGK equation, discrete velocity, discontinuous Galerkin, hybrid decomposition, asymptotic-preserving, gas-injection

1. Introduction

The Bhatnagar-Gross-Krook (BGK) equation is a well-known kinetic theory model used to simulate the non-equilibrium behavior of dilute gases. It is an approximation of the Boltzmann equation that replaces the Boltzmann collision operator, a nonlinear integral operator modeling binary collisions, by the BGK operator [2, 3], a simpler nonlinear relaxation model. This modification significantly reduces the computational cost of kinetic simulations of dilute gases while still recovering both the equilibrium and streaming behavior of the Boltzmann equation in collision-dominated and collisionless limits, respectively. The BGK operator also preserves the conservation and entropy dissipation properties of the Boltzmann operator, and recent work has extended these properties to the multispecies setting [4, 5, 6].

Approximation with the BGK equation does have two major drawbacks. First, it does not capture all of the transport properties of the Boltzmann equation in collisional regimes. Most notably, it fails to capture the correct Prandtl number (essentially the ratio of viscosity to thermal

*Corresponding authors
Email addresses: mjmj0210@gmail.com (Minwoo Shin), hauckc@ornl.gov (Cory D. Hauck), rmcclarr@nd.edu (Ryan G. McClarren)

© 2024 published by Elsevier. This manuscript is made available under the Elsevier user license
https://www.elsevier.com/open-access/userlicense/1.0/

conductivity). Hence it does not agree with the compressible Navier-Stokes equations that are derived from the Boltzmann equation in collisional regimes. However, this problem can be remedied via a slight modification that leads to an ellipsoidal statistical operator [7, 8], sometimes referred to as ES-BGK [9, Section 3.6]. The second drawback is that the structural properties of the BGK operator rely on the assumption that the collision frequency is independent of the microscopic velocity, which is known to be unphysical [9, 10]. Extensions to velocity-dependent frequencies have been explored in single species [11] and multi-species [5, 12] settings. While these models add more physical realism, they come with a significant increase in computational cost.

1.1. Related work

On the numerical side, the BGK equation has been the object of extensive study. Aside from the cost of discretizing phase space, there are two primary challenges, both of which occur in collisional regimes. The first challenge is the stiffness of the BGK operator, which suggests an implicit treatment in order to avoid excessively small time steps. (See, however, [13] for a recent alternative approach.) The second challenge is to recover—at the numerical level—consistent numerical discretizations of the Euler and Navier-Stokes system for compressible gas flow that are derived from the BGK model via a Chapman-Enskog expansion [9, Chapter 4].

In [14], it was shown that a backward Euler time discretization of the BGK operator could be implemented in an explicit fashion. This numerical "trick" enables stable time stepping for large time steps across a wide range of collisional regimes, and for this reason, it has become the basis of many subsequent numerical schemes. In [15], various discretizations of the BGK and Shakhov model [16] are considered and compared.[1] In [17], a method based on the decomposition of the kinetic distribution into macroscopic (equilibrium) and microscopic (non-equilibrium) parts is proposed in order to recover a consistent Navier-Stokes limit when the Knudsen number is sufficiently small. Simulations of the ES-BGK model for small Knudsen numbers have also been developed [18, 19, 20] in order to recover the correct Navier-Stokes equations at the numerical level. There has also been extensive research on implicit–explicit schemes for the BGK equations and ES-BGK equation; see for instance [21, 22, 23]. A gas-kinetic scheme for the BGK model was proposed and compared to the compressible Navier–Stokes equations in the collisional regime [24].

Even when the collision operator is treated implicitly, an explicit treatment of the advection operator in the BGK equation can lead to restrictive time steps. There are two common reasons: long time scales (i.e. a diffusive-type scaling) that lead to incompressible equations in the infinite collisional limit [25] or problems for which the maximum velocity in the computational domain is significantly larger than the fluid sound speed. One way to address such restrictions is to take a fully implicit approach, which is common for time-dependent kinetic equations in radiation transport contexts [26, Section 1.2] and has been proposed for electron transport problems in [27, 28]. In both these cases, diffusive limits are important. Fully implicit methods have also been proposed for collisional dilute gases [29, 30, 31] and collisionless plasmas [32]. All of these approaches require sophisticated iterative strategies to manage the cost and memory requirement of the implicit update. An alternative is to use semi-Lagrangian methods [33, 34, 35], which rely on a characteristic formulation. There the main effort lies in the interpolation of point-wise values since characteristics of the advection operator in the BGK equation do not always intersect grid points.

[1] The Shakhov model is designed to capture the correct Prandtl number, but unlike BGK, it is not guaranteed to generate a positive solution.

1.2. Hybrid approach: benefits and drawbacks

In the current paper, we focus on the BGK equation in a hyperbolic scaling and propose a method to address stiffness due to large phase space velocities. Rather than use an IMEX or fully implicit time-stepping strategy, we instead leverage the hybrid approach introduced in [1]. While this approach was originally proposed for linear transport equations with fully implicit time integration [1, 36, 37], we use it here to solve the BGK model in a semi-implicit fashion [38, 39].

The hybrid strategy is based on the method of *first-collision source* [40, 41], in which the kinetic equation is decomposed into the sum of an uncollided equation and a collided equation. The uncollided equation is a damped linear advection equation with no in-scattering and hence can be solved implicitly rather easily with high resolution and in parallel; meanwhile, the collided equation is expected to have a solution that can be approximated with a low-fidelity velocity discretization.

We use the Euler fluid approximation as the velocity discretization for the collided equation, but unlike previous applications of similar hybrid methods, the advection is solved explicitly with a 2nd-order Runge-Kutta (predictor-corrector) time integrator. These choices are flexible; for example, a Navier-Stokes model can be used for the collided equation and a semi-Lagrangian approach can be taken for the uncollided equation.

There are two main benefits of the hybrid decomposition. The first benefit is the simplicity of the numeric implementation. Indeed, if a fluid code is readily available, then one need only add a relatively simple implicit solver for the uncollided equations. Moreover, as mentioned above, the choice of discretization for each equation can be flexible. The second benefit is a less restrictive CFL condition that is based on the maximum eigenvalues of the flux Jacobian in the fluid solver rather than the maximum velocity in the computational domain. As a consequence, larger time steps can be taken compared to IMEX nodal DG methods for the micro-macro decomposition of the BGK system. In other words, the collision-noncollision splitting method offers a straightforward implementation of the BGK solver. Its formulation enables the adoption of a fully implicit approach for the uncollided equation and a semi-implicit method for the collided equation. This setup facilitates the exclusion of the maximum velocity numerator from the CFL condition, a requirement in the CFL for IMEX schemes with micro-macro decomposition [42]. This is especially beneficial for problems including high-speed flows, rarefied gas dynamics, high-temperature plasmas, etc.

The main drawback of the hybrid approach is that it induces both general time-stepping errors and consistency errors in the coarse approximation of the collided equation that depend on the collisionality of the problem. Because the method includes a remapping procedure after each time step, the mixing of errors can be difficult to track, making even a formal analysis difficult. What is certain, however, is that the consistency error from the hybrid approximation will eventually cause saturation in convergence as the space-time mesh is resolved. In our experiments, we observe that this saturation happens only at very low error levels, and if saturation does become a concern, there are ways to correct it [36, 37].

The remainder of the paper is organized as follows. We begin with a derivation of the uncollided-collided decomposition of a BGK equation in Section 2. We present the discretization details for the uncollided and collided equations in Section 3 and perform an asymptotic analysis. In Section 4, we present numerical results on several benchmark problems to demonstrate the simplicity, robustness, efficiency, asymptotic properties of the hybrid approach. We also compare the hybrid method to a previously developed implicit-explicit method. Finally, in Section 5, we draw conclusions and make suggestions for future work.

2. The hybrid formulation

2.1. The BGK equation

Given a positive integer d, let $X \subset \mathbb{R}^d$ be an open bounded domain with Lipschitz boundary. Let $\Gamma = \partial X \times \mathbb{R}^d$ and $\Gamma^{\pm} = \{(\boldsymbol{x}, \boldsymbol{v}) \in \Gamma \colon \pm \boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{v} > 0\}$, where $\boldsymbol{n}(\boldsymbol{x})$ is the outward normal with respect to X, defined at a.e. $\boldsymbol{x} \in \partial X$. The BGK model for the *kinetic distribution function* F with a source term S is

$$\begin{cases} \partial_t F(\boldsymbol{x}, \boldsymbol{v}, t) + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} F = \dfrac{1}{\epsilon} \left(\mathcal{M}_F(\boldsymbol{x}, \boldsymbol{v}, t) - F(\boldsymbol{x}, \boldsymbol{v}, t) \right) + S(\boldsymbol{x}, \boldsymbol{v}, t), & (\boldsymbol{x}, \boldsymbol{v}) \in X \times \mathbb{R}^d, \ t > 0, \quad \text{(1a)} \\ F(\boldsymbol{x}, \boldsymbol{v}, t) = G^-(\boldsymbol{x}, \boldsymbol{v}, t), & (\boldsymbol{x}, \boldsymbol{v}) \in \Gamma^-, \ t > 0, \quad \text{(1b)} \\ F(\boldsymbol{x}, \boldsymbol{v}, 0) = G^0(\boldsymbol{x}, \boldsymbol{v}), & (\boldsymbol{x}, \boldsymbol{v}) \in X \times \mathbb{R}^d, \quad \text{(1c)} \end{cases}$$

where the inflow G^-, initial condition G^0, and source term S are given non-negative functions. The function F, which depends on position $\boldsymbol{x} \in X$, velocity $\boldsymbol{v} \in \mathbb{R}^d$, and time $t \geq 0$, gives the mass density of particles with respect to the phase space measure $\boldsymbol{dvdx}$. The parameter $\epsilon > 0$ is the Knudsen number, i.e., the ratio of the mean free path between collisions to the length scale of the domain; in this work, we assume ϵ is a fixed constant. The function $\mathcal{M}_F$ is given by

$$\mathcal{M}_F = \mathcal{M}_F(\boldsymbol{x}, \boldsymbol{v}, t) = \frac{\rho_F(\boldsymbol{x}, t)}{(2\pi\theta_F(\boldsymbol{x}, t))^{d/2}} \exp\left(-\frac{|\boldsymbol{v} - \boldsymbol{u}_F(\boldsymbol{x}, t)|^2}{2\theta_F(\boldsymbol{x}, t)} \right), \tag{2}$$

where the *primitive variables* $\rho_F(\boldsymbol{x}, t) \geq 0$, $\boldsymbol{u}_F(\boldsymbol{x}, t) \in \mathbb{R}^d$, and $\theta_F(\boldsymbol{x}, t) \geq 0$ are the local mass density, mean velocity, and temperature, respectively; functions of the form in (2) are referred to as *Maxwellians*.

2.2. Moment equations

The local mass density ρ_F, momentum density $\boldsymbol{m}_F$, and total energy density E_F, are given by

$$\boldsymbol{q}_F := \begin{pmatrix} \rho_F \\ \boldsymbol{m_F} \\ E_F \end{pmatrix} = \int_{\mathbb{R}^d} \boldsymbol{e} F \, \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^d} \begin{pmatrix} F \\ \boldsymbol{v} F \\ \frac{1}{2}|\boldsymbol{v}|^2 F \end{pmatrix} \mathrm{d}\boldsymbol{v}, \tag{3}$$

where the components of $\boldsymbol{e} = \boldsymbol{e}(\boldsymbol{v}) := (1, \boldsymbol{v}, \frac{1}{2}|\boldsymbol{v}|^2)^T$ are referred to as *collision invariants*, since

$$\int_{\mathbb{R}^d} (\mathcal{M}_F - F) \boldsymbol{e} dv = 0. \tag{4}$$

We also denote by $\mathcal{E}$ the map which takes a set of moments to the associated Mawellian, i.e., $\mathcal{M}_F = \mathcal{E}(\boldsymbol{q}_F)$.

The condition in (4), which is intended to mimic properties of the Boltzmann collision operator (see e.g. [43, Chapter 3]), uniquely determines the primitive variables ρ_F, $\boldsymbol{u}_F$, θ_F that parameterize the Maxwellian in (2). It also establishes the relationship between the moments $\boldsymbol{m}_F$ and E_F and the variables $\boldsymbol{u}_F$ and θ_F. Specifically,

$$\boldsymbol{m}_F = \rho_F \boldsymbol{u}_F \qquad \text{and} \quad E_F = \frac{1}{2}\rho_F |\boldsymbol{u}_F|^2 + \frac{d}{2}\rho_F \theta_F. \tag{5}$$

If F is a solution to the BGK equation (1a), its moments satisfy the following system of conservation laws:

$$\frac{\partial \rho_F}{\partial t} + \nabla_x \cdot (\rho_F \boldsymbol{u}_F) = 0, \tag{6a}$$

$$\frac{\partial (\rho_F \boldsymbol{u}_F)}{\partial t} + \nabla_x \cdot (\rho_F \boldsymbol{u}_F \otimes \boldsymbol{u}_F + \boldsymbol{P}_F) = 0, \tag{6b}$$

$$\frac{\partial E_F}{\partial t} + \nabla_x \cdot (E_F \boldsymbol{u}_F + \boldsymbol{P}_F \boldsymbol{u}_F + Q_F) = 0, \tag{6c}$$

where

$$\boldsymbol{P}_F = \int_{\mathbb{R}^d} (\boldsymbol{v} - \boldsymbol{u}_F) \otimes (\boldsymbol{v} - \boldsymbol{u}_F) F \, \mathrm{d}\boldsymbol{v} \quad \text{and} \quad Q_F = \int_{\mathbb{R}^d} \frac{1}{2} (\boldsymbol{v} - \boldsymbol{u}_F) |\boldsymbol{v} - \boldsymbol{u}_F|^2 F \, \mathrm{d}\boldsymbol{v} \tag{7}$$

are the pressure tensor and heat flux, respectively. The equations in (6), which are not closed, are derived by integrating (1a) against the vector $\boldsymbol{e}$ of collision invariants and then invoking (4) to make the right-hand side vanish. To close (6), one must specify formulas for $\boldsymbol{P}_F$ and Q_F in terms of the moments ρ_F, $\boldsymbol{m}_F$, and E_F or, equivalently, the primitive variables ρ_F, $\boldsymbol{m}_F$, and θ_F. In the fluid limit, when $\epsilon \to 0$, the scaling in (1a) suggests that F can be approximated by $\mathcal{M}_F$ in the formulas for $\boldsymbol{P}_F$ and Q_F (see for example [44, Section 2.8]), in which case

$$\boldsymbol{P}_F \approx \int_{\mathbb{R}^d} (\boldsymbol{v} - \boldsymbol{u}_F) \otimes (\boldsymbol{v} - \boldsymbol{u}_F) \mathcal{M}_F \, \mathrm{d}\boldsymbol{v} = \rho_F \theta_F \boldsymbol{I} \quad \text{and} \quad Q_F \approx \int_{\mathbb{R}^d} \frac{1}{2} (\boldsymbol{v} - \boldsymbol{u}_F) |\boldsymbol{v} - \boldsymbol{u}_F|^2 \mathcal{M}_F \, \mathrm{d}\boldsymbol{v} = 0, \tag{8}$$

where $\boldsymbol{I}$ is the $d \times d$ identity matrix. Using these approximations for $\boldsymbol{P}_F$ and Q_F to close (6) recovers the Euler equations for a gamma-law gas with pressure law $p(\rho, \theta) = \rho\theta$. Because a generic gamma-law gas has equation of state $p = \frac{d}{2}(\gamma - 1)\rho_F \theta_F$ (see .g. [45, pg. 13]), it follows that $\gamma = \frac{d+2}{d}$.

2.3. The hybrid method

We adopt the collision-based hybrid approach introduced in [1] and decompose the distribution function F as the sum of an uncollided component F_{u} and an uncollided component F_{c}. Let $0 = t_0 < t_1 < \dots$ be a sequence of points in a temporal grid. For $t \in (t_n, t_{n+1})$, let F_{u} and F_{c} solve

$$\begin{cases} \partial_t F_{\mathrm{u}}(\boldsymbol{x}, \boldsymbol{v}, t) + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} F_{\mathrm{u}} + \dfrac{1}{\epsilon} F_{\mathrm{u}}(\boldsymbol{x}, \boldsymbol{v}, t) = S(\boldsymbol{x}, \boldsymbol{v}, t), & (\boldsymbol{x}, \boldsymbol{v}) \in X \times \mathbb{R}^d, \ t \in (t_n, t_{n+1}), \quad (9a) \\ F_{\mathrm{u}}(\boldsymbol{x}, \boldsymbol{v}, t) = G^-(\boldsymbol{x}, \boldsymbol{v}, t), & (\boldsymbol{x}, \boldsymbol{v}) \in \Gamma^-, \ t \in (t_n, t_{n+1}), \quad (9b) \\ F_{\mathrm{u}}(\boldsymbol{x}, \boldsymbol{v}, t_n^+) = G^n(\boldsymbol{x}, \boldsymbol{v}), & (\boldsymbol{x}, \boldsymbol{v}) \in X \times \mathbb{R}^d, \quad (9c) \end{cases}$$

$$\begin{cases} \partial_t F_{\mathrm{c}}(\boldsymbol{x}, \boldsymbol{v}, t) + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} F_{\mathrm{c}} + \dfrac{1}{\epsilon} F_{\mathrm{c}}(\boldsymbol{x}, \boldsymbol{v}, t) = \dfrac{1}{\epsilon} \mathcal{M}_F, & (\boldsymbol{x}, \boldsymbol{v}) \in X \times \mathbb{R}^d, \ t \in (t_n, t_{n+1}), \quad (10a) \\ F_{\mathrm{c}}(\boldsymbol{x}, \boldsymbol{v}, t) = 0, & (\boldsymbol{x}, \boldsymbol{v}) \in \Gamma^-, \ t \in (t_n, t_{n+1}), \quad (10b) \\ F_{\mathrm{c}}(\boldsymbol{x}, \boldsymbol{v}, t_n^+) = 0, & (\boldsymbol{x}, \boldsymbol{v}) \in X \times \mathbb{R}^d, \quad (10c) \end{cases}$$

where $G^0(\boldsymbol{x}, \boldsymbol{v})$ is given as in (1c) and for $n \geq 1$,

$$G^n(\boldsymbol{x}, \boldsymbol{v}) = F_{\mathrm{u}}(\boldsymbol{x}, \boldsymbol{v}, t_n^-) + F_{\mathrm{c}}(\boldsymbol{x}, \boldsymbol{v}, t_n^-), \tag{11}$$

and S is a source term. We assume the source S to be zero unless otherwise stated. The initial data and boundary conditions of the original system are inherited by the uncollided equation, while the

collided equation is assigned zero inflow and initial condition. For $s \geq 1$, the uncollided solution is reinitialized by adding together the current values of F_{u} and F_{c}. However, no splitting error is incurred by writing (1) as the sum of (9) and (10) until the equations are further discretized.

