

HIGH ORDER SEMI-LAGRANGIAN METHODS FOR THE BGK EQUATION*

MARIA GROPPÌ†, GIOVANNI RUSSO‡, AND GIUSEPPE STRACQUADANIO§

Abstract. A new class of high-order accuracy numerical methods for the BGK model of the Boltzmann equation is presented. The schemes are based on a semi-Lagrangian formulation of the BGK equation; time integration is dealt with DIRK (Diagonally Implicit Runge–Kutta) and BDF methods; the latter turn out to be accurate and computationally less expensive than the former. Numerical results and examples show that the schemes are reliable and efficient for the investigation of both rarefied and fluid regimes in gas dynamics.

Key words. Boltzmann equation, BGK equation, semi-Lagrangian methods, high-order schemes.

AMS subject classifications. 76P05, 65M25, 65L06.

1. Introduction

In the kinetic theory of gases, the dynamics of a monoatomic rarefied gas system is described by the Boltzmann equation [7]. The numerical approximation of this equation is not trivial due to the complex structure of the collision operator. The BGK equation, introduced by Bhatnagar, Gross, and Krook [4] and independently by Welander [29] is a simplified model of the Boltzmann equation. In the BGK model the collision operator is substituted by a relaxation operator; the initial value problem reads as

$$\begin{aligned} \frac{\partial f}{\partial t} + v \cdot \nabla_x f &= Q_{BGK}[f] \equiv \frac{1}{\varepsilon} (M[f] - f), \quad (x, v, t) \in \mathbb{R}^d \times \mathbb{R}^N \times \mathbb{R}^+ \\ f(x, v, 0) &= f_0(x, v), \end{aligned} \tag{1.1}$$

where d and N denote the dimension of the physical and velocity spaces, respectively, and ε^{-1} is the collision frequency, which throughout this paper is assumed to be a fixed constant for simplicity. $M[f]$ denotes the local Maxwellian with the same macroscopic moments of the distribution function $f(x, v, t)$, and is given by

$$M[f](x, v, t) = \frac{\rho(x, t)}{[2\pi RT(x, t)]^{N/2}} \exp\left(-\frac{(v - u(x, t))^2}{2RT(x, t)}\right), \tag{1.2}$$

where R is the ideal gas constant and $\rho(x, t) \in \mathbb{R}^+$, $u(x, t) \in \mathbb{R}^N$, and $T(x, t) \in \mathbb{R}^+$ denote the macroscopic moments of the distribution function f , that is: density, mean velocity, and temperature, respectively. They are obtained in the following way:

$$(\rho, \rho u, E)^T = \langle f \phi(v) \rangle, \quad \text{where} \quad \phi(v) = \left(1, v, \frac{1}{2}|v|^2\right)^T \quad \text{and} \quad \langle g \rangle = \int_{\mathbb{R}^N} g(v) d_N v. \tag{1.3}$$

The physical quantity $E(x, t)$ is the total energy, that is related to the temperature $T(x, t)$ by the underlying relation:

$$E(x, t) = \frac{1}{2} \rho(x, t) u(x, t)^2 + \frac{N}{2} \rho(x, t) RT(x, t).$$

*Received: July 22, 2014; accepted (in revised form): February 18, 2015. Communicated by Lorenzo Pareschi.

†Dept. of Mathematics and Computer Science, Univ. of Parma, Italy (maria.groppi@unipr.it).

‡Dept. of Mathematics and Computer Science, Univ. of Catania, Italy (russo@dmi.unict.it).

§Dept. of Mathematics and Computer Science, Univ. of Parma, Italy (giuseppe.stracquadiano@mo.unipr.it).

The BGK model (1.1) satisfies the main properties of the Boltzmann equation [4, 29], such as conservation of mass, momentum, and energy, as well as entropy dissipation. In details, this is

$$\langle M[f]\phi(v) \rangle = \langle f\phi(v) \rangle, \quad \int_{\mathbb{R}^N} Q_{BGK}[f] \log f dNv \leq 0. \quad (1.4)$$