The basic idea of the hybrid approach is to discretize (9) with a high-resolution method in velocity with the expectation that collisions will allow (10) to be solved with a low-resolution method in velocity without significantly degrading the accuracy of the solution over a time step. However, in the limit of zero collisions, the uncollided equation is exact and the collided equation plays no role. At the end of the time step, the collided solution is reconstructed in the high-resolution space and then added to the uncollided solution in order to approximate G^n in (11). The combined solution is then used to reinitialize the uncollided equation at the next time step, while the initial condition for the collided equation is again reset to zero. To advance numerically within each time step, we employ a predictor-corrector method, the details of which are given in Section 3.2.

In the original hybrid formulation [1], the procedure summarized above resulted in a significant improvement in the time-to-solution for a fully implicit time-stepping scheme when applied to a linear, kinetic transport equation. In the current context, we consider hyperbolic time scales of the bulk flow which typically do not require fully implicit approaches, unless the mesh is highly unstructured. Instead, only (9) is solved fully implicitly in order to step over hyperbolic time-scale restrictions imposed by high-velocity components of the distribution. Meanwhile, (10) can be solved using an approximate model with a less restrictive time step.

3. Discretization of the hybrid method

For the purposes of this paper, we consider the one-dimensional case, which implies that the gas constant $\gamma = 3$. We set $X = (x_{\mathrm{L}}, x_{\mathrm{R}})$ and restrict the velocity domain to a bounded set $V = [-v_{\max}, v_{\max}]$.

3.1. Velocity discretization

Many velocity discretization methods can easily fit into the hybrid strategy outlined above. In the current paper, we choose to solve (9a) with a discrete velocity method [18, 46, 47, 30] and (10a) using the Euler equations as a moment-based velocity approximation.

3.1.1. Discrete velocity model (DVM) for the uncollided equation

Let $\{v_k\}_{k=1}^{N_v}$ and $\{\omega_k\}_{k=1}^{N_v}$ be the points and weights, respectively, of a one-dimensional, N_v-point Gauss-Legendre quadrature set scaled to a truncated velocity domain $[-v_{\max}, v_{\max}]$, and for each $k \in 1, \ldots, N_v$, let $\gamma_k^{\pm} = \{x \in X : \pm n(x) v_k > 0\}$. This discrete velocity grid discretization is also suitable for problems that include an acceleration term such as the Lorentz force in kinetic plasmas [48, 49]. Then the discrete velocity model for (9a) which governs $f_{\mathrm{u},k}(x,t) \simeq F_{\mathrm{u}}(x, v_k, t)$ is given by

$$\begin{cases} \partial_t f_{\mathrm{u},k}(x,t) + v_k \partial_x f_{\mathrm{u},k}(x,t) + \dfrac{1}{\epsilon} f_{\mathrm{u},k}(x,t) = 0, & x \in X, \quad t \in (t_n, t_{n+1}), \qquad (12a) \\ f_{\mathrm{u},k}(x,t) = G^-(x, v_k, t), & x \in \gamma_k^-, \quad t \in (t_n, t_{n+1}), \qquad (12b) \\ f_{\mathrm{u},k}(x, t_n^+) = g_k^n(x), & x \in X. \qquad (12c) \end{cases}$$

where $g_k^0(x) = G^0(x, v_k)$ and, for $n \geq 1$,

$$g_k^n(x) = f_{\mathrm{u},k}(x, t_n^-) + f_{\mathrm{c},k}(x, t_n^-), \qquad (13)$$

with f_c defined in (19) below. The approximate uncollided moments $\boldsymbol{q}_u \approx \boldsymbol{q}_{F_u}$ are obtained from $f_{u,k}$ via the quadrature formula

$$\boldsymbol{q}_u(x,t) = \sum_{k=1}^{N_v} \omega_k \boldsymbol{e}_k f_{u,k}(x,t), \tag{14}$$

where $\boldsymbol{e}_k = (1, v_k, \frac{1}{2}|v_k|^2)^\mathsf{T}$. In vectorized form, the discrete velocity equation (12a) and the initial condition (12c) can be written as [2]

$$\begin{cases} \partial_t \boldsymbol{f}_u(x,t) + \boldsymbol{V}\partial_x \boldsymbol{f}_u(x,t) + \dfrac{1}{\epsilon}\boldsymbol{f}_u(x,t) = 0, & x \in X, \quad t \in (t_n, t_{n+1}), \qquad (15a) \\ \boldsymbol{f}_u(x, t_n^+) = \boldsymbol{g}^n, & x \in X, \qquad (15b) \end{cases}$$

where $\boldsymbol{f}_u = (\boldsymbol{f}_{u,1}, \ldots, \boldsymbol{f}_{u,N_v})^\mathsf{T}$, $\boldsymbol{g}^n = (g_1^n, \ldots, g_{N_v}^n)^\mathsf{T}$, and $\boldsymbol{V} = \mathrm{diag}(v_1, \ldots, v_{N_v})$.

3.1.2. Moment equations for the collided equation

The exact equations for the mass, momentum, and energy moments of F_c are given by the quadrature formula

$$\partial_t \boldsymbol{q}_{F_c} + \partial_x \int_{\mathbb{R}} v\boldsymbol{e} F_c dv = \frac{1}{\epsilon}\boldsymbol{q}_{F_u}, \tag{16}$$

where $\boldsymbol{q}_{F_c} = \int_{\mathbb{R}} \boldsymbol{e} F_c dv$. We approximate the evolution of $\boldsymbol{q}_{F_c}$ by a function $\boldsymbol{q}_c$ that satisfies the Euler approximation to (16); that is,

$$\partial_t \boldsymbol{q}_c + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_c) = \frac{1}{\epsilon}\boldsymbol{q}_u, \quad \text{where} \quad \boldsymbol{\Phi}(\boldsymbol{q}_c) = \int_{\mathbb{R}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}_c) dv, \tag{17}$$

and $\boldsymbol{q}_u$ is computed according to (14). With initial and boundary conditions, the complete system is

$$\begin{cases} \partial_t \boldsymbol{q}_c(x,t) + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_c(x,t)) = \dfrac{1}{\epsilon}\boldsymbol{q}_u(x,t), & x \in X, \quad t \in (t_n, t_{n+1}), \qquad (18a) \\ \boldsymbol{\Phi}(\boldsymbol{q}_c(x,t)) = \displaystyle\int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}_c(x,t))dv, & x \in \partial X, \quad t \in (t_n, t_{n+1}), \qquad (18b) \\ \boldsymbol{q}_c(x, t_n^+) = 0, & x \in X, \qquad (18c) \end{cases}$$

and the approximate kinetic solution is

$$f_{c,k}(x,t) = \mathcal{E}(\boldsymbol{q}_c(x,t))(v_k). \tag{19}$$

Remark 1. *The boundary condition in* (18b) *is derived by separating the flux integral in* (16) *into incoming and outgoing data, using the kinetic data (which is zero) for the incoming data and the approximation $F_c \approx \mathcal{E}(\boldsymbol{q}_{F_c})$ for the outgoing data; that is, for $x \in \partial X$,*

$$\begin{aligned} \boldsymbol{\Phi}(\boldsymbol{q}_c(x,t)) &\approx \int_{\mathbb{R}} v\boldsymbol{e}(v) F_c(x,v,t) dv \\ &= \int_{\{v:n(x)v<0\}} v\boldsymbol{e}(v) F_c(x,v,t) dv + \int_{\{v:n(x)v>0\}} v\boldsymbol{e}(v) F_c(x,v,t) dv \\ &\approx \int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}_{F_c}(x,t))(v) dv \approx \int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}_c(x,t))(v) dv. \end{aligned} \tag{20}$$

There are other approaches to assigning boundary conditions, including asymptotic conditions derived via half-space problems; see for example [50].

[2] Unfortunately it is difficult to specify (12b) cleanly in vectorized form since γ_k^- depends on k.

3.1.3. Euler Limit

An important property of the hybrid is that it recovers (at least formally) the Euler equations in the limit $\epsilon \to 0$. To investigate this limit, we apply the quadrature rule in (14) to the discrete velocity equation in (12) and add the result to the moment equation for $\boldsymbol{q}_{\mathrm{c}}$ in (17). This gives the following conservation law for $\boldsymbol{q} = \boldsymbol{q}_{\mathrm{u}} + \boldsymbol{q}_{\mathrm{c}}$:

$$\begin{cases} \partial_t \boldsymbol{q}(x,t) + \partial_x \left(\sum_{k=1}^{N_v} \omega_k v_k \boldsymbol{e}_k f_{\mathrm{u},k}(x,t) \right) + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{c}}(x,t)) = 0, & x \in X, t \in (t_n, t_{n+1}), \quad (21a) \\ \boldsymbol{\Phi}(\boldsymbol{q}(x,t)) = \sum_{\{k:n(x)v_k<0\}} \omega_k v_k \boldsymbol{e}_k G^-(x,v_k) + \int_{\{v:n(x)v>0\}} v \boldsymbol{e} \mathcal{E}(\boldsymbol{q}_{\mathrm{c}}(x,t))(v) dv, & x \in \partial X, t \in (t_n, t_{n+1}), \quad (21b) \\ \boldsymbol{q}(x,t_n^+) = \sum_{k=1}^{N_v} \omega_k \boldsymbol{e}_k \left(f_{\mathrm{u},k}(x,t_n^-) + \mathcal{E}(\boldsymbol{q}_{\mathrm{c}}(x,t))(v_k) \right), & x \in X \ , \quad (21c) \end{cases}$$

which depends on ϵ via $\boldsymbol{f}_{\mathrm{u}}$ and $\boldsymbol{q}_{\mathrm{c}}$. To assess the limiting behavior of (21), we use the exact solution for (12). For each $x \in X$, $k \in \{1, \ldots, N_v\}$, and $n \geq 0$ fixed, let

$$t^* := t^*(x, v_k, t) = \min_{\tau > 0} \{\tau : x - v_k(t - \tau) \in \gamma_k^-\}. \quad (22)$$

Then for $t \in (t_n, t_{n+1})$ and a.e. $x \in X$,

$$f_{\mathrm{u},k}(x,t) = \begin{cases} e^{-\frac{t-t_n}{\epsilon}} g^n(x - v_k t_n, v_k), & t - t^* < t_n, \\ e^{-\frac{t-t^*}{\epsilon}} G^-(x - v_k t^*, v_k, t - t^*), & t - t^* > t_n. \end{cases} \quad (23)$$

In either case above, $f_{\mathrm{u},k} \to 0$ as $\epsilon \to 0$ in the interior of the domain. Thus the contribution of f_{u} to the flux in (21a) and (21c) vanishes. Furthermore, $\boldsymbol{q}_c \equiv \boldsymbol{q} - \boldsymbol{q}_u \to \boldsymbol{q}$ as $\epsilon \to 0$, so that (21) recovers the Euler system for $\boldsymbol{q}$; that is, in the limit as $\epsilon \to 0$,

$$\begin{cases} \partial_t \boldsymbol{q}(x,t) + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}(x,t)) = 0, & x \in X, t \in (t_n, t_{n+1}), \quad (24a) \\ \boldsymbol{\Phi}(\boldsymbol{q}(x,t)) = \sum_{\{k:n(x)v_k<0\}} \omega_k v_k \boldsymbol{e}_k G^-(x,v_k) + \int_{\{v:n(x)v>0\}} v \boldsymbol{e} \mathcal{E}(\boldsymbol{q}(x,t))(v) dv, & x \in \partial X, t \in (t_n, t_{n+1}), \quad (24b) \\ \boldsymbol{q}(x,t_n^+) = \begin{cases} \sum_{k=1}^{N_v} \omega_k \boldsymbol{e}_k G_0(x,v_k), & n = 0, \\ \sum_{k=1}^{N_v} \omega_k \boldsymbol{e}_k \mathcal{E}(\boldsymbol{q}(x,t))(v) & n > 1, \end{cases} & x \in X. \quad (24c) \end{cases}$$

The system (24) is a consistent discretization of the Euler equations. However,

1. There is an error introduced at the beginning of each time step due to the quadrature error in velocity. At time t^{n+1} the moment $\boldsymbol{q}(x, t_n^-)$, which is extracted at the end of the interval (t_n, t_{n+1}) from the solution of (24a), is replaced by the reinitialization value $\boldsymbol{q}(x, t_n^+) = \sum_{k=1}^{N_v} \omega_k \boldsymbol{e}_k \mathcal{E}(\boldsymbol{q}_{\mathrm{c}}(x,t))(v)$.
2. In practice (i.e. for finite ϵ), $\boldsymbol{f}_{\mathrm{u}}$ will decay to zero over a boundary layer of width ϵ. If this boundary layer is not well-resolved by the spatial and velocity discretization, the boundary condition (24b) will introduce errors into the overall solution. Such issues are well-known in the transport [51] and gas kinetic theory literature [50] and are not specifically induced by the hybrid method. In the numerical results of the current paper, solutions are computed on domains with equilibrium boundary conditions that are imposed far away from the interesting dynamics of the solution. In such situations, these boundary issues will not pollute the solution.

3.2. Time Discretization

In this section, we introduce two different time discretizations methods used to evolve the hybrid algorithm, formed by the components in (12) and (18), over a fixed time interval (t_n, t_{n+1}).

3.2.1. The hybrid BERK2 time discretization

The base method combines two backward Euler steps for (12) with a second-order predictor-corrector method for (18). This Backward Euler Runge-Kutta (BERK2) method for a single time step is given in Algorithm 1. The fully implicit treatment of (12) removes the time step restrictions otherwise induced by the advection operator. When combined with upwind DG discretizations in space, the resulting triangular linear system can be solved using sweep techniques [52, 53, 54] that are especially easy to implement on regular Cartesian grids. Moreover, because the discrete velocity components of $\boldsymbol{f}_{\mathrm{u}}$ are independent, the implementation is easy to parallelize. The explicit predictor-corrector method for (18) then induces a time step that scales inversely with Λ, the magnitude of the largest eigenvalue in the Euler system. Meanwhile, standard IMEX methods such as [42] require a time step that scales inversely with $v_{\max}$.

Algorithm 1 Hybrid BERK2 temporal update from t_n to t_{n+1}

Require: G^- ▷ Boundary data (used in (3) and (6))
Require: $\{v_k\}_{k=1}^{N_v}$ ▷ Discrete velocities
Require: $\{\boldsymbol{e}_k\}_{k=1}^{N_v}$ ▷ Discrete collision invariants
Require: $\{w_k\}_{k=1}^{N_v}$ ▷ Quadrature weights
Input: $\boldsymbol{g}^n$, $\boldsymbol{q}_c^n (= 0)$ ▷ Initial data

1: $\boldsymbol{E} \leftarrow [w_1\boldsymbol{e}_1 | \cdots | w_{N_v}\boldsymbol{e}_{N_v}]$
2: $\boldsymbol{f}_{\mathrm{u}}^n \leftarrow \boldsymbol{g}^n$
3: Solve $\dfrac{\boldsymbol{f}_{\mathrm{u}}^{n+1/2} - \boldsymbol{f}_{\mathrm{u}}^n}{\Delta t/2} + \boldsymbol{V}\partial_x \boldsymbol{f}_{\mathrm{u}}^{n+1/2} + \dfrac{1}{\epsilon}\boldsymbol{f}_{\mathrm{u}}^{n+1/2} = 0$ for $\boldsymbol{f}_{\mathrm{u}}^{n+1/2}$ ▷ Uncollided backward Euler
4: $\boldsymbol{q}_{\mathrm{u}}^{n+1/2} \leftarrow \boldsymbol{E}\boldsymbol{f}_{\mathrm{u}}^{n+1/2}$ ▷ Uncollided moments
5: Solve $\dfrac{\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}} - \boldsymbol{q}_{\mathrm{c}}^n}{\Delta t/2} + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{c}}^n) = \dfrac{1}{\epsilon}\boldsymbol{q}_{\mathrm{u}}^{n+\frac{1}{2}}$ for $\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}}$ ▷ Collided moment predictor
6: Solve $\dfrac{\boldsymbol{f}_{\mathrm{u}}^{n+1/2} - \boldsymbol{f}_{\mathrm{u}}^n}{\Delta t} + \boldsymbol{V}\partial_x \boldsymbol{f}_{\mathrm{u}}^{n+1} = -\dfrac{1}{\epsilon}\boldsymbol{f}_{\mathrm{u}}^{n+1}$ for $\boldsymbol{f}_{\mathrm{u}}^{n+1}$ ▷ Uncollided backward Euler
7: $\boldsymbol{q}_{\mathrm{u}}^{n+1} \leftarrow \boldsymbol{E}\boldsymbol{f}_{\mathrm{u}}^{n+1}$ ▷ Uncollided moments
8: Solve $\dfrac{\boldsymbol{q}_{\mathrm{c}}^{n+1} - \boldsymbol{q}_{\mathrm{c}}^n}{\Delta t} + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}}) = \dfrac{1}{\epsilon}\boldsymbol{q}_{\mathrm{u}}^{n+1}$ for $\boldsymbol{q}_{\mathrm{c}}^{n+1}$ ▷ Collided moment corrector
9: **for** $k = 1, \ldots, N_v$ **do**
10: $\quad f_{\mathrm{c},k}^{n+1} \leftarrow \mathcal{E}(\boldsymbol{q}_{\mathrm{c}}^{n+1})(v_k)$ ▷ Collided reconstruction
11: **end for**
12: $\boldsymbol{g}^{n+1} = \boldsymbol{f}_{\mathrm{u}}{}^{n+1} + \boldsymbol{f}_{\mathrm{c}}{}^{n+1}$ ▷ Uncollided relabeling
13: $\boldsymbol{q}_{\mathrm{c}}^{n+1} \leftarrow 0$ ▷ Collided relabeling

Output: $\boldsymbol{g}^{n+1}$, $\boldsymbol{q}_c^{n+1}$

3.2.2. The Euler limit

We investigate the formal limit of the BERK2 scheme when $\epsilon \to 0$. Similar to the time-continuous setting in Section 3.1.3, we take moments of f_{u} in Lines 3 and 6 of Algorithm 1 and add to Lines 5 and 8, respectively. Let $\boldsymbol{q}^\alpha = \boldsymbol{q}_u^\alpha + \boldsymbol{q}_c^\alpha$, for $\alpha \in \{n, n+1/2, n+1\}$. Then

$$\begin{cases} \dfrac{\boldsymbol{q}^{n+1/2}(x)-\boldsymbol{q}^n(x)}{\Delta t/2}+\displaystyle\sum_{k=1}^{N_v}\omega_k\boldsymbol{e}_k v_k\partial_x f_{\mathrm{u},k}^{n+1/2}(x)+\partial_x\boldsymbol{\Phi}(\boldsymbol{q}_c^n(x))=0, & x\in X, \quad (25a)\\ \boldsymbol{\Phi}(\boldsymbol{q}^n(x))=\displaystyle\sum_{\{k:n(x)v_k<0\}}\omega_k v_k\boldsymbol{e}_k G^-(x,v_k)+\int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}_c^n(x))(v)dv, & x\in\partial X, \quad (25b)\end{cases}$$

$$\begin{cases} \dfrac{\boldsymbol{q}^{n+1}(x)-\boldsymbol{q}^n(x)}{\Delta t}+\displaystyle\sum_{k=1}^{N_v}\omega_k\boldsymbol{e}_k v_k\partial_x f_{\mathrm{u},k}^{n+1}(x)+\partial_x\boldsymbol{\Phi}(\boldsymbol{q}_c^{n+1/2}(x))=0, & x\in X, \quad (26a)\\ \boldsymbol{\Phi}(\boldsymbol{q}^n(x))=\displaystyle\sum_{\{k:n(x)v_k<0\}}\omega_k v_k\boldsymbol{e}_k G^-(x,v_k)+\int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}_c^{n+1/2}(x))(v)dv, & x\in\partial X. \quad (26b)\end{cases}$$

The update equations for $\boldsymbol{f}_{\mathrm{u}}$ in Lines 3 and 6 of Algorithm 1 take the form

$$\frac{f_{\mathrm{u},k}^{\alpha}(x)-f_{\mathrm{u},k}^{n}(x)}{\tau}+v_k\partial_x f_{\mathrm{u},k}^{*}(x)=-\frac{1}{\epsilon}f_{\mathrm{u},k}^{*}(x), \tag{27}$$

for some $\tau>0$ and $\alpha\in\{n+1/2,n+1\}$. Let

$$s^*:=s^*(x,v_k)=\min_{s>0}\{s:x-v_k s\in\gamma_k^-\}\quad\text{and}\quad x_*:=x^*(x,v_k)=x-v_k s_*. \tag{28}$$

Then the solution of (27) is

$$f_{\mathrm{u},k}^{\alpha}(x)=\begin{cases} e^{-\frac{\sigma(x-x_*)}{v_k}}G^-(x_*,v_k,t_\alpha)+\frac{1}{\tau v_k}\int_{x_*}^{x}e^{-\frac{\sigma(x-\xi)}{v_k}}f_{\mathrm{u},k}^{n}(\xi)d\xi, & v_k\neq 0,\\ \frac{\epsilon}{\tau+\epsilon}f_{\mathrm{u},k}^{n}(x), & v_k=0,\end{cases} \tag{29}$$

where $\sigma=\frac{1}{\tau}+\frac{1}{\epsilon}$. Assume that $\sup_\xi|f_{\mathrm{u},k}^n(\xi)|\le M<\infty$. Then

$$\frac{1}{\tau v_k}\int_{x_*}^{x}e^{-\frac{\sigma(x-\xi)}{v_k}}f_{\mathrm{u},k}^{n}(\xi)d\xi\le\frac{M}{\tau v_k}\int_{x_*}^{x}e^{-\frac{\sigma(x-\xi)}{v_k}}d\xi=\frac{M}{\tau\sigma}\left(1-e^{-\frac{\sigma(x-x_*)}{v_k}}\right)\le\frac{M\epsilon}{1+\epsilon}, \tag{30}$$

whereas the other terms in (29) clearly decay as $\epsilon\to 0$. Thus if $f_{\mathrm{u},k}^n$ is bounded in ξ, it follows that $f_{\mathrm{u},k}^{\alpha}(x)\to 0$ and $\boldsymbol{q}_c^{\alpha}\to\boldsymbol{q}^{\alpha}$ as $\epsilon\to 0$ for any $x\in X$. As a result, in the limit $\epsilon\to 0$, (25) and (32) transition to a second-order predictor-corrector method for the Euler equations:

$$\begin{cases} \dfrac{\boldsymbol{q}^{n+1/2}-\boldsymbol{q}^n}{\Delta t/2}+\partial_x\boldsymbol{\Phi}(\boldsymbol{q}^n)=0, & x\in X, \quad (31a)\\ \boldsymbol{\Phi}(\boldsymbol{q}^n(x))=\displaystyle\sum_{\{k:n(x)v_k<0\}}\omega_k v_k\boldsymbol{e}_k G^-(x,v_k,t^{n+1/2})+\int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}^n(x))(v)dv, & x\in\partial X, \quad (31b)\end{cases}$$

$$\begin{cases} \dfrac{\boldsymbol{q}^{n+1}-\boldsymbol{q}^n}{\Delta t}+\partial_x\boldsymbol{\Phi}(\boldsymbol{q}^{n+1/2})=0, & x\in X, \quad (32a)\\ \boldsymbol{\Phi}(\boldsymbol{q}^n(x))=\displaystyle\sum_{\{k:n(x)v_k<0\}}\omega_k v_k\boldsymbol{e}_k G^-(x,v_k,t^{n+1})+\int_{\{v:n(x)v>0\}} v\boldsymbol{e}\mathcal{E}(\boldsymbol{q}^{n+1/2}(x))(v)dv, & x\in\partial X. \quad (32b)\end{cases}$$

3.2.3. Correction and conservation fix

A modified version of the BERK2 time discretization is given in Algorithm 1. This modified version includes (i) a conservation fix to fix the deficiency caused by consistency errors in the discrete velocity quadrature and (ii) a relabeling step that solves the original BGK equation using a Maxwellian constructed from the hybrid strategy. The details of these modifications are given in Algorithm 2.

Conservation of mass, momentum, and energy in the BGK equation is a consequence of (4). To correct consistency errors that violate (4), we adopt the technique first introduced in [55]. Given a discrete velocity vector $\boldsymbol{f} = [f_1, \ldots, f_{N_v}]^{\mathsf{T}} \in \mathbb{R}^{N_v}$, let $\boldsymbol{E} = [w_1\boldsymbol{e}_1|\cdots|w_{N_v}\boldsymbol{e}_{N_v}]$ and let

$$\boldsymbol{q}_{\boldsymbol{f}} = \sum_{k=1}^{N_v} w_k \boldsymbol{e}_k f_k = \boldsymbol{E}\boldsymbol{f} \in \mathbb{R}^3 \tag{33}$$

be the moments associated to $\boldsymbol{f}$. In order to match a moment $\widetilde{\boldsymbol{q}} \in \mathbb{R}^3$ where $\widetilde{\boldsymbol{q}} \neq \boldsymbol{q}_f$, the corrected discrete velocity vector $\widetilde{\boldsymbol{f}}$ is given by

$$\widetilde{\boldsymbol{f}} = \underset{\boldsymbol{g}\in\mathbb{R}^{N_v}}{\operatorname{argmin}}\{\|\boldsymbol{f}-\boldsymbol{g}\|_2^2 : \boldsymbol{E}\boldsymbol{g} = \widetilde{\boldsymbol{q}}\}. \tag{34}$$

The solution to this optimization problem satisfies $\boldsymbol{q}_{\widetilde{\boldsymbol{f}}} \equiv \boldsymbol{E}\boldsymbol{f} = \widetilde{\boldsymbol{q}}$ and is given by

$$\widetilde{\boldsymbol{f}} = \boldsymbol{f} + \boldsymbol{E}^{\mathsf{T}}(\boldsymbol{E}\boldsymbol{E}^{\mathsf{T}})^{-1}(\widetilde{\boldsymbol{q}} - \boldsymbol{E}\boldsymbol{f}). \tag{35}$$

The relabeling step constructs a discrete Mawellian from the moments $\boldsymbol{q}^{n+1} = \boldsymbol{q}_{\mathrm{u}}^{n+1} + \boldsymbol{q}_{\mathrm{c}}^{n+1}$. A final implicit update of the BGK equation uses this discrete Maxwellian as a fixed source. Because there is a fixed source, this update can be efficiently completed without iteration. The implicit update has been done through the backward differentiation formula (BDF2) discretization. This approach to relabeling was introduced in [56] and tends to yield more accurate answers because errors in the moments that are induced by the hybrid approximation are smaller than the errors in the kinetic distribution.

3.3. Spatial discretization

Both (12) and (17) are discretized with a discontinuous Galerkin method [57, 58]. Since this method is by now well-known, the presentation here will be naturally brief. For simplicity, we discretize the time continuous equations.

Given $N_x + 1$ grid points $x_{\mathrm{L}} = x_{1/2} < x_{3/2} < \cdots < x_{N_x+1/2} = x_{\mathrm{R}}$, let $S_i = (x_{i-1/2}, x_{i+1/2})$ be a cell of width $h_i = x_{i+1/2} - x_{i-1/2}$ and center $\frac{1}{2}(x_{i+1/2} + x_{i-1/2})$. Let $\mathbb{P}^{N_p}$ be the set of all polynomials with maximum polynomial degree p, and let $h = \max_i h_i$. Define the broken finite element space

$$\mathcal{W}_{N_p}^h := \left\{ w^h \in L^2\left[x_{\mathrm{L}}, x_{\mathrm{R}}\right] :\, w^h\big|_{S_i} \in \mathbb{P}^{N_p} \quad \forall i \in \{1, \ldots, N_x\} \right\}. \tag{36}$$

A basis for $\mathcal{W}_{N_p}^h$ is formed by the functions

$$w_{i,\ell}(x) = p_\ell\left(\frac{x - x_i}{h_i/2}\right), \qquad i \in \{1, \ldots, N_x\}, \quad \ell \in \{1, \ldots, N_p\}, \tag{37}$$

where $\{p_\ell\}_{\ell=1}^{N_p}$ is a set of degree $\leq p$ Lagrange polynomials on $[-1, 1]$. These polynomials are constructed from the Gauss-Lobatto points $\{\xi_\ell\}_{\ell=1}^{N_p}$ on $[-1, 1]$:

$$p_\ell\left(\xi\right) = \prod_{\substack{\ell'=1 \\ \ell'\neq\ell}}^{N_p} \frac{(\xi - \xi_{\ell'})}{(\xi_\ell - \xi_{\ell'})}, \qquad \ell \in \{1, \ldots, N_p\}. \tag{38}$$

Algorithm 2 Hybrid BERK2 temporal update from t_n to t_{n+1} with [redacted] and conservation fix

Require: G^- ▷ Boundary data (used in (3) and (6))
Require: $\{v_k\}_{k=1}^{N_v}$ ▷ Discrete velocities
Require: $\{\boldsymbol{e}_k\}_{k=1}^{N_v}$ ▷ Discrete collision invariants
Require: $\{w_k\}_{k=1}^{N_v}$ ▷ Quadrature weights
Input: $\boldsymbol{g}^n$, $\boldsymbol{q}_c^n (= 0)$ ▷ Initial data

$\boldsymbol{E} \leftarrow [w_1\boldsymbol{e}_1|\cdots|w_{N_v}\boldsymbol{e}_{N_v}]$
$\boldsymbol{f}_{\mathrm{u}}^n \leftarrow \boldsymbol{g}^n$
Solve $\dfrac{\boldsymbol{f}_{\mathrm{u}}^{n+1/2} - \boldsymbol{f}_{\mathrm{u}}^n}{\Delta t/2} + \boldsymbol{V}\partial_x \boldsymbol{f}_{\mathrm{u}}^{n+1/2} = -\dfrac{1}{\epsilon}\boldsymbol{f}_{\mathrm{u}}^{n+1/2}$ for $\boldsymbol{f}_{\mathrm{u}}^{n+1/2}$ ▷ Uncollided backward Euler
$\boldsymbol{q}_{\mathrm{u}}^{n+\frac{1}{2}} \leftarrow \boldsymbol{E}\boldsymbol{f}_{\mathrm{u}}$ ▷ Uncollided moments
Solve $\dfrac{\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}} - \boldsymbol{q}_{\mathrm{c}}^n}{\Delta t/2} + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{c}}^n) = \frac{1}{\epsilon}\boldsymbol{q}_{\mathrm{u}}^{n+\frac{1}{2}}$ for $\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}}$
$\boldsymbol{f}_{\mathrm{c}}^{n+\frac{1}{2}} \leftarrow \mathcal{E}(\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}})$ ▷ Discrete collided distribution
$\boldsymbol{g}^{n+\frac{1}{2}} \leftarrow \boldsymbol{f}_{\mathrm{u}}^{n+\frac{1}{2}} + \boldsymbol{f}_{\mathrm{c}}^{n+\frac{1}{2}}$ ▷ midstep for BDF2 correction
Solve $\dfrac{\boldsymbol{f}_{\mathrm{u}}^{n+1} - \boldsymbol{f}_{\mathrm{u}}^n}{\Delta t} + \boldsymbol{V}\partial_x \boldsymbol{f}_{\mathrm{u}}^{n+1} = -\dfrac{1}{\epsilon}\boldsymbol{f}_{\mathrm{u}}^{n+1}$ for $\boldsymbol{f}_{\mathrm{u}}^{n+1}$ ▷ Uncollided backward Euler
$\boldsymbol{q}_{\mathrm{u}}^{n+1} \leftarrow \boldsymbol{E}\boldsymbol{f}_{\mathrm{u}}$ ▷ Uncollided moments
Solve $\dfrac{\boldsymbol{q}_{\mathrm{c}}^{n+1} - \boldsymbol{q}_{\mathrm{c}}^n}{\Delta t} + \partial_x \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{c}}^{n+\frac{1}{2}}) = \dfrac{1}{\epsilon}\boldsymbol{q}_{\mathrm{u}}^{n+1}$ for $\boldsymbol{q}_{\mathrm{c}}^{n+1}$ ▷ Collided moment corrector
$\boldsymbol{q}^{n+1} = \boldsymbol{q}_{\mathrm{u}}^{n+1} + \boldsymbol{q}_{\mathrm{c}}^{n+1}$ ▷ Full moments
for $k = 1, \ldots, N_v$ **do**
    $M_k^{n+1} \leftarrow \mathcal{E}(\boldsymbol{q}^{n+1})(v_k)$ ▷ Discrete Maxwellian
end for
$\boldsymbol{M}^{n+1} \leftarrow [M_1^{n+1}, \ldots, M_{N_v}^{n+1}]^{\mathsf{T}}$
$\boldsymbol{M}^{n+1} \leftarrow \boldsymbol{M}^{n+1} + \boldsymbol{E}^{\mathsf{T}}(\boldsymbol{E}\boldsymbol{E}^{\mathsf{T}})^{-1}(\boldsymbol{q}^{n+1} - \boldsymbol{E}\boldsymbol{M}^{n+1})$ ▷ Conservation fix
[redacted] ▷ BDF2 correction
$\boldsymbol{f}^{n+1} \leftarrow \boldsymbol{f}^{n+1} + \boldsymbol{E}^{\mathsf{T}}(\boldsymbol{E}\boldsymbol{E}^{\mathsf{T}})^{-1}(\boldsymbol{q}^{n+1} - \boldsymbol{E}\boldsymbol{f}^{n+1})$ ▷ Conservation fix
$\boldsymbol{g}^{n+1} \leftarrow \boldsymbol{f}^{n+1}$
$\boldsymbol{q}_c^{n+1} \leftarrow 0$

Output: $\boldsymbol{g}^{n+1}$, $\boldsymbol{q}_c^{n+1}$

3.3.1. Uncollided equation

To discretize (12), we seek for each $k \in \{1, \ldots, N_v\}$, a function

$$f_{\mathrm{u},k}^h(x,t) = \sum_{i=1}^{N_x}\sum_{\ell=1}^{N_p} f_{k,i,\ell}(t) w_{i,\ell}(x), \tag{39}$$

such that for all $i \in \{1, \ldots, N_x\}$ and $\ell \in \{1, \ldots, N_p\}$,

$$\begin{aligned} \partial_t \int_{S_i} f^h_{\mathrm{u},k}(x,t) w_{i,\ell}\, dx + v_k \left(f^{h,*}_{\mathrm{u},k}(x_{i+1/2},t) w_{i,\ell}(x^-_{i+1/2}) - f^{h,*}_{\mathrm{u},k}(x_{i-1/2},t) w_{i,\ell}(x^+_{i-1/2}) \right) \\ - v_k \int_{S_i} f^h_{\mathrm{u},k}(x,t) w'_{i,\ell}(x)\, dx + \frac{1}{\varepsilon} \int_{S_i} f^h_{\mathrm{u},k}(x,t) w_{i,\ell}(x)\, dx = 0, \end{aligned} \tag{40}$$

where $f^{h,*}_{\mathrm{u},k}$ is the upwind trace:

$$f^{h,*}_{\mathrm{u},k}(x_{i+1/2},t) = \begin{cases} f^h_{\mathrm{u},k}(x^-_{i+1/2},t), & v_k > 0, \quad (41a) \\ f^h_{\mathrm{u},k}(x^+_{i+1/2},t), & v_k < 0. \quad (41b) \end{cases}$$

In terms of $\boldsymbol{f}_{k,i} = (f_{k,i,1}, \ldots, f_{k,i,N_p})^{\mathsf{T}}$, (40) takes the form, for $i \in \{1, \ldots, N_x\}$,

$$\begin{cases} \dfrac{h_i}{2} \boldsymbol{J} \dfrac{d\boldsymbol{f}_{k,i}}{dt}(t) + v_k \left[\boldsymbol{L}_1 \boldsymbol{f}_{k,i}(t) - \boldsymbol{L}_2 \boldsymbol{f}_{k,i-1}(t) - \boldsymbol{K} \boldsymbol{f}_{k,i}(t) \right] + \dfrac{h_i}{2\varepsilon} \boldsymbol{J}_i \boldsymbol{f}_{k,i}(t) = 0, \quad \nu_k > 0, & (42a) \\ \dfrac{h_i}{2} \boldsymbol{J} \dfrac{d\boldsymbol{f}_{k,i}}{dt}(t) + v_k \left[\boldsymbol{L}_3 \boldsymbol{f}_{k,i+1}(t) - \boldsymbol{L}_4 \boldsymbol{f}_{k,i}(t) - \boldsymbol{K} \boldsymbol{f}_{k,i}(t) \right] + \dfrac{h_i}{2\varepsilon} \boldsymbol{J} \boldsymbol{f}_{k,i}(t) = 0, \quad \nu_k < 0, & (42b) \end{cases}$$

where $\boldsymbol{J} \in R^{N_p \times N_p}$ and $\boldsymbol{K} \in R^{N_p \times N_p}$ have components

$$\boldsymbol{J}_{\ell,\ell'} = \int_{-1}^{1} \Phi_\ell(\xi)\, \Phi_{\ell'}(\xi)\, \mathrm{d}\xi \qquad \text{and} \qquad \boldsymbol{K}_{\ell,\ell'} = \int_{-1}^{1} \Phi_\ell(\xi)\, \Phi'_{\ell'}(\xi)\, \mathrm{d}\xi, \tag{43}$$

and the matrices $\boldsymbol{L}_r$ for $r \in \{1, 2, 3, 4\}$, have components

$$(\boldsymbol{L}_1)_{\ell,\ell'} = \Phi_\ell(1)\Phi_{\ell'}(1) = \delta_{\ell,N}\delta_{\ell',N}, \qquad (\boldsymbol{L}_2)_{\ell,\ell'} = \Phi_\ell(-1)\Phi_{\ell'}(1) = \delta_{\ell,1}\delta_{\ell',N}, \tag{44a}$$

$$(\boldsymbol{L}_3)_{\ell,\ell'} = \Phi_\ell(1)\Phi_{\ell'}(-1) = \delta_{\ell,N}\delta_{\ell',1}, \qquad (\boldsymbol{L}_4)_{\ell,\ell'} = \Phi_\ell(-1)\Phi_{\ell'}(-1) = \delta_{\ell,1}\delta_{\ell',1}, \tag{44b}$$

with δ the usual Kronecker delta.