The equilibrium solutions are clearly Maxwellians, indeed the collision operator vanishes for $f = M[f]$. The BGK model is computationally less expensive than the Boltzmann equation, due mainly to the simpler form of the collision operator, but it still provides qualitatively correct solutions for the macroscopic moments near the fluid regime¹. These two aspects, the lower computational complexity and the correct description of the hydrodynamic limit, explain the interest in the BGK model over the last several years. Without expecting to be exhaustive, we refer to, for instance, [2, 18, 20, 21, 22, 26, 30] and the references therein for a more in-depth analysis of both theoretical and numerical aspects of BGK models. In particular, in the last few years many numerical schemes have been proposed to solve the BGK equation in an efficient way, by adopting schemes that are able to capture the limiting behavior of the solution as $\varepsilon \rightarrow 0$ (asymptotic preserving schemes, AP). For example, in [22] the authors use IMEX schemes from [19], in which the implicit part is L -stable, thus guaranteeing that the schemes project the numerical solution onto the discrete Maxwellian as $\varepsilon \rightarrow 0$. Such schemes, originating from the pioneering work by Jin [14], have gained considerable interest since they also allow the reproduction of the Euler fluid limit at a discrete level. As additional examples, the very recent paper [9] is concerned with a fast method with AP property based on splitting techniques, while the AP scheme proposed in [23] takes advantage of the explicit advancing in time of the macroscopic fields involved in the BGK operator.

The aim of this paper is to develop high order semi-Lagrangian numerical schemes for the BGK equation. Semi-Lagrangian methods for BGK models have recently received increasing interest [11, 24], since they well describe either a rarefied or a fluid regime. The relaxation operator is treated implicitly, and the semi-Lagrangian treatment of the convective part avoids the classical CFL stability restriction. Moreover, in this work time integration is dealt with BDF methods along characteristics, which turn out to be accurate but computationally less expensive than Diagonally Implicit Runge–Kutta (DIRK) methods introduced in [25] and analyzed in [24].

The paper is organized as follows. In Section 2 the semi-Lagrangian method is introduced and the first order method is described; in Section 3 higher order methods are presented, based on BDF or DIRK schemes for time integrations; the possibility to avoid interpolation is investigated in Section 4. For simplicity, all schemes are described for the 1+1D BGK model. In Section 5 we describe how to extend the methods to 1D in space and 3D in velocity in slab geometry (Chu reduction [8]). Numerical results are shown in Section 6, with the aim of showing the performance and the accuracy of the proposed methods in various examples. Results and future developments are briefly commented on in Section 7.

¹More precisely, from the BGK model, to zero-th order in ε , one obtains the compressible Euler equations in the fluid-dynamic limit, while to first order in ε , the moments satisfy equations of compressible Navier–Stokes type, but with the wrong value for the Prandtl number. This problem can be fixed by resorting to the so-called ES-BGK model [16], but in the present paper we shall restrict to the classical BGK model.

2. Lagrangian formulation and first order scheme

We shall restrict to the BGK equation in one space and velocity dimension (namely $d = N = 1$ in (1.1), (1.2)). In the Lagrangian formulation, the time evolution of $f(x, v, t)$ along the characteristic lines is given by the following system:

$$\begin{aligned} \frac{df}{dt} &= \frac{1}{\varepsilon} (M[f] - f), \\ \frac{dx}{dt} &= v, \\ x(0) &= \tilde{x}, \quad f(x, v, 0) = f_0(x, v), \quad t \geq 0, \quad x, v \in \mathbb{R}. \end{aligned} \tag{2.1}$$

For simplicity, we assume constant time step Δt and uniform grid in physical and velocity space, with mesh spacing Δx and Δv , respectively, and denote the grid points by $t^n = n\Delta t$, $x_i = x_0 + i\Delta x, i = 0, \dots, N_x$, $v_j = j\Delta v, j = -N_v, \dots, N_v$, where $N_x + 1$ and $2N_v + 1$ are the number of grid nodes in space and velocity, respectively, so that $[x_0, x_{N_x}]$ is the space domain. We also denote the approximate solution $f(x_i, v_j, t^n)$ by f_{ij}^n . Relaxation time ε is typically of the order of the Knudsen number, defined as the ratio between the molecular mean free path length and a representative macroscopic length; thus, the Knudsen number can vary in a wide range, from order greater than one (in rarefied regimes) to very small values (in fluid dynamic regimes).

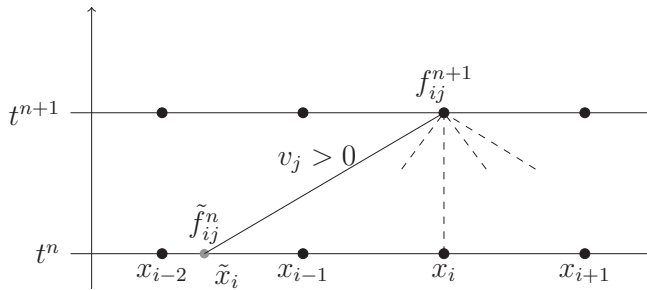


FIG. 2.1. Representation of the implicit first order scheme. The foot of the characteristic does not lie on the grid, and some interpolation is needed to compute \tilde{f}_{ij}^n .