The implementation of (42) requires the specification of ghost values $\boldsymbol{f}_{k,0}$ when $\nu_k > 0$ and $\boldsymbol{f}_{k,N_x+1}$ when $\nu_k < 0$. In the numerical examples of Section 4, either periodic or Dirichlet boundary conditions are used. For the periodic case, we set

$$\begin{cases} \boldsymbol{f}_{k,0}(t) = \boldsymbol{f}_{k,0}, & \nu_k > 0, \quad (45) \\ \boldsymbol{f}_{k,N_x+1}(t) = \boldsymbol{f}_{k,0}(t), & \nu_k < 0. \quad (46) \end{cases}$$

For the Dirichlet case, we set the nodal values equal to the boundary condition on each end:

$$\begin{cases} f_{k,0,\ell}(t) = G^-(x_{\mathrm{L}}, \nu_k, t), & \nu_k > 0, \quad \ell = 1, \ldots, N_p, \quad (47) \\ f_{k,N_x+1,\ell}(t) = G^-(x_{\mathrm{R}}, \nu_k, t), & \nu_k < 0, \quad \ell = 1, \ldots, N_p. \quad (48) \end{cases}$$

In practice, the equations in (42) are solved using backward Euler or BDF2. In either case, an effective steady-state problem is created which can be solved efficiently using sweeps that update $\boldsymbol{f}_{k,i}$ using the information from $\boldsymbol{f}_{k,i-1}$ when $\nu_k > 0$ and information from $\boldsymbol{f}_{k,i+1}$ when $\nu_k < 0$. The sweeps are independent across discrete velocities and require only the inversion of an $N_p \times N_p$ per discrete velocity to update the degrees of freedom in each spatial cell. For problems with incoming boundary data given, only one sweep across the spatial mesh is needed for each discrete velocity. For problems with periodic or reflecting boundaries, several sweep iterations are needed to reach convergence. For the numerical examples in Section 4 with periodic boundary conditions, we update the ghost cells after each sweep and observe that only one or two iterations are required for convergence [52, 53, 54].

3.3.2. Collided equation

To discretize (17), we seek for each a vector-valued function

$$\boldsymbol{q}_{\mathrm{c}}^h = \sum_{i=1}^{N_x} \sum_{\ell=1}^{N_p} \boldsymbol{q}_{i,\ell}(t) w_{i,\ell}(x), \tag{49}$$

such that for all $i \in \{1, \dots, N_x\}$ and $\ell \in \{1, \dots, N_p\}$,

$$\begin{aligned} \partial_t \int_{S_i} \boldsymbol{q}_{\mathrm{c}}^h(x,t) w_{i,\ell}(x) dx + \boldsymbol{\Phi}^*(\boldsymbol{q}_{\mathrm{c}}^h(x_{i+1/2}^-, t), (\boldsymbol{q}_{\mathrm{c}}^h(x_{i+1/2}^+, t)) w_{i,\ell}(x_{i+1/2}^-) \\ - \boldsymbol{\Phi}^*(\boldsymbol{q}_{\mathrm{c}}^h(x_{i-1/2}^-, t), \boldsymbol{q}_{\mathrm{c}}^h(x_{i-1/2}^+, t)) w_{i,\ell}(x_{i-1/2}^+) \\ + \int_{S_i} \boldsymbol{\Phi}(\boldsymbol{q}_{\mathrm{c}}^h(x,t)) w_{i,\ell}'(x)\, dx = \frac{1}{\varepsilon} \int_{S_i} \boldsymbol{q}_{\mathrm{u}}^h(x,t) w_{i,\ell}(x)\, dx, \end{aligned} \tag{50}$$

where

$$\boldsymbol{q}_{\mathrm{u}}^h(x,t) = \sum_{k=1}^{N_v} \omega_k \boldsymbol{e}_k f_{\mathrm{u,k}}^h(x,t), \tag{51}$$

and the numerical flux $\boldsymbol{\Phi}^*$ is the local Lax-Friedrichs flux:

$$\boldsymbol{\Phi}^*(\boldsymbol{q}) = \boldsymbol{\Phi}^*(\boldsymbol{q}^+, \boldsymbol{q}^-) = \frac{1}{2} \left(\boldsymbol{\Phi}(\boldsymbol{q}^+) + \boldsymbol{\Phi}(\boldsymbol{q}^-) - \max_{\boldsymbol{q} \in \{\boldsymbol{q}^-, \boldsymbol{q}^+\}} |\lambda(\boldsymbol{\Phi}'(\boldsymbol{q}))| (\boldsymbol{q}^+ - \boldsymbol{q}^-) \right). \tag{52}$$

The flux Jacobian matrix

$$\boldsymbol{\Phi}'(\boldsymbol{q}) = \begin{pmatrix} 0 & 1 & 0 \\ \frac{(\gamma-3)}{2} \frac{q_2^2}{q_1^2} & (3-\gamma)\frac{q_2}{q_1} & (\gamma-1) \\ -\gamma \frac{q_2 q_3}{{q_1}^2} + (\gamma-1)\frac{{q_2}^3}{{q_1}^3} & \gamma \frac{q_3}{q_1} - \frac{3}{2}(\gamma-1)\frac{{q_2}^2}{{q_1}^2} & \gamma \frac{q_2}{q_1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ -3\frac{q_2 q_3}{{q_1}^2} + 2\frac{{q_2}^3}{{q_1}^3} & 3\frac{q_3}{q_1} - 3\frac{{q_2}^2}{{q_1}^2} & 3\frac{q_2}{q_1} \end{pmatrix} \tag{53}$$

has eigenvalues $\lambda = \{u, u-c, u+c\}$, with

$$u = \frac{q_2}{q_1} \quad \text{and} \quad c = \sqrt{\frac{\gamma(2q_1 q_3 - {q_2}^2)}{{q_1}^2}} = \sqrt{\gamma\theta} = \sqrt{3\theta}. \tag{54}$$

Thus

$$\Lambda = \max_{\boldsymbol{q} \in \{\boldsymbol{q}^-, \boldsymbol{q}^+\}} |\lambda(\boldsymbol{\Phi}'(\boldsymbol{q}))| = \max_{\boldsymbol{q} \in \{\boldsymbol{q}^-, \boldsymbol{q}^+\}} \big(\max(|u|, |u-c|, |u+c|) \big). \tag{55}$$

Once $\boldsymbol{q}_{\mathrm{c}}^h$ is computed, the associated Maxwellian $\mathcal{E}(\boldsymbol{q}_{\mathrm{c}}^h)$ can be computed at each DG node.

In order to prevent spurious oscillations, we post-process the moments after each update $\boldsymbol{q}_{\mathrm{c}}^h$ using the total variation bounded (TVB) slope limiter introduced in [59]. This limiter requires the specification of a TVB parameter M which is chosen to be $M = 20$. However, it is only applied to certain test problems; see details in the next section.

4. Numerical Results

In this section, we present numerical results. For all tests, we compare the BERK2 hybrid methods to the solutions obtained using the IMEX-DG scheme. Both implementations utilize the same discrete velocity quadratures for velocity discretization. Additionally, the IMEX-DG scheme relies on the stiffly accurate IMEX-SSP2(3,2,2) scheme [60] for 2nd order and the IMEX-ARS(4,4,3)

scheme [61] for the 3rd-order in the temporal domain. See [60, 61] for more details about these IMEX schemes. Note that DGk refers to the k-th order DG scheme, contrasting with the conventional $(k+1)$-th order scheme.

The time step for BERK2-DG is $\Delta t = C\Lambda^{-1}\Delta x$, where

$$\Lambda \equiv \max_{0\le i\le N_x} \max_{\boldsymbol{q}\in\{\boldsymbol{q}(x^-_{i+1/2}),\boldsymbol{q}(x^+_{i+1/2})\}} |\lambda(\boldsymbol{\Phi}'(\boldsymbol{q}))| \tag{56}$$

approximates the maximum wave speed associated to the flux matrix $\Phi'(\boldsymbol{q})$ for (18a) over the spatial domain X. Meanwhile, the time step for the IMEX-DG is $\Delta t = Cv^{-1}_{\max}\Delta x$, where $v_{\max}$ is chosen to ensure that the amount of mass lost in the distribution function is within acceptable bounds. For the tests in Sections 4.1 and 4.2, the CFL constant C for the BERK2-DG4 hybrid is given in Table 1. In the accuracy test case (Section 4.2), the constants are chosen to be smaller to postpone saturation in convergence due to temporal error. In Sections 4.3-4.6 a DG3 scheme is used for all spatial discretizations and the CFL constant C is given in Table 2.

DG slope limiters are not used for the tests in Sections 4.1 and 4.2, which focus on smooth solutions. In the remaining tests, the TVB slope limiter [59] is used in the simulation of the collided moments $\boldsymbol{q}_{\rm c}$ and for simulations of the Euler equations.

In all the numerical tests, the initial distribution function f is given by a Maxwellian, as defined in (2). That is, $f(x,v,t=0) = \mathcal{E}(\boldsymbol{q}(x,t=0))(v)$, where $\boldsymbol{q}(x,t=0)$ is determined from the variables $\rho(x,t=0)$, $u(x,t=0)$, and $p(x,t=0)$.

Reference solutions for the Euler equations are computed with the DoGPack software package [62] using a 4th-order DG scheme with $N_x = 3000$ spatial points and a 4th-order low-storage strong stability Runge-Kutta method [63] for time integration. Reference solutions for the BGK equations are computed using the 3rd-order IMEX3 scheme with refined grid points. To be more specific, the reference solution named IMEX3H is used in Sections 4.3-4.5 and in Section 4.6. This is a highly resolved IMEX3-DG3 solution with a grid size of $N_x = N_v = 1000$ in Sections 4.3-4.5 and $N_x = 800$ and $N_v = 1100$ in Section 4.6.

The purpose of each numerical test is as follows: The asymptotic test in Section 4.1 is to assess convergence to the Euler limit as $\epsilon \to 0$. The accuracy test in Section 4.2 is to investigate the accuracy of the BERK2-DG method for different values of ϵ. In particular, it is expected that high-resolution, high-order schemes will experience temporal accuracy saturation; understanding where this saturation occurs is important for practical applications. The tests in Sections 4.3-4.5 are standard benchmarks from gas dynamics, meant to assess how the BERK2-DG scheme captures shocks, contacts, and rarefactions. Finally, the test in Section 4.6 is designed to emphasize the main advantage of the hybrid BERK2 scheme which is the efficiency afforded by the relaxed time step restriction when $\Lambda < v_{\max}$.

Table 1: CFL constant C, used in sections 4.1 and 4.2.

section	CFL constant	BERK2-DG2	BERK2-DG3	BERK2-DG4
4.1	C	-	-	0.1
4.2	C	0.2	0.1	0.05

4.1. Asymptotic test

In this test, we confirm numerically the Euler limit established in Sections 3.1.3 and 3.2.2. The domain is $X \times V = [-\pi,\pi]\times[-7,7]$ and the boundary conditions are periodic. The initial data is

Table 2: Comparisons of the number of time steps, CFL constant C, maximum velocity $v_{\max}$, and the wave speed Λ in each of the benchmark problems from Sections 4.3-4.6. The IMEX3L is an IMEX-ARS(4,4,3), IMEX2L is an IMEX-SSP2(3,2,2), BERK2L is a BERK2 solution, and BERK2LS is BERK2 solutions with reduced CFL constants. Since the wave-speed Λ changes dynamically in each time-step, the time-step size for the BERK2 methods also changes, especially for the gas-injection problem. Note that the ratio $v_{\max}/\Lambda$ indicates the efficiency of BERK2 scheme in general.

section	time-stepping	# of time steps			C	$v_{\max}$	Λ	$v_{\max}/\Lambda$
		$\epsilon = 1$	10^{-2}	10^{-6}				
4.3 (Sod)	IMEX3L	429	429	429	0.14	6.0	1.732-2.934	2.045-3.464
	IMEX2L	300	300	300	0.2			
	BERK2L	134	137	138	0.2			
	BERK2LS	274	274	274	0.1			
4.4 (Lax)	IMEX3L	536	536	536	0.14	15.0	5.575-8.110	1.850-2.691
	IMEX2L	375	375	375	0.2			
	BERK2L	140	140	189	0.2			
	BERK2LS	280	280	280	0.1			
4.5 (Shu-Osher)	IMEX3L	1800	1800	1800	0.14	14.0	6.206-8.085	1.732-2.256
	IMEX2L	1260	1260	1260	0.2			
	BERK2L	559	583	686	0.2			
	BERK2LS	1148	1148	1148	0.1			
4.6 (gas-injection)	IMEX3L	1000	1000	1000	0.1	110.0	0.548-8.549	12.867-200.730
	IMEX2L	1000	1000	1000	0.1			
	BERK2L	14	27	49	0.1			
	BERK2LS	61	117	210	0.025			

$G^0(x,v) = \mathcal{E}(\boldsymbol{q}(x,0))(v)$, where $\boldsymbol{q}$ is determined from the variables

$$\rho(x,t=0) = 1 + 0.2\sin 10x, \qquad u(x,t=0) = 1, \qquad p(x,t=0) = 1, \tag{57}$$

and $\epsilon = 10^{-12}$. Here, the Euler solution is used as the reference solution with the initial condition $\rho_0(x) = 1+0.2\sin 10x$. The results for the mass density at time $t = 0.1$ are given in Table 3. All three versions—standard BERK2, BERK2 with BDF2 correction step, and BERK2 with BDF2 correction and the conservation fix—exhibit the 2nd-order temporal convergence rate that is expected based on the limiting equations in (31) and (32).

4.2. Accuracy test

In this test, we consider the smooth initial condition from [42] to test the order of accuracy of the hybrid for arbitrary various ϵ. The domain is $X \times V = [-\pi, \pi] \times [-7, 7]$, and the boundary conditions are periodic. The initial data is $G^0(x,v) = \mathcal{E}(\boldsymbol{q}(x,0))(v)$, where $\boldsymbol{q}$ is determined from the variables

$$\rho(x,t=0) = 1 + 0.2\sin 10x, \qquad u(x,t=0) = 1, \qquad p(x,t=0) = 1. \tag{58}$$

The final time is $t = 0.1$. A velocity grid with $N_v = 100$ is used throughout. Because the analytic solution is not available, the solution with the half mesh size is used as the reference solution. We consider the L^2-errors for ρ, given by $E_2^h := \|\rho^h - \rho^{h/2}\|_{L^2}$. The convergence order ν is defined by $\nu = \log\left(E_2^h / E_2^{h/2}\right)/\log 2$.

The results are given in Table 4. While the standard BERK2 method exhibits degradation in convergence order, especially for $\epsilon \in \{1, 0.1, 0.01\}$, the order improves significantly when the BDF2 correction step and the conservation fix are employed.

Table 3: Asymptotic test from Section 4.1. The L_2 errors and order of accuracy for the density at $t = 0.1$ with $\epsilon = 10^{-12}$. The spatial discretization is DG4. Left: BERK2 without the correction step. Middle: BERK2 with BDF2 correction step. Right BERK2 with both BDF2 correction step and conservation fix. The bulk velocity u is excluded since $u = 1$. The influences of the BDF2 correction and the conservation fix on accuracy are very small in the asymptotic test.

			No correction		BDF2 correction		BDF2 Correction + conservation fix	
	ϵ	N_x	$L^2(\rho - \rho_{\text{Euler}})$	order	$L^2(\rho - \rho_{\text{Euler}})$	order	$L^2(\rho - \rho_{\text{Euler}})$	order
ρ	10^{-12}	8	2.376e-01	-	2.376e-01	-	2.376e-01	-
		16	3.455e-02	2.781	3.455e-02	2.781	3.455e-02	2.781
		32	2.612e-03	3.726	2.612e-03	3.726	2.612e-03	3.726
		64	1.078e-04	4.599	1.078e-04	4.599	1.078e-04	4.599
		128	1.427e-05	2.917	1.425e-05	2.919	1.425e-05	2.919
		256	3.261e-06	2.129	3.242e-06	2.136	3.242e-06	2.136
		512	8.136e-07	2.003	7.960e-07	2.026	7.960e-07	2.026
θ	10^{-12}	8	2.542e-01	-	2.542e-01	-	2.542e-01	-
		16	4.384e-02	2.536	4.384e-02	2.536	4.384e-02	2.536
		32	4.088e-03	3.423	4.088e-03	3.423	4.088e-03	3.423
		64	3.189e-04	3.680	3.189e-04	3.680	3.189e-04	3.680
		128	6.083e-05	2.390	6.083e-05	2.390	6.083e-05	2.390
		256	1.467e-05	2.052	1.467e-05	2.052	1.467e-05	2.052
		512	3.487e-06	2.073	3.487e-06	2.073	3.487e-06	2.073

Table 4: Accuracy test from Section 4.2. The L_2-errors E_h^2 and convergence order ν of the density for various ϵ. The time step size is $\Delta t = C\Delta x \Lambda^{-1}$, where $C = 0.2$ for DG2, $C = 0.1$ for DG3, and $C = 0.05$ for DG4. The final time is $t = 0.1$. BERK2 without the correction step is on the left and BERK2 with both the BDF2 correction step and the conservation fix is on the right.

		No correction						BDF2 correction + conservation fix					
		DG2		DG3		DG4		DG2		DG3		DG4	
ϵ	N_x	E_h^2	ν	E_h^2	ν	E_h^2	ν	E_h^2	ν	E_h^2	ν	E_h^2	ν
1	16	2.56e-02	-	1.13e-03	-	6.34e-05	-	2.61e-02	-	1.16e-03	-	6.40e-05	-
	32	1.04e-02	1.3	2.03e-04	2.5	4.84e-06	3.7	1.02e-02	1.3	1.93e-04	2.6	4.64e-06	3.8
	64	3.76e-03	1.5	2.91e-05	2.8	3.39e-07	3.8	3.29e-03	1.6	2.46e-05	3.0	3.00e-07	3.9
	128	1.16e-03	1.7	4.38e-06	2.7	2.42e-08	3.8	8.34e-04	2.0	3.05e-06	3.0	1.89e-08	4.0
	256	3.82e-04	1.6	7.34e-07	2.6	1.88e-09	3.7	2.03e-04	2.0	3.80e-07	3.0	1.18e-09	4.0
10^{-1}	16	2.56e-02	-	1.13e-03	-	6.34e-05	-	2.61e-02	-	1.16e-03	-	6.40e-05	-
	32	1.03e-02	1.3	1.99e-04	2.5	4.85e-06	3.7	1.02e-02	1.3	1.95e-04	2.6	4.70e-06	3.8
	64	3.64e-03	1.5	2.81e-05	2.8	3.35e-07	3.9	3.31e-03	1.6	2.48e-05	3.0	3.04e-07	4.0
	128	1.11e-03	1.7	4.23e-06	2.7	2.37e-08	3.8	8.41e-04	2.0	3.07e-06	3.0	1.91e-08	4.0
	256	3.72e-04	1.6	7.18e-07	2.6	1.83e-09	3.7	2.03e-04	2.0	3.78e-07	3.0	1.18e-09	4.0
10^{-2}	16	2.56e-02	-	1.13e-03	-	6.34e-05	-	2.61e-02	-	1.16e-03	-	6.40e-05	-
	32	1.08e-02	1.2	1.89e-04	2.6	5.07e-06	3.6	1.03e-02	1.3	1.94e-04	2.6	4.94e-06	3.7
	64	3.77e-03	1.5	3.23e-05	2.5	3.89e-07	3.7	3.37e-03	1.6	2.54e-05	2.9	3.20e-07	4.0
	128	1.75e-03	1.1	7.91e-06	2.0	3.77e-08	3.4	9.17e-04	1.9	3.27e-06	3.0	2.01e-08	4.0
	256	9.78e-04	0.8	2.09e-06	1.9	4.42e-09	3.1	2.37e-04	2.0	4.10e-07	3.0	1.24e-09	4.0
10^{-6}	16	2.56e-02	-	1.13e-03	-	6.34e-05	-	2.61e-02	-	1.16e-03	-	6.40e-05	-
	32	1.28e-02	1.0	1.79e-04	2.7	5.17e-06	3.6	1.28e-02	1.0	1.79e-04	2.7	5.17e-06	3.6
	64	3.49e-03	1.9	2.34e-05	2.9	3.26e-07	4.0	3.49e-03	1.9	2.34e-05	2.9	3.26e-07	4.0
	128	8.67e-04	2.0	3.19e-06	2.9	2.21e-08	3.9	8.67e-04	2.0	3.19e-06	2.9	2.21e-08	3.9
	256	2.25e-04	1.9	6.75e-07	2.2	4.14e-09	2.4	2.25e-04	1.9	6.73e-07	2.2	4.09e-09	2.4

Table 5: Comparisons of CPU time and speed-up for each problem. I3L, I2L, B2L, and B2LS denote IMEX3L, IMEX2L, BERK2L, and BERK2LS, respectively.

		CPU time (s)				Speed-up (B2L)		Speed-up (B2LS)	
	ϵ	I3L	I2L	B2L	B2LS	vs I3L	vs I2L	vs I3L	vs I2L
Sod	1	19.87	8.64	1.06	2.29	18.75	8.15	8.68	3.77
	10^{-2}	20.94	8.71	1.11	2.35	18.86	7.85	8.91	3.71
	10^{-6}	18.71	8.57	1.05	2.31	17.82	8.16	8.10	3.71
Lax	1	25.56	11.91	1.74	3.71	14.69	6.84	6.89	3.21
	10^{-2}	24.99	12.16	1.74	3.77	14.36	6.99	6.63	3.23
	10^{-6}	25.21	11.37	1.71	3.82	14.74	6.65	6.60	2.98
Shu-Osher	1	292.33	120.93	29.77	62.1	9.82	4.06	4.71	1.95
	10^{-2}	287.11	121.02	29.50	61.2	9.73	4.10	4.69	1.98
	10^{-6}	291.56	120.36	29.92	61.7	9.74	4.02	4.73	1.95
gas-injection	1	697.96	399.83	3.09	10.33	225.88	129.39	67.57	38.71
	10^{-2}	699.51	404.01	6.09	18.84	114.86	66.34	37.13	21.44
	10^{-6}	700.4	405.88	10.21	33.85	68.60	39.75	20.69	11.99

4.3. Sod shock tube problem

The Sod shock tube problem [64] is used to determine how well a numerical solver captures fundamental flow features. The initial conditions are $G^0(x,v) = \mathcal{E}(\boldsymbol{q}(x,0))(v)$, where $\boldsymbol{q}$ is determined from the variables

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1,0,1), & 0 \le x \le 0.5, \\ (0.125,0,0.1), & 0.5 < x \le 1. \end{cases} \tag{59}$$

The spatial domain $[0,1]$ is discretized with $N_x = 100$ spatial cells and the velocity domain $[-6,6]$ is discretized with a $N_v = 100$ point Gauss-Legendre quadrature set, scaled to fit the velocity domain. The boundary conditions are chosen to coincide with the initial constant moments at both ends of the spatial domain. The final time is $t = 0.1$. Numerical solutions are computed with the hybrid BERK2 time discretization and a 3rd-order DG discretization in space.

Figure 1 includes five solutions. EulerH is the Euler solution that is obtained by the Euler solver from DoGPack [62] with $N_x = 3000$. IMEX3H is a highly resolved IMEX3-DG solution with $N_x = N_v = 1000$; it is used as a reference. IMEX3L is an IMEX3-DG solution with $N_x = N_v = 100$. IMEX2L is an IMEX3-DG solution with $N_x = N_v = 100$. BERK2L is the hybrid BERK2 solution with $N_x = N_v = 100$. All schemes are plotted with the CFL constant C given in Table 2. The differences in wave speeds, time-step size, and the size of the velocity domain are presented in Table 2 to facilitate an effective comparison of efficiency for all of the test problems in Sections 4.3-4.6.

We plot three differences to compare the accuracy and CPU time for each method of the IMEX-DG and BERK2 methods: (i) IMEX3H-IMEX3L, (ii) IMEX3H-IMEX2L, and (iii) IMEX3H-BERK2L. From the results in Figure 1, we conclude that the BERK2L scheme provides a good approximation within a smaller CPU time than IMEX2L and IMEX3L. In this particular velocity domain setting, BERK2L shows the CPU time corresponding to approximately 1/8 of the CPU time for IMEX2L and 1/18 of the CPU time for IMEX3L as shown in Table 5. The only significant differences in accuracy occur when $\epsilon = 0.01$. In that instance, the IMEX-DG schemes provide better accuracy. For this reason, we have included a plot of the three solutions together to highlight that these differences occur primarily near the sharp transitions in the solution profile. The fourth row

(BERK2LS) of Figure 1 emphasizes the BERK2L solutions with a smaller CFL constant of $C = 0.1$, which yields comparable accuracy with less computation time than the IMEX solutions. Within the BERK2LS configuration, the CPU time is roughly 1/3 that of IMEX2L and approximately 1/8 the time of IMEX3L.

4.4. Lax shock tube problem

The Lax shock tube problem [65] is another Riemann problem that is similar to the previous Sod problem, but uses a non-zero initial bulk velocity. The initial conditions are $G^0(x, v) = \mathcal{E}(\boldsymbol{q}(x, 0))(v)$, where $\boldsymbol{q}$ is determined from the variables

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (0.445, 0.698, 3.528), & -0.5 \le x \le 0.5, \\ (0.5, 0, 0.571), & 0.5 < x \le 1.5. \end{cases} \tag{60}$$

The spatial domain $[-0.5, 1.5]$ is discretized with $N_x = 100$ spatial cells and the velocity domain $[-15, 15]$ is discretized with $N_v = 100$ points. Numerical results are obtained through the 3rd order nodal DG scheme up to time $t = 0.1$. The boundary conditions are set to be the same way as the previous Sod shock tube problem.

Similar to the Sod problem, Figure 2 includes five solutions. EulerH is the Euler solution that is obtained by the Euler solver from DoGPack [62] with $N_x = 3000$. IMEX3H is a highly resolved IMEX3-DG solution with $N_x = N_v = 1000$ that will be used as a reference solution. IMEX2L is an IMEX3-DG solution with $N_x = N_v = 100$. IMEX2L is an IMEX3-DG solution with $N_x = N_v = 100$. BERK2L is the hybrid BERK2 solution with $N_x = N_v = 100$. All schemes are plotted with the CFL constant C given in Table 2.

We plot three differences to compare the accuracy and CPU time for each method of the IMEX-DG and BERK2 methods: (i) IMEX3H-IMEX3L, (ii) IMEX3H-IMEX2L, and (iii) IMEX3H-BERK2L. The results in Figure 2 are similar to those from the Sod problem. Again, we conclude that the BERK2L scheme provides a good approximation within a smaller CPU time than IMEX2L and IMEX3L. In this particular velocity domain setting, BERK2L shows the CPU time corresponding to approximately 1/6 of the CPU time for IMEX2L and 1/14 of the CPU time for IMEX3L as shown in Table 5. Again, the only significant differences in accuracy occur when $\epsilon = 0.01$. In that instance, the IMEX-DG schemes provide better accuracy. To fix this, we have included the fourth row in which the BERK2LS solutions are presented with a reduced CFL constant of $C = 0.1$, resulting in comparable accuracy with less computation time compared to the IMEX solutions. In the context of the BERK2LS framework, the CPU time is estimated to be about 1/3 of the time required for IMEX2L and roughly 1/6 of the time required for IMEX3L.

4.5. Shu-Osher problem

The Shu-Osher problem [66] is a simulation of a shock-turbulence interaction where a shock propagates into a perturbed density field, thus it can be used to determine the ability of a numerical solver to capture a shock wave, its interaction with an unsteady density field, and the waves propagating downstream of the shock [67]. The initial conditions are $G^0(x, v) = \mathcal{E}(\boldsymbol{q}(x, 0))(v)$, where $\boldsymbol{q}$ is determined from the variables

$$(\rho(x,0), u(x,0), p(x,0)) = \begin{cases} (1.756757, 2.005122, 10.333333), & x \le -4, \\ (1 + 0.2\sin 5x, 0, 1), & x > -4. \end{cases} \tag{61}$$

The initial shock is located at $x = -4$. In this problem, more spatial and discrete velocity grids are used due to the increased size of the spatial and velocity domain. The spatial domain $[-10, 10]$

and the velocity domain $[-14, 14]$ are discretized with $N_x = 200$ uniform grid points and $N_v = 200$ Gauss-Legendre quadrature points, respectively. We compute the solutions with the 3rd order DG scheme up to time $t = 1.8$. The same boundary conditions are used as before.

Similar to the Sod and Lax problems, Figure 3 includes five solutions. EulerH is the Euler solution that is obtained by the Euler solver from DoGPack [62] with $N_x = 3000$. IMEX3H is a highly resolved IMEX3-DG solution with $N_x = N_v = 1000$ that will be used as a reference solution. All IMEX2L, IMEX3L, and BERK2L solutions are computed with $N_x = N_v = 200$. All schemes are plotted with the CFL constant C given in Table 2.

We plot three differences to compare the accuracy and CPU time for each method of the IMEX-DG and BERK2 methods in the same order as before. The results in Figure 3 are similar to those from the Sod problem. Again, we conclude that the BERK2L scheme provides a good approximation within a smaller CPU time than IMEX2L and IMEX3L. In this particular velocity domain setting, BERK2L shows the CPU time corresponding to 1/4 of the CPU time for IMEX2L and 1/9 of the CPU time for IMEX3L as shown in Table 5. In this problem, we do not observe significant differences in accuracy found in the previous problems when $\epsilon = 0.01$. However, we still include BERK2LS in the 4th row to demonstrate the efficiency of the BERK2 scheme. Operating within the BERK2LS environment, the computational time used is nearly half of what is used for IMEX2L and close to 1/4 of the time consumed by IMEX3L.

4.6. Gas injection problems

The gas injection problem was designed to emphasize the benefit of the BERK2 method when $v_{\max}$ is very large relative to Λ, i.e., the ratio $v_{\max}/\Lambda$ is large. The initial condition is

$$G^0(x, v) = \mathcal{E}(\boldsymbol{q}_1(x, 0))(v), \tag{62}$$

where $\boldsymbol{q}_1$ is determined from the variables

$$\rho_1(x, 0) = 1, \qquad u_1(x, 0) = 0, \qquad \theta_1(x, 0) = 0.1. \tag{63}$$

The source is

$$S(x, v, t) = \eta(x)\mathcal{E}(\boldsymbol{q_2}(x, 0))(v), \tag{64}$$

where $\boldsymbol{q}_2$ is determined from the variables

$$\rho_2(x, 0) = 0.01, \qquad u_2(x, 0) = 100, \qquad \theta_2(x, 0) = 100, \tag{65}$$

and η is

$$\eta(x) = c_0 \exp\left(\frac{-(x - x_0)^2}{2\sigma^2}\right), \quad x_0 = 0.5, \;\; \sigma = 0.1. \tag{66}$$

Here, c_0 is a normalization constant, defined such that $\int_0^1 \eta(x)\, dx = 1$. This problem requires a large velocity domain by construction. The spatial domain and the velocity domain are chosen to be $[-3, 19]$ and $[-v_{\max}, v_{\max}]$, respectively, where $v_{\max} = 110$. We present our results for BERK2L in Figures 4-6 and BERK2LS in Figures 7-9.

The first rows of Figures 4-9 use the IMEX3H solutions as a reference. The remaining rows of Figure 4-9 show the ρ, u, and θ of the gas injection problem, and the leftmost column tells the computational time for the IMEX3L, IMEX2L, and hybrid BERK2L schemes. Setting $u_2 = 100$ greatly magnifies the computation time differences between BERK2L and IMEX2,3L schemes due to the increased size of the velocity domain with $v_{\max} = 110$. We choose more discrete velocity

grids with $N_v = 1000$ to correctly capture the velocity integration with minimal quadrature error. The spatial grid is chosen to be uniform with $N_x = 200$. All schemes are plotted with the CFL constant C given in Table 2.

Table 5 shows CPU time for the gas injection problem, and it is clear to see the time difference between the hybrid BERK2L and the rest. On average, for the $u_2 = 100$ case, the speed-up is about 225.88X, 114.86X, and 68.60X versus IMEX3L and 129.39X, 66.34X, and 39.75X versus IMEX2L when $\epsilon = 1.0$, $\epsilon = 10^{-2}$, and $\epsilon = 10^{-6}$, respectively. The corresponding solution profiles are shown in Figure 4-6. The figure format in this section has been altered from earlier sections to enhance the visualization of the shock wave front. It is clear from these plots that the BERK2L solution has smeared out the high-speed structure due to stepping over the fastest time scales in the problem when $\epsilon = 1$ and $\epsilon = 10^{-2}$. This is the price to be paid for the efficiency and stability of the implicit calculation of the uncollided solution. The required number of time-step in BERK2 is significantly smaller than IMEX schemes as it is shown in Table 5, i.e., BERK2 can handle a much larger time-step size than IMEX schemes due to the relaxed CFL condition in this problem.

For BERK2LS, which uses a small CFL constant $C = 0.025$, the speed-up is 67.57X, 37.13X, and 20.69X versus IMEX3L and 38.71X, 21.44X, and 11.99X versus IMEX2L when $\epsilon = 1.0$, $\epsilon = 10^{-2}$, and $\epsilon = 10^{-6}$, respectively. The corresponding solution profiles are shown in Figure 7-9. In this case, the profile accuracy (BERK2LS) is more comparable to IMEX3H. Note that, for this particular problem, the explicit advection of IMEX schemes handle the bump-on-tail more accurately, though the errors are small in BERK2LS.

5. Conclusion

Our hybrid BERK2 scheme for the uncollided-collided decomposition works uniformly for all Knudsen numbers. The comparisons of the scheme with non-splitting IMEX schemes indicate that the hybrid BERK2 scheme is more effective than the IMEX scheme with a coarse spatial discretization especially when the ratio of $v_{\max}$ to Λ is large.