For this reason, if we want to capture the fluid-dynamic limit, we have to use an L-stable scheme in time. An implicit first order L-stable semi-Lagrangian scheme (Figure 2.1) can be achieved in this simple way

$$f_{ij}^{n+1} = \tilde{f}_{ij}^n + \frac{\Delta t}{\varepsilon} (M[f]_{ij}^{n+1} - f_{ij}^{n+1}). \tag{2.2}$$

The quantity $\tilde{f}_{ij}^n \simeq f(x_i - v_j\Delta t, v_j, t^n)$ can be computed by suitable reconstruction from $\{f_{\cdot j}^n\}$; linear reconstruction will be sufficient for first order scheme, while higher order reconstructions, such as ENO or WENO [6], may be used to achieve high order avoiding oscillations. The convergence of this first order scheme has been studied in [24].

$M[f]_{ij}^{n+1}$ is the discrete Maxwellian constructed with the macroscopic moments of f^{n+1} :

$$M[f]_{ij}^{n+1} = M[f](x_i, v_j, t^{n+1}) = \frac{\rho_i^{n+1}}{\sqrt{2\pi RT_i^{n+1}}} \exp\left(-\frac{(v_j - u_i^{n+1})^2}{2RT_i^{n+1}}\right).$$

This formula requires the computation of the discrete moments of f^{n+1} , through a numerical approximation of the integrals in (1.3). This is obtained in the following standard way²:

$$\begin{aligned} \rho_i^{n+1} &= \sum_{j=-N_v}^{N_v} f_{ij}^{n+1} \Delta v, \\ u_i^{n+1} &= \frac{1}{\rho_i^{n+1}} \sum_{j=-N_v}^{N_v} v_j f_{ij}^{n+1} \Delta v, \\ E_i^{n+1} &= \sum_{j=-N_v}^{N_v} \frac{1}{2} v_j^2 f_{ij}^{n+1} \Delta v. \end{aligned} \tag{2.3}$$

From now on, we will denote formulas in (2.3) with the more compact notation: $(\rho_i^{n+1}, (\rho u)_i^{n+1}, E_i^{n+1}) = m[f_i^{n+1}]$, where, in general, $m[f]$ will indicate the approximated macroscopic moments related to the distribution function f .

Now it is evident that equation (2.2) cannot be immediately solved for f_{ij}^{n+1} . It is a non-linear implicit equation because the Maxwellian depends on f^{n+1} itself through its moments. To solve this implicit step one can act as follows. Let us take the moments of equation (2.2); this is obtained at the discrete level multiplying both sides by $\phi_j \Delta v$, where $\phi_j = \{1, v_j, v_j^2\}$ and summing over j as in (2.3). Then we have

$$\Delta v \sum_j (f_{ij}^{n+1} - \tilde{f}_{ij}^n) \phi_j = \Delta v \frac{\Delta t}{\varepsilon} \sum_j (M[f_{ij}^{n+1}] - f_{ij}^{n+1}) \phi_j,$$

which implies that

$$\sum_j f_{ij}^{n+1} \phi_j = \sum_j \tilde{f}_{ij}^n \phi_j,$$

because, by definition, the Maxwellian at time t^{n+1} has the same moments as f^{n+1} and we assume that equation (2.3) is accurate enough. This in turn gives

$$m[f_i^{n+1}] \simeq m[\tilde{f}_i^n]. \tag{2.4}$$

Once the Maxwellian at time t^{n+1} is known using the approximated macroscopic moments $m[\tilde{f}_i^n]$, the distribution function f_{ij}^{n+1} can be explicitly computed as

$$f_{ij}^{n+1} = \frac{\varepsilon \tilde{f}_{ij}^n + \Delta t M_{ij}^{n+1}}{\varepsilon + \Delta t}. \tag{2.5}$$

This approach has already been used in [24, 25, 27], and in [22] in the context of Eulerian schemes.

²The left-hand sides of (2.3) correspond to the moments of the continuous Maxwellian $M_i^{n+1}(v) = \frac{\rho_i^{n+1}}{\sqrt{2\pi RT_i^{n+1}}} \exp(-\frac{(v-u_i^{n+1})^2}{2RT_i^{n+1}})$; the right-hand sides are obtained by taking the discrete moments of f^{n+1} , and therefore they are not exactly equal to the moments of the discrete Maxwellian $M_{ij}^{n+1} = \frac{\rho_i^{n+1}}{\sqrt{2\pi RT_i^{n+1}}} \exp(-\frac{(v_j-u_i^{n+1})^2}{2RT_i^{n+1}})$. The discrepancy is very small if the distribution function is smooth and the number of points in velocity space is large enough, because midpoint rule is spectrally accurate for smooth functions having (numerically) compact support. However, for small values of N_v , such discrepancy can be noticeable. To avoid this drawback, Mieussens introduced a proper discrete Maxwellian [18, 17]. The computation of the parameters of such Maxwellian requires the solution of a non-linear system. A comparison between the continuous and discrete Maxwellian can be found, for example, in [1]. Here we shall neglect this effect, and assume that, using equation (2.3), M_{ij}^{n+1} , and $M_i^{n+1}(v)$ have the same moments with sufficient approximation.

3. High order methods

3.1. Runge–Kutta methods. The scheme of the previous section corresponds to the implicit Euler method applied to the BGK model in characteristic form. High order discretization in time can be obtained by Runge–Kutta or BDF methods.

In [25, 27], the relaxation operator has been dealt with an L-stable diagonally implicit Runge–Kutta (DIRK) scheme [15]. DIRK schemes are completely characterized by the triangular $\nu \times \nu$ matrix $A = (a_{lk})$, and by the coefficients vectors $\underline{c} = (1, \dots, c_\nu)^T$ and $\underline{b} = (b_1, \dots, b_\nu)^T$, which are derived by imposing accuracy and stability constraints [15].

DIRK schemes can be represented through the Butcher’s table

$$\frac{\underline{c}}{\quad} \left| \begin{array}{c} A \\ \underline{b}^T \end{array} \right.$$

Here we consider the following DIRK schemes

$$\text{RK2} = \frac{\alpha}{1} \left| \begin{array}{cc} \alpha & 0 \\ 1-\alpha & \alpha \\ 1-\alpha & \alpha \end{array} \right., \quad \text{RK3} = \frac{\frac{1}{2}}{(1+\gamma)/2} \left| \begin{array}{cc} \gamma & 0 \\ (1-\gamma)/2 & \gamma \\ 1-\delta-\gamma & \delta \\ 1-\delta-\gamma & \delta \end{array} \right.$$

which are a second and third order L-implicit schemes, respectively [3]. The coefficient α is

$$\alpha = 1 - \frac{\sqrt{2}}{2},$$

while γ is the middle root of $6x^3 - 18x^2 + 9x - 1$, $\gamma \simeq 0.4358665215$, and $\delta = 3/2\gamma^2 - 5\gamma + 5/4 \simeq -0.644363171$. Both RK schemes have the property that the last row of the matrix A equals \underline{b}^T , therefore the numerical solution is equal to the last stage value. Such schemes are called “stiffly accurate”. An A-stable scheme which is stiffly accurate is also L-stable [15].

Here we illustrate the application of the DIRK schemes RK2 and RK3 to the characteristic formulation of the BGK equation (first equation in (2.1)).

We have to point out that, in a semi-Lagrangian DIRK scheme with ν stages, the intermediate stage values, that will be denoted by a capital F , are computed along the ν -characteristics corresponding to each x_i and v_j (see Figures 3.1 and 3.2). Then we need two indices, one for the characteristic and one for the stage value. In detail, $F_{ij}^{(s,\ell)}$ will be the ℓ -th stage value along the s -th characteristic. When $\ell = 0$, $F_{ij}^{(s,0)} = \tilde{f}(t^n, x_i - c_s \Delta t v_j, v_j)$, where \tilde{f} is a suitable reconstruction of f obtained from $\{f_j^n\}$, that is, $F_{ij}^{(s,0)}$ are the values of the distribution function at the feet of the characteristics at time t^n . High order methods in time are coupled with high order interpolation techniques in space; in Appendix A. we show the WENO [6] reconstruction adopted in this paper.

3.1.1. RK2. The expression of RK2 (see Figure 3.1) applied to the BGK equation (2.1) is:

$$f_{ij}^{n+1} = F_{ij}^{(2,0)} + \Delta t (b_1 K_{ij}^{(2,1)} + b_2 K_{ij}^{(2,2)}), \tag{3.1}$$

where

$$K_{ij}^{(2,\ell)} = \frac{1}{\varepsilon} (M[F_{ij}^{(2,\ell)}] - F_{ij}^{(2,\ell)}), \quad \ell = 1, 2,$$

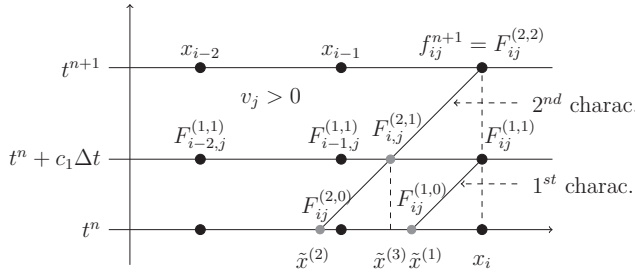


FIG. 3.1. Representation of the RK2 scheme. The black circles denote grid nodes, the gray ones the points where interpolation is needed. Note that since we use a stiffly accurate scheme, the numerical solution f_{ij}^{n+1} is equal to the second stage value $F_{ij}^{(2,2)}$.