Fully implicit treatment for the uncollided equation can be easily achieved by the linearity of the uncollided equation. This approach can remove the velocity CFL restriction existing in IMEX time discretizations, which is especially beneficial for problems that have a large ratio of $v_{\max}$ to Λ. The obtained uncollided solution is updated as a source term in the collided equation. This approach makes the collided equation possible to be solved semi-implicitly with explicit calculations, which allows much less restrictive CFL conditions. The biggest potentiality of this scheme is that the kinetic equations can be coupled to existing high-order hydrodynamic codes without suffering from very restrictive CFL conditions. The splitting error in our hybrid scheme can be further reduced by the method, e.g., the integral deferred correction scheme [36]. Reducing the splitting error of the hybrid method is another interesting topic that can be studied in the future.

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.

Acknowledgements

This research is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, as part of their Applied Mathematics Research Program. The work was performed at the Oak Ridge National Laboratory, which is

managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). In addition, the work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2023-00220762), and in part by the Yonsei University Research Fund(Post Doc. Researcher Supporting Program) of 2023 (project no.: 2023-12-0018).

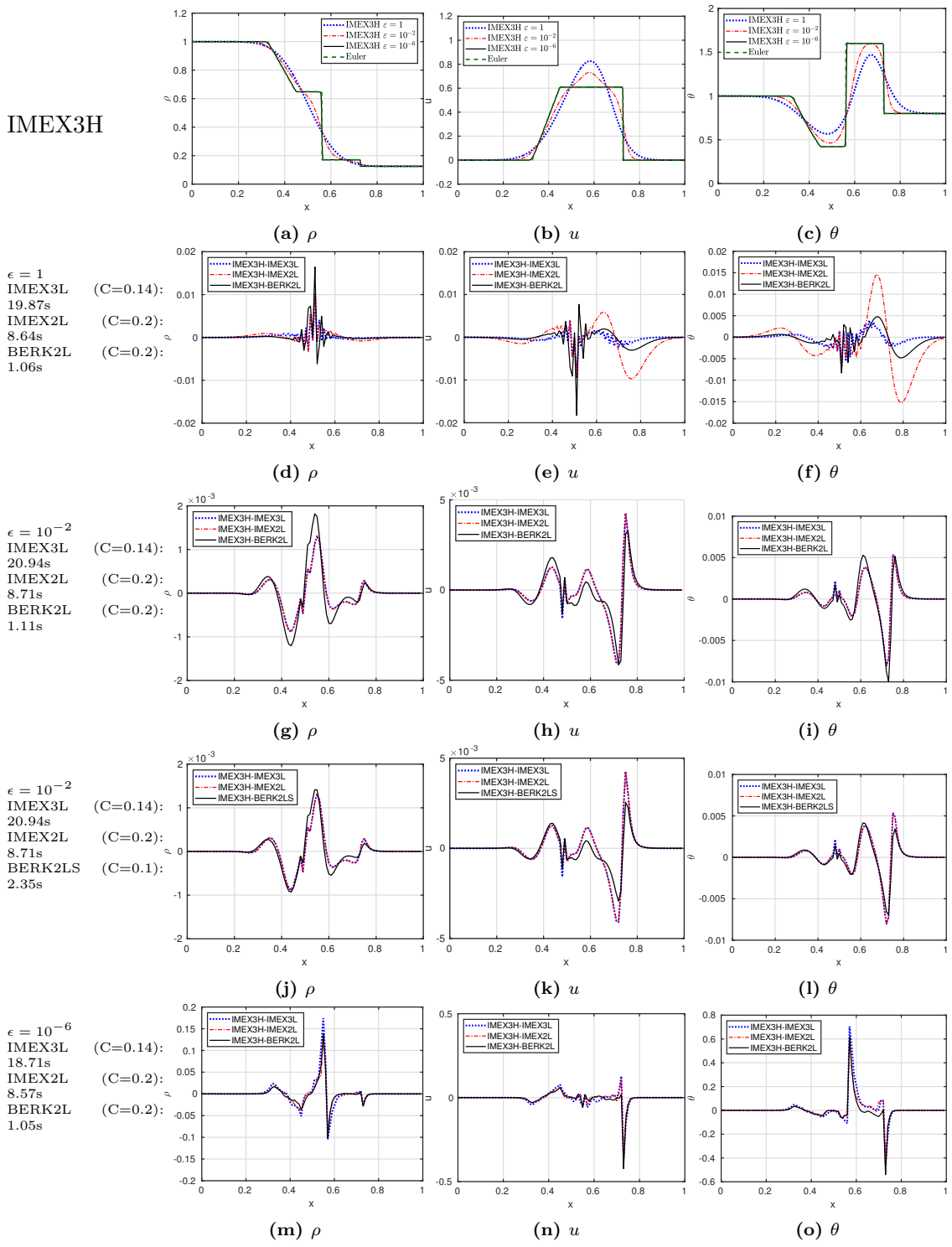


Figure 1: Comparison of numerical solutions for the Sod problem at t=0.1 with various ϵ. Row 1: Euler and IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for each unknown (ρ, u, θ), we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2L. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 3, but with a smaller CFL constant $C = 0.1$. Row 5: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the leftmost column. (Parameters for each method - Euler: $N_x = 3000$, $C = 0.1$; IMEX3H: $N_x = N_v = 1000$, $C = 0.14$; IMEX3L: $N_x = N_v = 100$, $C = 0.14$; IMEX2L: $N_x = N_v = 100$, $C = 0.2$; BERK2L: $N_x = N_v = 100$, $C = 0.2$)

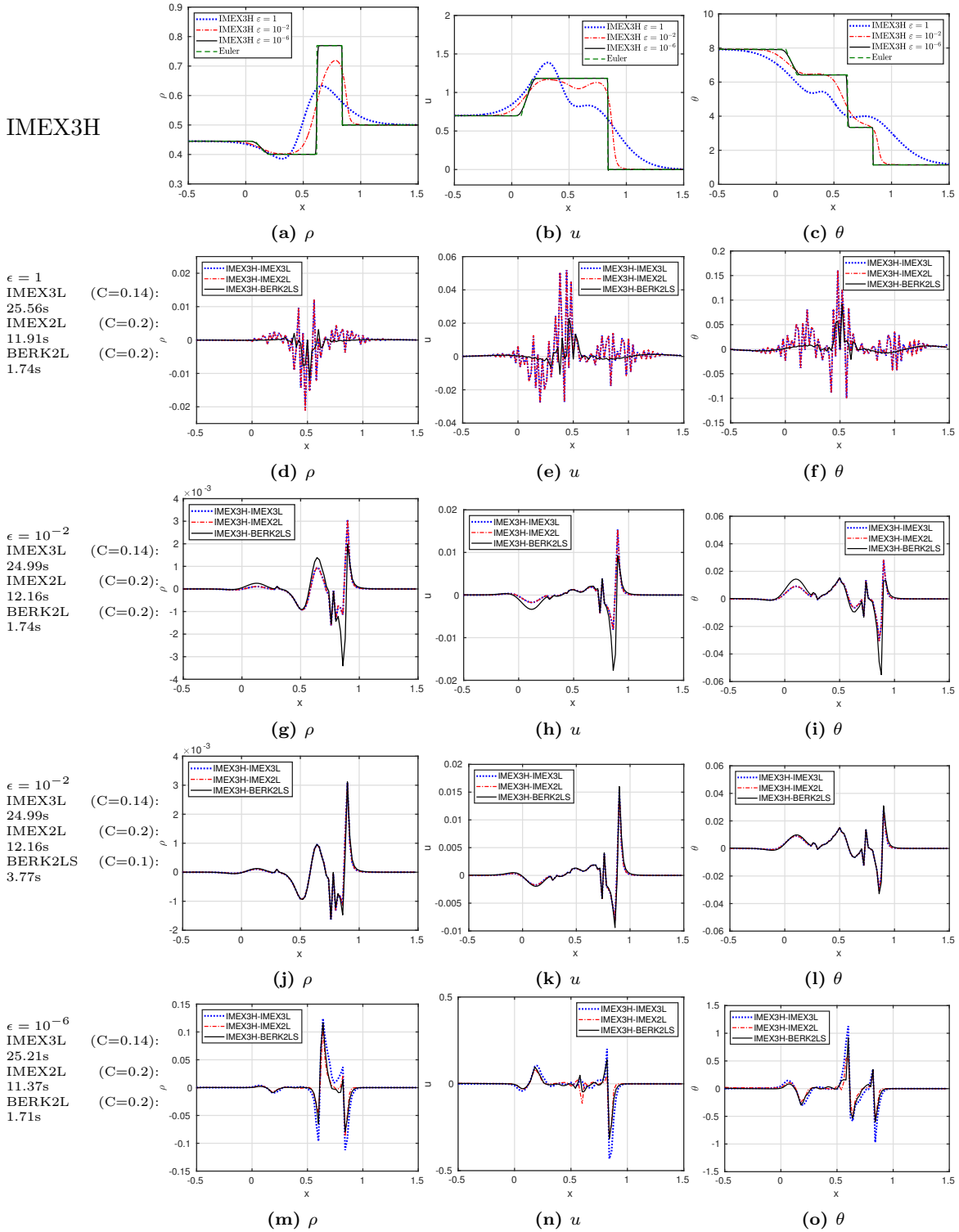


Figure 2: Comparison of numerical solutions for the Lax problem at t=0.1 with various ϵ. Row 1: Euler and IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for each unknown (ρ, u, θ), we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2L. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. Row 5: Comparisons of low-resolution solution for $\epsilon = 0.01$. (Parameters for each method - Euler: $N_x = 3000$, $C = 0.1$; IMEX3H: $N_x = N_v = 1000$, $C = 0.14$; IMEX3L: $N_x = N_v = 100$, $C = 0.14$; IMEX2L: $N_x = N_v = 100$, $C = 0.2$; BERK2L: $N_x = N_v = 100$, $C = 0.2$)

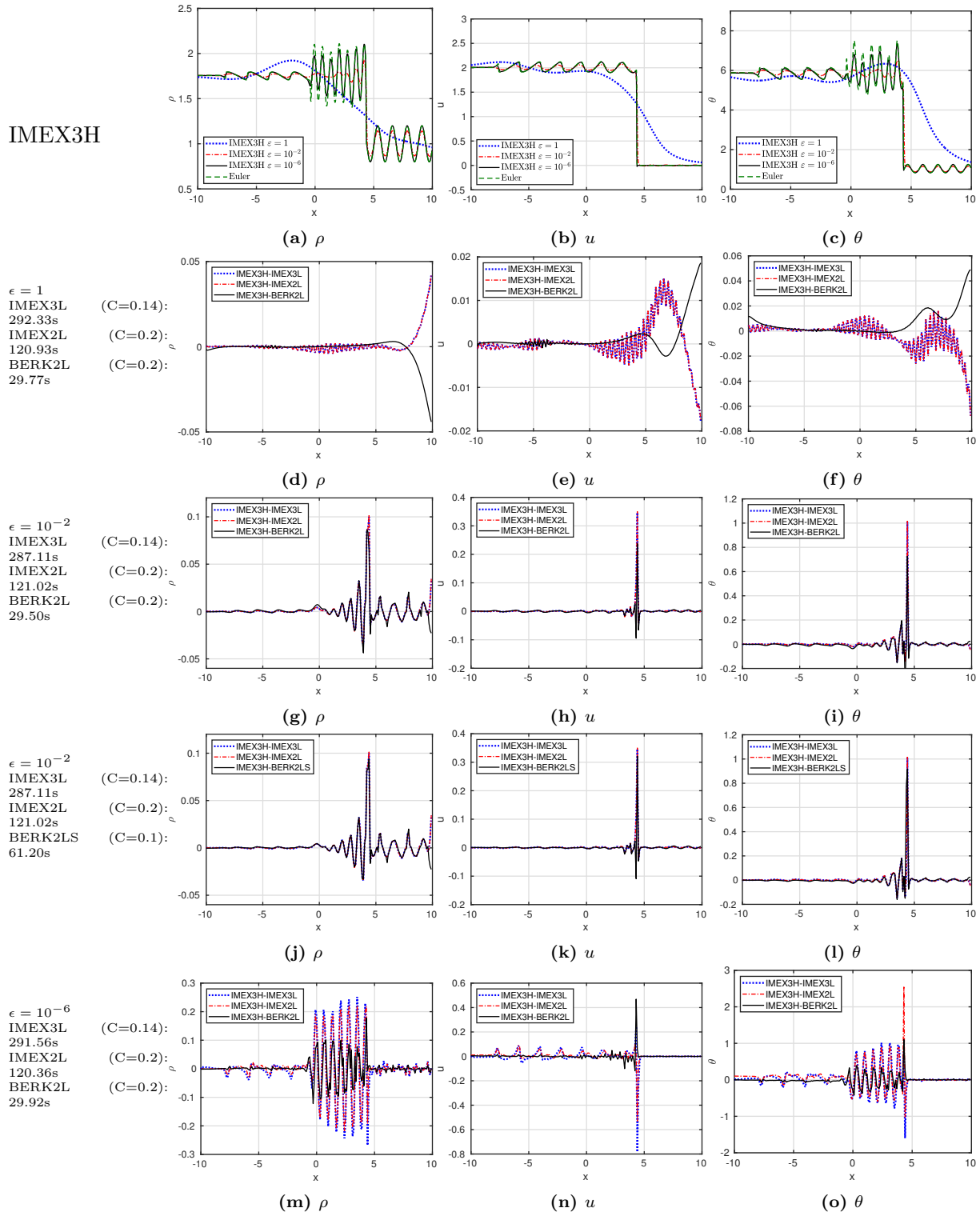
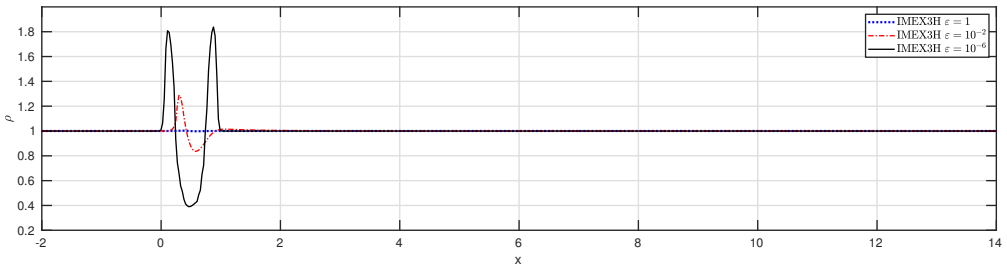


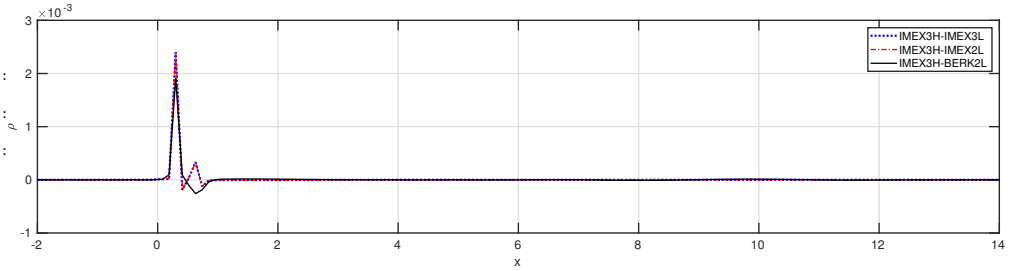
Figure 3: Comparison of numerical solutions for the Shu-Osher problem at t=1.8 with various ϵ. Row 1: Euler and IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for each unknown (ρ, u, θ), we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2L. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. Row 5: Comparisons of low-resolution solution for $\epsilon = 0.01$. (Parameters for each method - Euler: $N_x = 3000$, $C = 0.1$; IMEX3H: $N_x = N_v = 1000$, $C = 0.14$; IMEX3L: $N_x = N_v = 200$, $C = 0.14$; IMEX2L: $N_x = N_v = 200$, $C = 0.2$; BERK2L: $N_x = N_v = 200$, $C = 0.2$)

IMEX3H



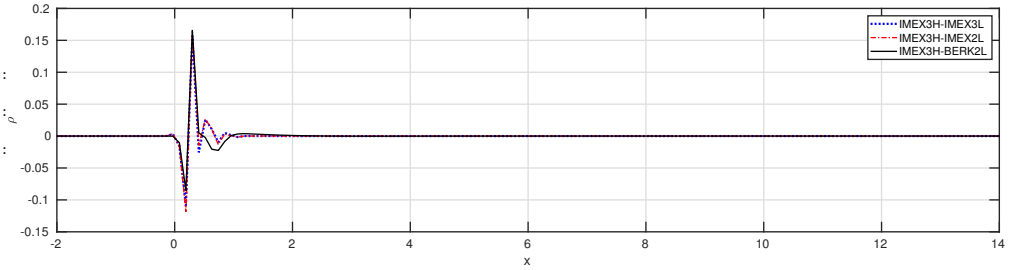
(a) ρ

$\epsilon = 1$
IMEX3L (C=0.1): 697.96s
IMEX2L (C=0.1): 399.83s
BERK2L (C=0.1): 3.09s



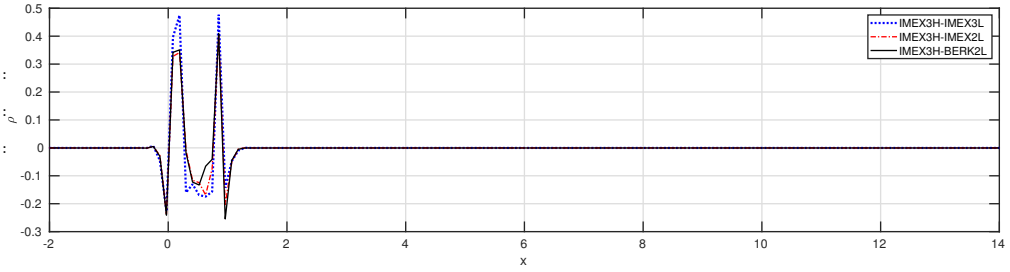
(b) ρ

$\epsilon = 10^{-2}$
IMEX3L (C=0.1): 699.51s
IMEX2L (C=0.1): 404.01s
BERK2L (C=0.1): 6.09s