denotes the RK fluxes on the second characteristic $x = x_i + v_j(t - t^{n+1})$, and

$$F_{ij}^{(2,\ell)} = F_{ij}^{(2,0)} + \Delta t \sum_{k=1}^{\ell} a_{\ell k} K_{ij}^{(\ell,k)}, \quad \ell = 1, 2,$$

are the stage values along the second characteristic. Moreover $F_{ij}^{(2,0)} = \tilde{f}(t^n, x_i - c_2 \Delta t v_j, v_j)$.

In order to solve the implicit equation involving $F_{ij}^{(2,1)}$ we have first to compute $F_{ij}^{(1,1)}$ on the grid. Indeed, it is worth noticing that the ℓ -th stage value, say $F_{ij}^{(2,\ell)}$, is evaluated by solving an implicit equation involving only $F_{ij}^{(2,\ell)}$, since the previous stage values have already been computed, due to the triangular structure of the matrix A . In our case this is not so easy, in particular for $F_{ij}^{(2,1)}$, because the point $\tilde{x}^{(3)}$ corresponding to first stage along the second characteristics in general is not a grid point, and it is not possible to compute the moments of the Maxwellian at that point in space-time; indeed, after multiplying by ϕ_j and summing over j , the elements of the sum would be computed in variable space points, so we could not take advantage of the useful properties of the collision invariants. For this reason, we need two kinds of stage values: the stage values along the characteristics, $F_{ij}^{(2,1)}$ and $F_{ij}^{(2,2)}$, and the stage values on the grid, $F_{ij}^{(1,1)}$ (see Figure 3.1).

Thus, we compute $F_{ij}^{(1,1)}$ in the grid node using the implicit first order methods with time step $c_1 \Delta t = a_{11} \Delta t$, that is

$$F_{ij}^{(1,1)} = \frac{\varepsilon F_{ij}^{(1,0)} + a_{11} \Delta t M_{ij}^{(1,1)}}{\varepsilon + a_{11} \Delta t}.$$

The Maxwellian $M_{ij}^{(1,1)} = M[F_{ij}^{(1,1)}]$ can be evaluated by means of the macroscopic moments $m[F_{ij}^{(1,0)}]$, using an argument similar to the one adopted in (2.4). $F_{ij}^{(1,0)} = \tilde{f}(t^n, x_i - c_1 \Delta t v_j, v_j)$ can be computed by interpolation.

Once the implicit step is solved, that is, the first stage value $F_{ij}^{(1,1)}$ along the grid is computed, we are able to compute the first stage value $F_{ij}^{(2,1)}$ along the second characteristic, by interpolation from the values of $F_{ij}^{(1,1)}$.

Then the second stage value $F_{ij}^{(2,2)}$ can be computed by

$$F_{ij}^{(2,2)} = F_{ij}^{(2,0)} + \Delta t \left(a_{21} K_{ij}^{(2,1)} + a_{22} \frac{1}{\varepsilon} (M[F_{ij}^{(2,2)}] - F_{ij}^{(2,2)}) \right). \tag{3.2}$$

Equation (3.2) can be solved using the same technique adopted for the implicit first order scheme: taking the moments of both sides of equation (3.2), we can compute the moments of $F_{ij}^{(2,2)}$ since the elements of the sum on j containing the Maxwellian $M[F_{ij}^{(2,2)}]$ are now on fixed space points. Indeed

$$\sum_j \left(F_{ij}^{(2,2)} - F_{ij}^{(2,0)} - \Delta t a_{21} K_{ij}^{(2,1)} \right) \phi_j = a_{22} \frac{\Delta t}{\varepsilon} \sum_j \left(M[F_{ij}^{(2,2)}] - F_{ij}^{(2,2)} \right) \phi_j = 0,$$

thus the moments are given by $m[F_{ij}^{(2,2)}] = m[F_{ij}^{(2,0)} + \Delta t a_{21} K_{ij}^{(2,1)}]$, so we can compute $M[F_{ij}^{(2,2)}]$ and solve the implicit step for $F_{ij}^{(2,2)}$.

Notice that $f_{ij}^{n+1} = F_{ij}^{(2,2)}$, because the scheme is stiffly accurate, i.e., the last row of the matrix A is equal to the vector of weights.