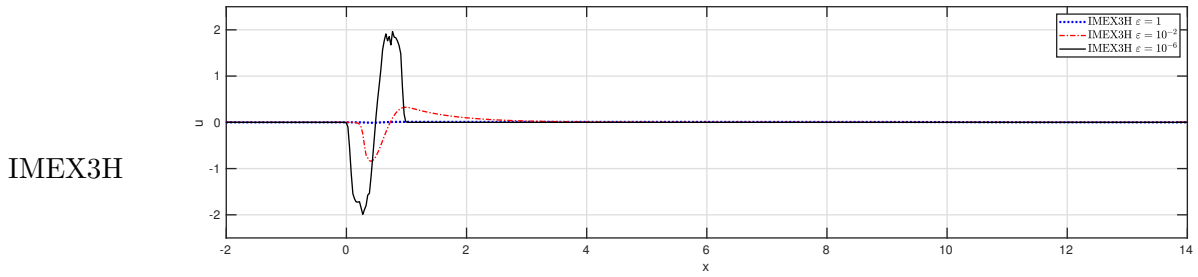
(c) ρ

$\epsilon = 10^{-6}$
IMEX3L (C=0.1): 700.40s
IMEX2L (C=0.1): 405.88s
BERK2L (C=0.1): 10.21s

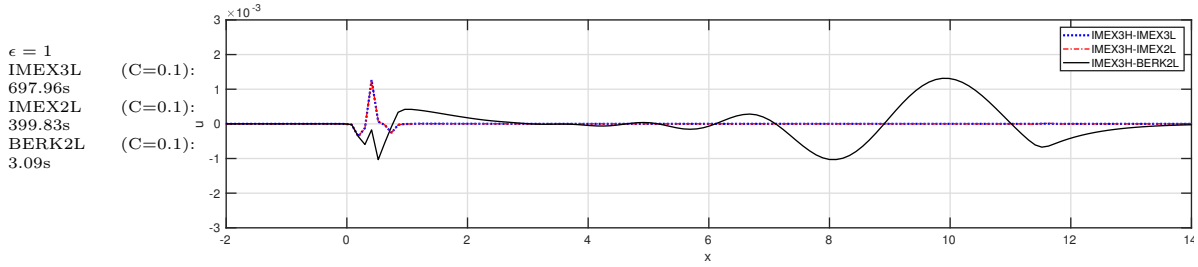


(d) ρ

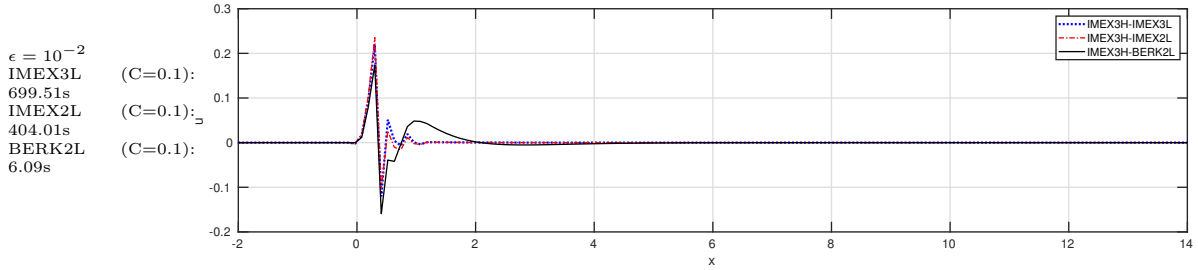
Figure 4: Comparison of numerical solutions for the gas injection problem at t=0.1 with various ϵ. Row 1: IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for ρ, we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2L. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. (Parameters for each method - IMEX3H: $N_x = 800$, $N_v = 1100$, $C = 0.1$; IMEX3L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; IMEX2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; BERK2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$).



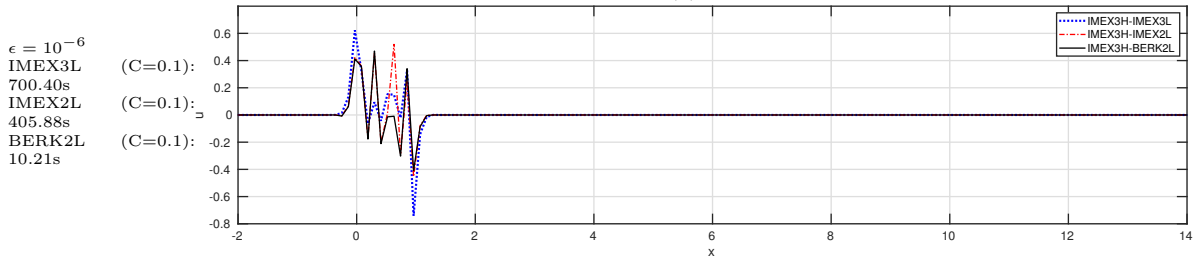
(a) u



(b) u



(c) u



(d) u

Figure 5: Comparison of numerical solutions for the gas injection problem at t=0.1 with various ϵ. Row 1: IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for u, we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2L. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. (Parameters for each method - IMEX3H: $N_x = 800$, $N_v = 1100$, $C = 0.1$; IMEX3L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; IMEX2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; BERK2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$).

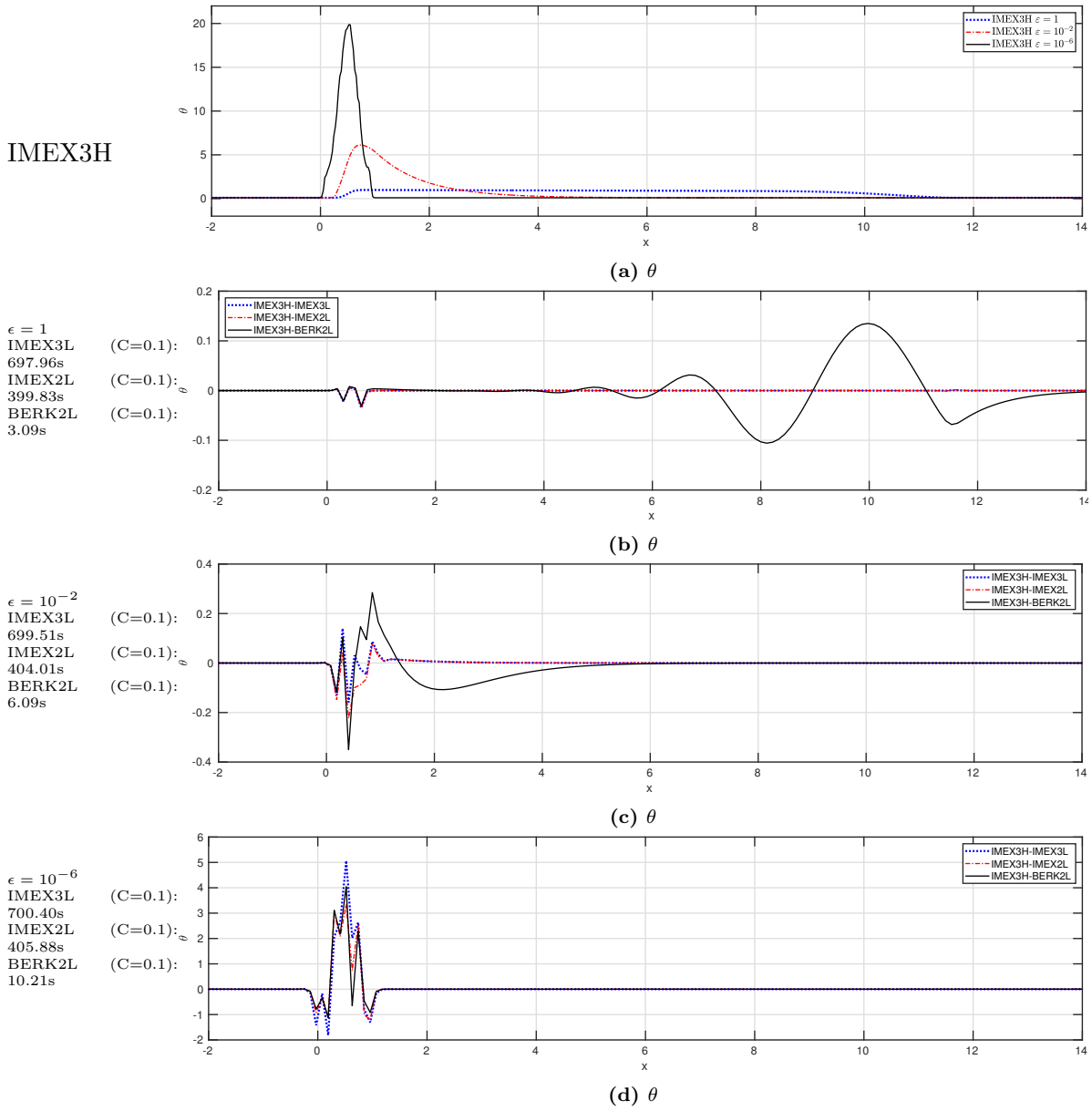


Figure 6: Comparison of numerical solutions for the gas injection problem at t=0.1 with various ϵ. Row 1: IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for θ, we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2L. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. (Parameters for each method - IMEX3H: $N_x = 800$, $N_v = 1100$, $C = 0.1$; IMEX3L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; IMEX2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; BERK2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$).

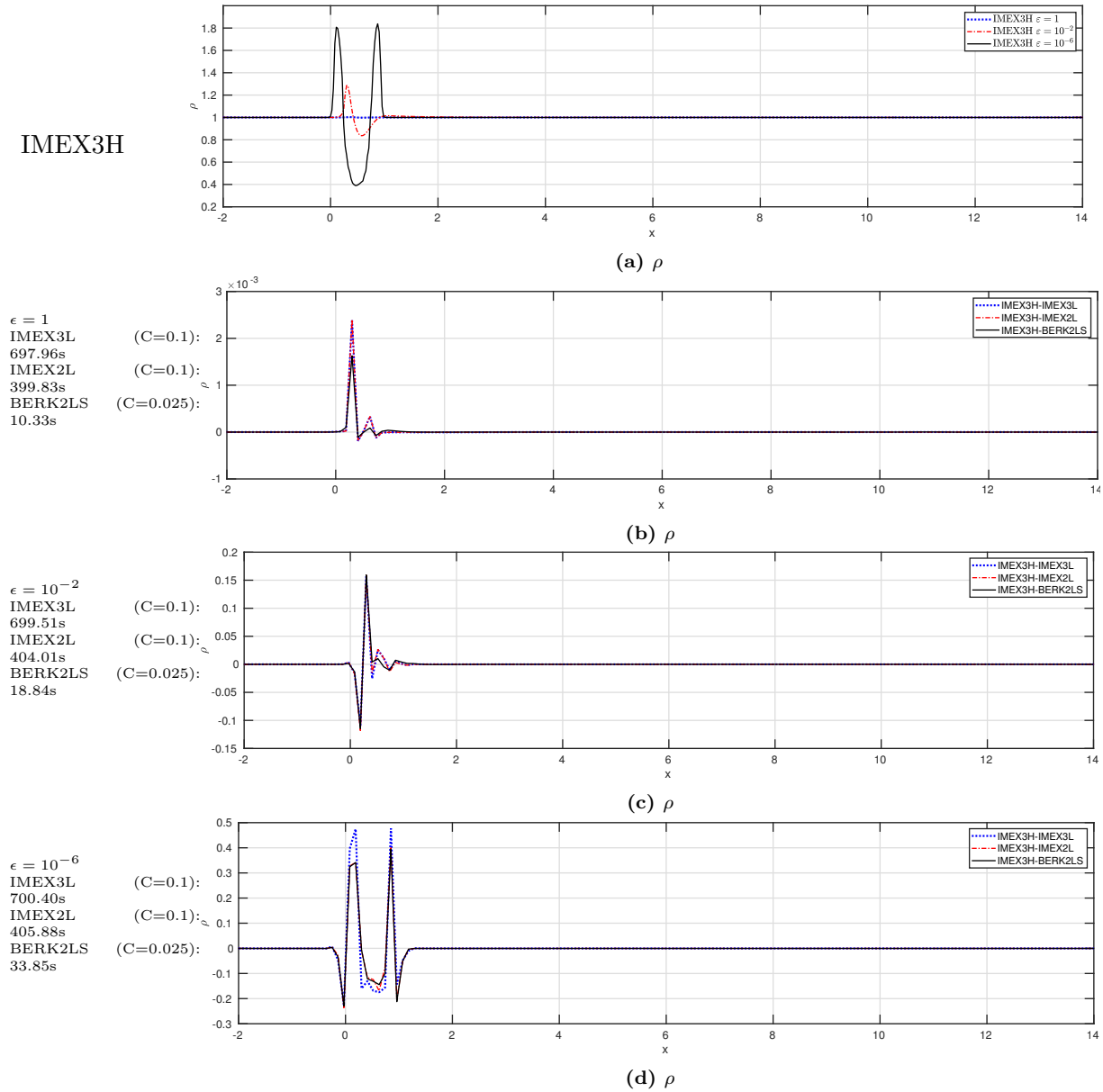


Figure 7: Comparison of numerical solutions for the gas injection problem at t=0.1 with various ϵ. Row 1: IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for ρ, we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2LS. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. (Parameters for each method - IMEX3H: $N_x = 800$, $N_v = 1100$, $C = 0.1$; IMEX3L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; IMEX2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; BERK2LS: $N_x = 200$, $N_v = 1000$, $C = 0.025$).

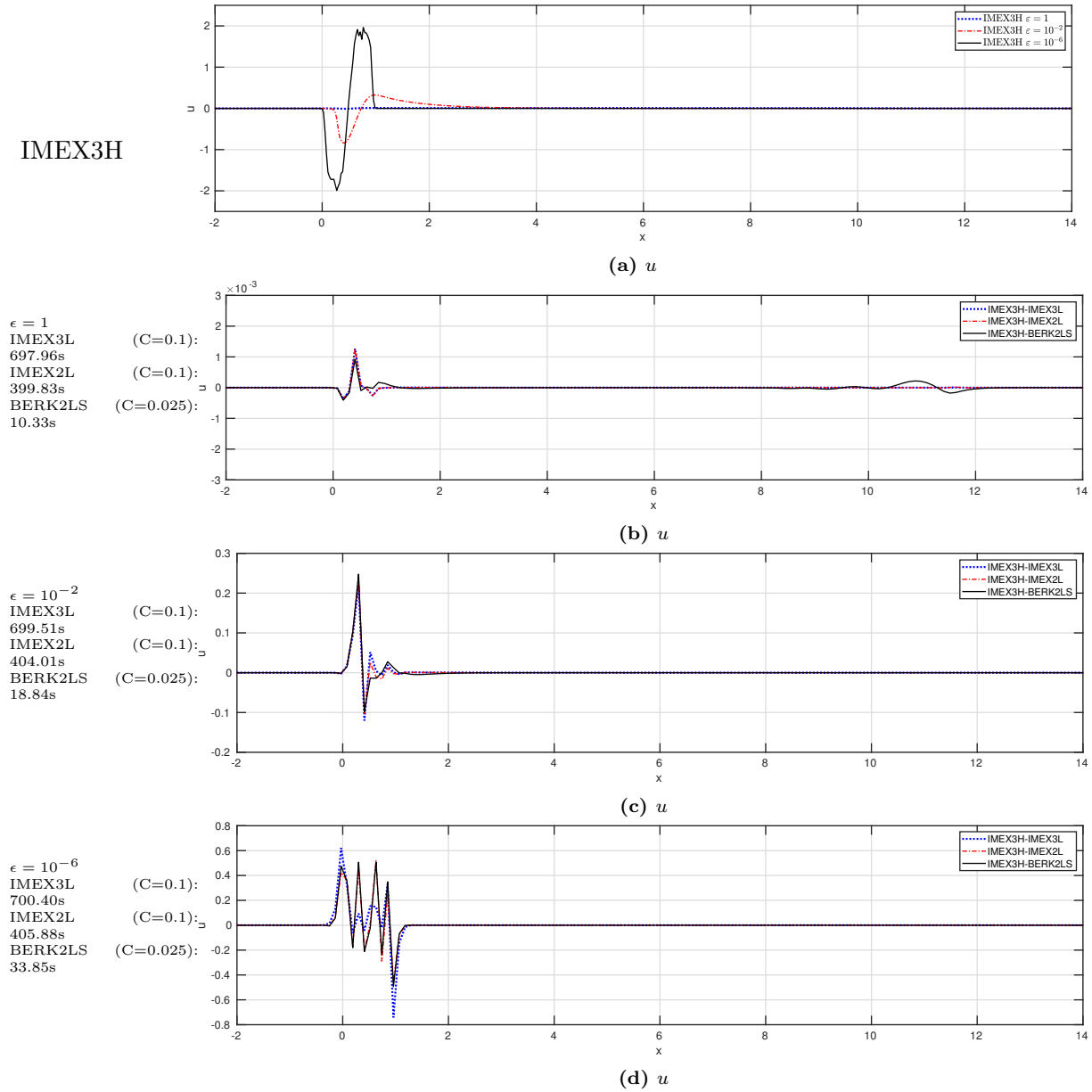


Figure 8: Comparison of numerical solutions for the gas injection problem at t=0.1 with various ϵ. Row 1: IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for u, we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2LS. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. (Parameters for each method - IMEX3H: $N_x = 800$, $N_v = 1100$, $C = 0.1$; IMEX3L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; IMEX2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; BERK2LS: $N_x = 200$, $N_v = 1000$, $C = 0.025$).