3.1.2. RK3. The RK3 scheme works in a similar way, and Figure 3.2 shows the procedure.

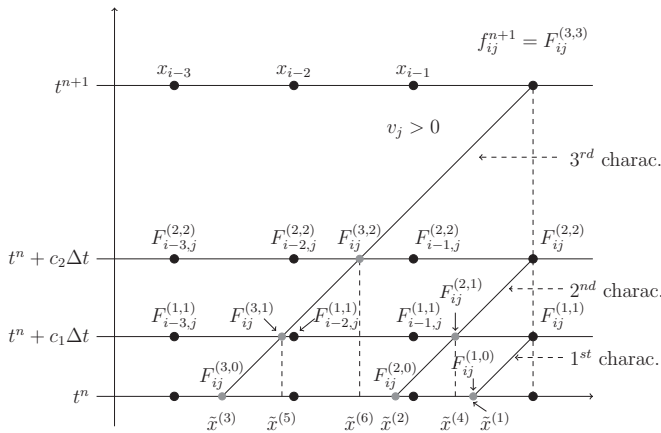


FIG. 3.2. Representation of the RK3 scheme. The black circles denote grid nodes, the gray ones the points where interpolation is needed. Note that since we use a stiffly accurate scheme, the numerical solution f_{ij}^{n+1} is equal to the third stage value $F_{ij}^{(3,3)}$.

Algorithm (RK3)

- Calculate

$$F_{i,j}^{(1,0)} = \tilde{f}(t^n, \tilde{x}^{(1)} = x_i - c_1 v_j \Delta t, v_j),$$

$$F_{i,j}^{(2,0)} = \tilde{f}(t^n, \tilde{x}^{(2)} = x_i - c_2 v_j \Delta t, v_j),$$

$$F_{i,j}^{(3,0)} = \tilde{f}(t^n, \tilde{x}^{(3)} = x_i - v_j \Delta t, v_j)$$

by interpolation from $\{f_{\cdot,j}^n\}$;

- Calculate $F_{ij}^{(1,1)}$ in the grid node using the technique (2.4), (2.5), with Δt replaced by $c_1\Delta t$. Given $F_{ij}^{(1,1)}$, one can compute $F_{ij}^{(2,1)}$ and $F_{ij}^{(3,1)}$, respectively along the second and the third characteristic, by interpolation from $F_j^{(1,1)}$ in $\tilde{x}^{(4)} = x_i - (c_2 - c_1)v_j\Delta t$ and $\tilde{x}^{(5)} = x_i - (1 - c_1)v_j\Delta t$, and then evaluate $K_{ij}^{(2,1)} = \frac{1}{\varepsilon} \left(M[F_{ij}^{(2,1)}] - F_{ij}^{(2,1)} \right)$ and $K_{ij}^{(3,1)} = \frac{1}{\varepsilon} \left(M[F_{ij}^{(3,1)}] - F_{ij}^{(3,1)} \right)$;
- Calculate $F_{ij}^{(2,2)}$ in the grid node using a procedure similar to the RK2 scheme described in the previous section with time step $c_2\Delta t$. Given $F_{ij}^{(2,2)}$, one can calculate $F_{ij}^{(3,2)}$ by interpolation from $F_j^{(2,2)}$ in $\tilde{x}^{(6)} = x_i - (1 - c_2)v_j\Delta t$ and then evaluate $K_{ij}^{(3,2)} = \frac{1}{\varepsilon} \left(M[F_{ij}^{(3,2)}] - F_{ij}^{(3,2)} \right)$ along the third characteristic;
- Now one can update f_{ij}^{n+1} taking into account that the method is stiffly accurate, using the properties of the collision invariants to solve the implicit step,

$$f_{ij}^{n+1} = F_{ij}^{(3,3)} = F_{ij}^{(3,0)} + \Delta t \left(b_1 K_{ij}^{(3,1)} + b_2 K_{ij}^{(3,2)} + b_3 \frac{1}{\varepsilon} (M[F_{ij}^{(3,3)}] - F_{ij}^{(3,3)}) \right).$$

3.1.3. Summary of the Runge–Kutta schemes. Three schemes based on RK are tested in the paper:

- scheme RK2W23: uses WENO23 for the interpolation and RK2, as described above, for time integration;
- scheme RK3W23: uses WENO23 for the interpolation and RK3, as described above, for time integration;
- scheme RK3W35: uses WENO35 for interpolation and RK3 for time integration.

REMARK 3.1. In practice, the Runge–Kutta fluxes can be computed from the internal stages. For example, using RK2, we have

$$K_{ij}^{(1,1)} = \frac{1}{\varepsilon} (M[F_{ij}^{(1,1)}] - F_{ij}^{(1,1)}) = \frac{F_{ij}^{(1,1)} - F_{ij}^{(1,0)}}{a_{11}\Delta t}.$$

The latter expression can be used in the limit $\varepsilon \rightarrow 0$.

3.2. BDF methods. In this section we present a new family of high order semi-Lagrangian schemes, based on BDF. The backward differentiation formulas are implicit linear multistep methods for the numerical integration of ordinary differential equations $y' = g(t, y)$ [15]. Using the linear polynomial interpolating y^n and y^{n-1} one obtains the simplest BDF method (BDF1) that corresponds to backward Euler, used in Section 2.

Here the characteristic formulation of the BGK model, that leads to ordinary differential equations, is approximated by using BDF2 and BDF3 methods, in order to obtain high order approximation. The relevant expressions, under the hypothesis that the time step Δt is fixed, are:

$$\text{BDF2} := y^{n+1} = \frac{4}{3}y^n - \frac{1}{3}y^{n-1} + \frac{2}{3}\Delta t g(t^{n+1}, y^{n+1}), \tag{3.3}$$

$$\text{BDF3} := y^{n+1} = \frac{18}{11}y^n - \frac{9}{11}y^{n-1} + \frac{2}{11}y^{n-2} + \frac{6}{11}\Delta t g(t^{n+1}, y^{n+1}). \tag{3.4}$$

The stability properties of these schemes are adequate to treat the stiffness of the BGK operator even in the limit $\varepsilon \rightarrow 0$ [15].

Here we apply the BDF methods along the characteristics.

3.2.1. BDF2. The numerical time integration of the first equation in (2.1), to the second order accuracy adopting BDF strategy, is easily obtained using (3.3):

$$\text{BDF2} := f_{i,j}^{n+1} = \frac{4}{3}f_{i,j}^{n,1} - \frac{1}{3}f_{i,j}^{n-1,2} + \frac{2}{3}\frac{\Delta t}{\varepsilon}(M[f]_{i,j}^{n+1} - f_{i,j}^{n+1}), \tag{3.5}$$

where $f_{i,j}^{n-(s-1),s} \simeq f(t^{n-(s-1)}, x_i - s v_j \Delta t, v_j)$ can be computed by suitable reconstruction from $\{f_{i,j}^{n-(s-1)}\}$; high order reconstruction will be needed for BDF2 and BDF3 schemes, and again we make use of WENO techniques [6] for accurate non-oscillatory reconstruction.

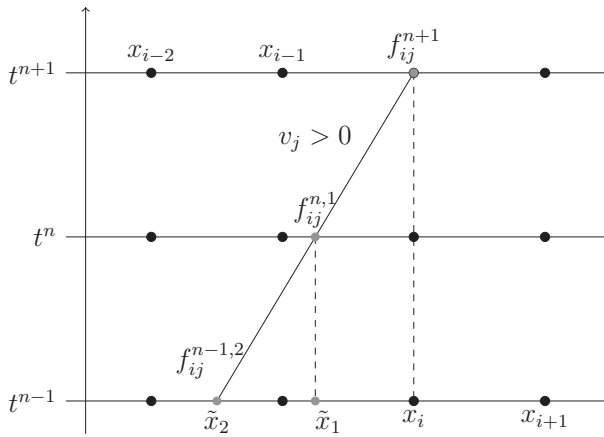


FIG. 3.3. Representation of the BDF2 scheme. The black circles denote grid nodes, the gray ones the points where interpolation is needed.

To compute the solution $f_{i,j}^{n+1}$ from equation (3.5), also in this case one has to solve a non-linear implicit equation. We can act as previously done for the backward Euler method, by taking advantage of the properties of the collision invariants. Thus we multiply both sides of equation (3.5) by ϕ_j and sum over j , getting

$$\sum_j \left(f_{i,j}^{n+1} - \frac{4}{3}f_{i,j}^{n,1} + \frac{1}{3}f_{i,j}^{n-1,2} \right) \phi_j = \frac{2\Delta t}{3\varepsilon} \sum_j (M[f]_{i,j}^{n+1} - f_{i,j}^{n+1}) \phi_j,$$

which implies that

$$\sum_j (f_{i,j}^{n+1}) \phi_j = \sum_j \left(\frac{4}{3}f_{i,j}^{n,1} - \frac{1}{3}f_{i,j}^{n-1,2} \right) \phi_j,$$

so in equation (3.5) we can compute $M[f_{i,j}^{n+1}]$ with the usual procedure, adopting the approximated macroscopic moments

$$(\rho_i^{n+1}, (\rho u)_i^{n+1}, E_i^{n+1}) = m \left[\frac{4}{3}f_{i,\cdot}^{n,1} - \frac{1}{3}f_{i,\cdot}^{n-1,2} \right]. \tag{3.6}$$

Once the Maxwellian $M[f_{ij}^{n+1}]$ is computed, the distribution function value f_{ij}^{n+1} can be easily obtained from schemes (3.5) for BDF2. The procedure for BDF2 is sketched in Figure 3.3 and described in the following algorithm.

Algorithm (BDF2)

- Calculate $f_{ij}^{n-1,2} = \tilde{f}(t^{n-1}, \tilde{x}_2 = x_i - 2v_j \Delta t, v_j)$, $f_{ij}^{n,1} = \tilde{f}(t^n, \tilde{x}_1 = x_i - v_j \Delta t, v_j)$ by interpolation from $f_{.j}^{n-1}$ and $f_{.j}^n$, respectively;
- Compute the Maxwellian $M[f_{ij}^{n+1}]$ using (3.6) and upgrade the numerical solution f_{ij}^{n+1} .

A similar algorithm is obtained using BDF3, as we will see in the next subsection.

3.2.2. BDF3. The numerical solution of the BGK equation in (2.1), to the third order of accuracy using BDF methods, is obtained applying (3.4) to the first equation in (2.1):

$$f_{i,j}^{n+1} = \frac{18}{11} f_{ij}^{n,1} - \frac{9}{11} f_{ij}^{n-1,2} + \frac{2}{11} f_{ij}^{n-2,3} + \frac{6}{11} \frac{\Delta t}{\varepsilon} (M[f]_{ij}^{n+1} - f_{ij}^{n+1}), \tag{3.7}$$

where $f_{i,j}^{n-(s-1),s}$ can be computed by suitable reconstruction from $\{f_{.j}^{n-(s-1)}\}$.

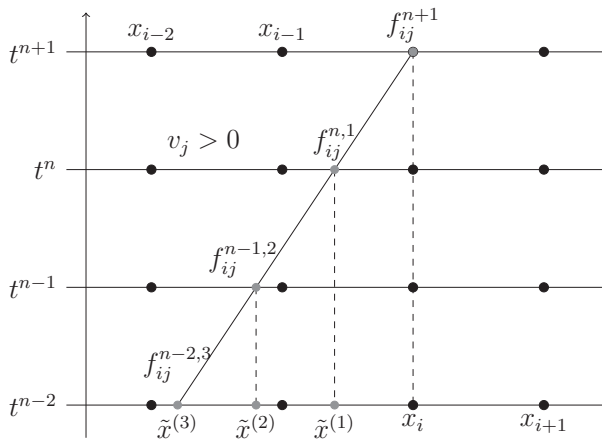


FIG. 3.4. Representation of the BDF3 scheme. The black circles denote grid nodes, the gray ones the points where interpolation is needed.

To compute the solution f_{ij}^{n+1} from equation (3.7) we need again to take moments of such equation

$$\sum_j \left(f_{ij}^{n+1} - \frac{18}{11} f_{ij}^{n,1} + \frac{9}{11} f_{ij}^{n-1,2} - \frac{2}{11} f_{ij}^{n-2,3} \right) \phi_j = \frac{6\Delta t}{11\varepsilon} \sum_j (M[f]_{ij}^{n+1} - f_{ij}^{n+1}) \phi_j,$$

which implies that

$$\sum_j (f_{ij}^{n+1}) \phi_j = \sum_j \left(\frac{18}{11} f_{ij}^{n,1} - \frac{9}{11} f_{ij}^{n-1,2} + \frac{2}{11} f_{ij}^{n-2,3} \right) \phi_j,$$

so in equation (3.7) we can compute $M[f_{ij}^{n+1}]$ with the usual procedure, adopting the approximated macroscopic moments

$$(\rho_i^{n+1}, (\rho u)_i^{n+1}, E_i^{n+1}) = m \left[\frac{18}{11} f_i^{n,1} - \frac{9}{11} f_i^{n-1,2} + \frac{2}{11} f_i^{n-2,3} \right].$$

Once the Maxwellian $M[f_{ij}^{n+1}]$ is computed, the distribution function value f_{ij}^{n+1} can be easily obtained from schemes (3.7) for BDF3. This procedure is sketched in Figure 3.4.

To compute the starting values f_{ij}^1 for BDF2 and f_{ij}^1, f_{ij}^2 for BDF3 we have used, as predictor, Runge–Kutta methods of order 2 and 3, respectively.

3.2.3. Summary of the BDF schemes. Three schemes based on BDF are tested in the paper:

- scheme