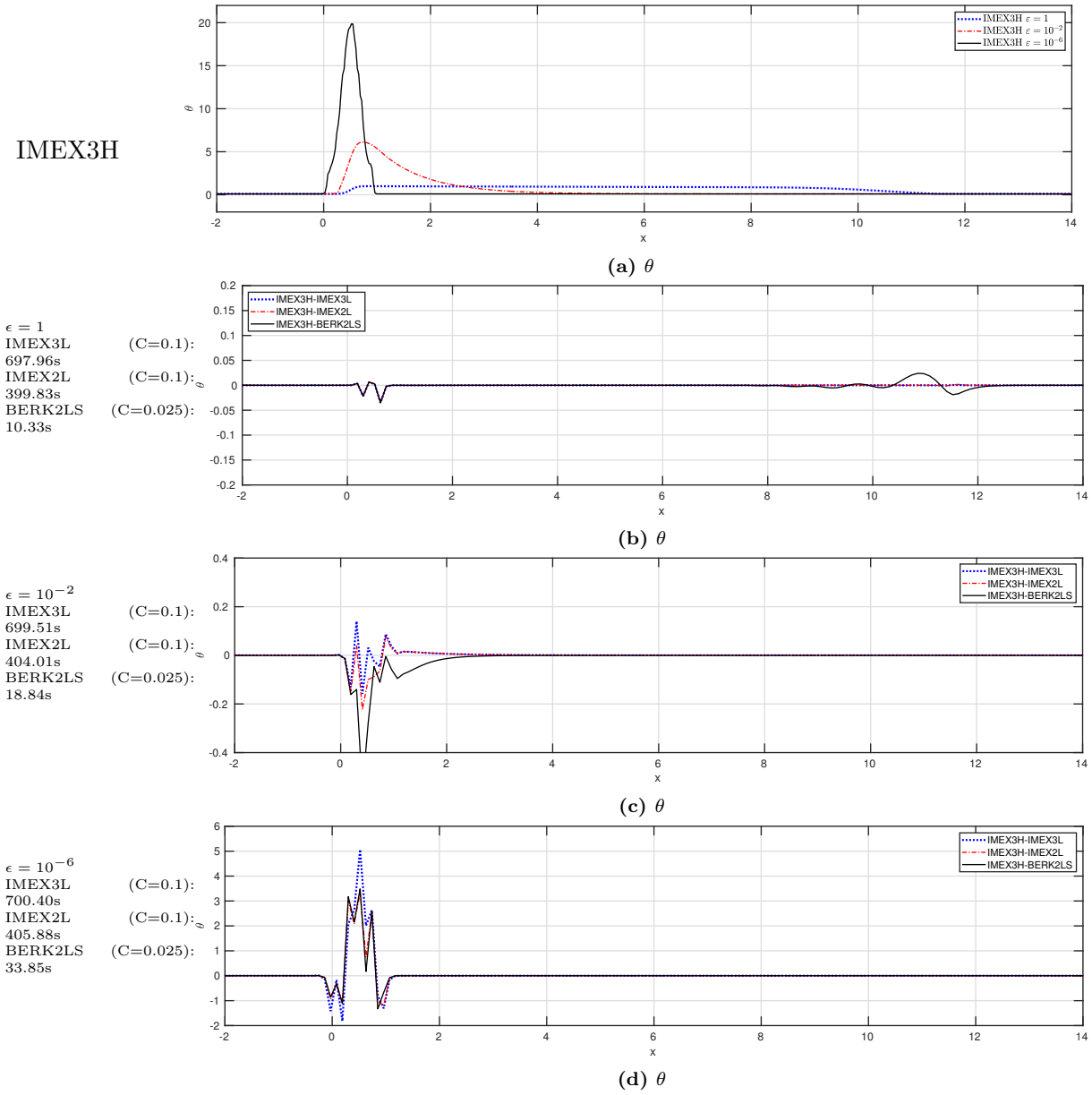


Figure 9: Comparison of numerical solutions for the gas injection problem at t=0.1 with various ϵ. Row 1: IMEX3H at different values of ϵ. Row 2: For $\epsilon = 1$, and for θ, we plot the differences; IMEX3H-IMEX3L, IMEX3H–IMEX2L and IMEX3H–BERK2LS. Row 3: Same as Row 1, but for $\epsilon = 10^{-2}$, Row 4: Same as Row 1, but for $\epsilon = 10^{-6}$. The computation time for each method is shown in the first column. (Parameters for each method - IMEX3H: $N_x = 800$, $N_v = 1100$, $C = 0.1$; IMEX3L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; IMEX2L: $N_x = 200$, $N_v = 1000$, $C = 0.1$; BERK2LS: $N_x = 200$, $N_v = 1000$, $C = 0.025$).

Bibliography

[1] C. D. Hauck, R. G. McClarren, A collision-based hybrid method for time-dependent, linear, kinetic transport equations, Multiscale Model. Simul. 11 (4) (2013) 1197–1227. doi:10.1137/110846610.

[2] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (1954) 511–525. doi:10.1103/PhysRev.94.511.

[3] B. Perthame, Global existence to the BGK model of Boltzmann equation, J. of Differ. Equ. 82 (1) (1989) 191–205. doi:10.1016/0022-0396(89)90173-3.

[4] C. Klingenberg, M. Pirner, G. Puppo, A consistent kinetic model for a two-component mixture with an application to plasma, Kinetic & Related Models 10 (2) (2017) 445. doi:10.3934/krm.2017017.

[5] J. Haack, C. D. Hauck, C. Klingenberg, M. Pirner, S. Warnecke, A consistent BGK model with velocity-dependent collision frequency for gas mixtures, Journal of Statistical Physics 184 (3) (9 2021). doi:10.1007/s10955-021-02821-2.

[6] J. R. Haack, C. D. Hauck, M. S. Murillo, A conservative, entropic multispecies BGK model, Journal of Statistical Physics 168 (4) (2017) 826–856. doi:10.1007/s10955-017-1824-9.

[7] J. Holway, Lowell H., Kinetic theory of shock structure using an ellipsoidal distribution function, in: Rarefied Gas Dynamics, Volume 1, Vol. 1, 1965, p. 193.

[8] P. Andries, P. Le Tallec, J.-P. Perlat, B. Perthame, The Gaussian-BGK model of Boltzmann equation with small Prandtl number, European Journal of Mechanics-B/Fluids 19 (6) (2000) 813–830. doi:10.1016/S0997-7546(00)01103-1.

[9] H. Struchtrup, Macroscopic transport equations for rarefied gas flows, in: Macroscopic transport equations for rarefied gas flows, Springer, 2005, pp. 145–160. doi:10.1007/3-540-32386-4.

[10] Y. T. Lee, R. M. More, An electron conductivity model for dense plasmas, The Physics of Fluids 27 (5) (1984) 1273–1286. doi:10.1063/1.864744.

[11] H. Struchtrup, The BGK-model with velocity-dependent collision frequency, Continuum Mechanics and Thermodynamics 9 (1) (1997) 23–31. doi:10.1007/s001610050053.

[12] J. Haack, C. D. Hauck, C. F. Klingenberg, M. Pirner, S. Warnecke, Numerical schemes for a multi-species BGK model with velocity-dependent collision frequency, Journal of Computational Physics 473 (2023) 111729. doi:https://doi.org/10.1016/j.jcp.2022.111729.

[13] W. Melis, T. Rey, G. Samaey, Projective and telescopic projective integration for the nonlinear BGK and Boltzmann equations, The SMAI journal of computational mathematics 5 (2019) 53–88. doi:10.5802/smai-jcm.43.

[14] F. Coron, B. Perthame, Numerical passage from kinetic to fluid equations, SIAM J. Numer. Anal. 28 (1) (1991) 26–42. doi:10.1137/0728002.

[15] J. Y. Yang, J. C. Huang, Rarefied flow computations using nonlinear model Boltzmann equations, J. Comput. Phys. 120 (1995) 323–339. doi:10.1006/jcph.1995.1168.

[16] E. M. Shakhov, Generalization of the Krook kinetic relaxation equation, Fluid dynamics 3 (5) (1968) 95–96. doi:10.1007/BF01029546.

[17] M. Bennoune, M. Lemou, L. Mieussens, Uniformly stable numerical schemes for the Boltzmann equation preserving the compressible Navier–Stokes asymptotics, J. Comput. Phys. 227 (2008) 3781–3803. doi:10.1016/j.jcp.2007.11.032.

[18] L. Mieussens, Discrete-velocity models and numerical schemes for the Boltzmann-BGK equation in plane and axisymmetric geometries, J. Comput. Phys. 162 (2) (2000) 429–466. doi:10.1006/jcph.2000.6548.

[19] P. Andries, J. F. Bourgat, P. Le-Tallec, B. Perthame, Numerical comparison between the Boltzmann and ES-BGK models for rarefied gases, Comput. Method Appl. M. 191 (31) (2002) 33–69. doi:10.1016/S0045-7825(02)00253-0.

[20] J. Hu, X. Zhang, On a class of implicit–explicit Runge–Kutta schemes for stiff kinetic equations preserving the navier–stokes limit, J. Sci. Comput. 73 (2017) 797—-818. doi:10.1007/s10915-017-0499-3.

[21] S. Pieraccini, G. Puppo, Implicit–explicit schemes for BGK kinetic equations, J. Sci. Comput. 32 (2007) 1–28. doi:10.1007/s10915-006-9116-6.

[22] S. Pieraccini, G. Puppo, Microscopically implicit–macroscopically explicit schemes for the BGK equation, J. Comput. Phys. 231 (2012) 299–327. doi:10.1016/j.jcp.2011.08.027.

[23] F. Filbet, S. Jin, An asymptotic preserving scheme for the ES–BGK model of the Boltzmann equation, J. Sci. Comput. 46 (2011) 204–224. doi:10.1007/s10915-010-9394-x.

[24] K. Xu, A gas-kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method, J. Comput. Phys. 171 (2001) 289–335. doi:10.1006/jcph.2001.6790.

[25] C. Bardos, F. Golse, D. Levermore, Fluid dynamic limits of kinetic equations. I. Formal derivations, Journal of statistical physics 63 (1) (1991) 323–344. doi:10.1007/BF01026608.

[26] E. W. Larsen, J. E. Morel, Advances in discrete-ordinates methodology, Nuclear Computational Science (2010) 1–84doi:10.1007/978-90-481-3411-3_1.

[27] V. P. DeCaria, C. D. Hauck, M. T. P. Laiu, Analysis of a new implicit solver for a semiconductor model, SIAM Journal on Scientific Computing 43 (3) (2021) B733–B758. doi:10.1137/20M1365922.

[28] M. P. Laiu, Z. Chen, C. D. Hauck, A fast implicit solver for semiconductor models in one space dimension, Journal of Computational Physics 417 (2020) 109567. doi:10.1016/j.jcp.2020.109567.

[29] W. T. Taitano, D. A. Knoll, L. Chacón, J. M. Reisner, A. K. Prinja, Moment-based acceleration for neutral gas kinetics with BGK collision operator, Journal of Computational and Theoretical Transport 43 (1-7) (2014) 83–108. doi:10.1080/00411450.2014.910228.

[30] A. V. Bobylev, A. Palczewski, J. Schneider, On approximation of the Boltzmann equation by discrete velocity models, C. R. Acad. Sci. Paris Ser. I Math. 320 (1995) 639–644. doi:10.1007/BF02180142.

[31] L. Mieussens, Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics, Mathematical Models and Methods in Applied Sciences 10 (08) (2000) 1121–1149. doi:10.1142/S0218202500000562.

[32] C. K. Garrett, C. D. Hauck, A fast solver for implicit integration of the Vlasov-Poisson system in the eulerian framework, SIAM Journal on Scientific Computing 40 (2) (2018) B483–B506. doi:10.1137/17M1134184.

[33] M. Ding, J.-M. Qiu, R. Shu, Semi-Lagrangian nodal discontinuous Galerkin method for the BGK model, arXiv preprint arXiv:2105.02421 (2021).

[34] S. Cho, S. Boscarino, G. Russo, S. Yun, Conservative semi-Lagrangian schemes for kinetic equations part II: Applications, Journal of Computational Physics 436 (2021) 110281. doi:10.1016/j.jcp.2021.110281.

[35] M. Groppi, G. Russo, G. Stracquadanio, High order semi-lagrangian methods for the BGK equation, arXiv preprint arXiv:1411.7929 (2014). doi:10.48550/arXiv.1411.7929.

[36] M. M. Crockatt, A. J. Christlieb, C. K. Garrett, C. D. Hauck, An arbitrary-order, fully implicit, hybrid kinetic solver for linear radiative transport using integral deferred correction, J. Comput. Phys. 346 (2017) 212–241. doi:10.1016/j.jcp.2017.06.017.

[37] M. M. Crockatt, A. J. Christlieb, C. K. Garrett, C. D. Hauck, Hybrid methods for radiation transport using diagonally implicit Runge–Kutta and space–time discontinuous Galerkintime integration, J. Comput. Phys. 376 (2019) 455–477. doi:10.1016/j.jcp.2018.09.041.

[38] R. G. McClarren, J. A. Rossmanith, M. Shin, Semi-implicit hybrid discrete (H_N^T) approximation of thermal radiative transfer, Journal of Scientific Computing 90 (1) (2021) 2. doi:10.1007/s10915-021-01686-7.

[39] R. G. McClarren, T. M. Evans, R. B. Lowrie, J. D. Densmore, Semi-implicit time integration for PN thermal radiative transfer, Journal of Computational Physics 227 (16) (2008) 7561–7586. doi:10.1016/j.jcp.2008.04.029.

[40] R. E. Alcouffe, A first collision source method for coupling Monte Carlo and discrete ordinates for localized source problems, in: R. Alcouffe, R. Dautray, A. Forster, G. Ledanois, B. Mercier (Eds.), Monte-Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1985, pp. 352–366. doi:10.1007/BFb0049062.

[41] R. E. Alcouffe, R. D. O'Dell, F. W. Brinkley, A first-collision source method that satisfies discrete Sn transport balance, Nuclear Science and Engineering 105 (2) (1990) 198–203. doi:10.13182/NSE90-A23749.

[42] T. Xiong, J. Jang, F. Li, J.-M. Qiu, High order asymptotic preserving nodal discontinuous Galerkin IMEX schemes for the BGK equation, J. Comput. Phys. 284 (2015) 70–94. doi:doi.org/10.1016/j.jcp.2014.12.021.

[43] C. Cercignani, R. Illner, M. Pulvirenti, The mathematical theory of dilute gases, Vol. 106, Springer Science & Business Media, 2013.

[44] C. Cercignani, The Boltzmann equation and its applications, Springer, 1988.

[45] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Science & Business Media, 2013.

[46] L. Mieussens, Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamic, Math Models Methods Appl Sci. 10 (8) (2000) 1121–1149. doi:10.1142/S0218202500000562.

[47] A. Palczewski, J. Schneider, A. V. Bobylev, A consistency result for a discrete-velocity model of the Boltzmann equation, SIAM J. Numer. Anal. 34 (1997) 1865–1883. doi:10.1137/S0036142995289007.

[48] Y. Cheng, A. J. Christlieb, X. Zhong, Energy-conserving discontinuous Galerkin methods for the Vlasov–Ampère system, Journal of Computational Physics 256 (2014) 630–655. doi:10.1016/j.jcp.2013.09.013.

[49] F. Filbet, E. Sonnendrücker, P. Bertrand, Conservative numerical schemes for the Vlasov equation, Journal of Computational Physics 172 (1) (2001) 166–187. doi:10.1006/jcph.2001.6818.

[50] C. Bardos, F. Golse, Y. Sone, Half-space problems for the Boltzmann equation: A survey, Journal of statistical physics 124 (2006). doi:10.1007/s10955-006-9077-z.

[51] A. Bensoussan, P.-L. Lions, G. C. Papanicolaou, Boundary layers and homogenizatlon of transport processes, Publications of the Research Institute for Mathematical Sciences 15 (1) (1979) 53–157. doi:10.2977/prims/1195188427.

[52] G. I. Bell, S. Glasstone, Nuclear reactor theory, Tech. rep., US Atomic Energy Commission, Washington, DC (United States) (1970).

[53] E. E. Lewis, W. F. Miller, Computational Methods of Neutron Transport, American Nuclear Society Scientific Publications, 1993.

[54] C. D. H. Ben Whewell, Ryan G. McClarren, M. Shin, Multigroup neutron transport using a collision-based hybrid method, Nuclear Science and Engineering 197 (7) (2023) 1386–1405. doi:10.1080/00295639.2022.2154119.

[55] I. M. Gamba, S. H. Tharkabhushanam, Spectral-lagrangian methods for collisional models of non-equilibrium statistical states, Journal of Computational Physics 228 (6) (2009) 2012–2036. doi:10.1016/j.jcp.2008.09.033.

[56] M. M. Crockatt, A. J. Christlieb, C. D. Hauck, Improvements to a class of hybrid methods for radiation transport: Nyström reconstruction and defect correction methods, Journal of Computational Physics 422 (2020) 109765. doi:10.1016/j.jcp.2020.109765.

[57] W. H. Reed, T. R. Hill, Triangular mesh methods for the neutron transport equation, Tech. rep., Los Alamos Scientific Lab., N. Mex.(USA) (1973).

[58] C.-W. Shu, Discontinuous Galerkin methods: general approach and stability, Numerical solutions of partial differential equations 201 (2009).

[59] B. Cockburn, S.-Y. Lin, Tvb Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems, J. Comput. Phys. 84 (1989) 90–113. doi:10.1016/0021-9991(89)90183-6.

[60] L. Pareschi, G. Russo, Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation, J. Sci. Comput. 25 (2005) 129–155. doi:10.1007/s10915-004-4636-4.

[61] U. M. Ascher, S. J. Ruuth, R. J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Applied Numerical Mathematics 25 (2) (1997) 151–167, special Issue on Time Integration. doi:10.1016/S0168-9274(97)00056-1.

[62] J. A. Rossmanith, DoGPack: Discontinuous Galerkin package, http://dogpack-code.org/.

[63] D. I. Ketcheson, Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations, SIAM J. on Scientific Computing 30 (4) (2008) 2113–2136. doi:10.1137/07070485X.

[64] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comp. Phys. 27 (1978) 1–31. doi:10.1016/0021-9991(78)90023-2.

[65] P. D. Lax, Weak solution of non-linear hyperbolic equations and their numerical computations, Commun. Pure Appl. Math. 7 (1954) 159–193. doi:10.1002/cpa.3160070112.

[66] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comp. Phys. 83 (1989) 32–78. doi:10.1016/0021-9991(88)90177-5.

[67] R. Bürger, S. K. Kenettinkara, D. Zorío, Approximate Lax–Wendroff discontinuous Galerkin methods for hyperbolic conservation laws, Comput. Math. with Appl. 74 (6) (2017) 1288–1310. doi:10.1016/j.camwa.2017.06.019.

Declaration of interests

☒ 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.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: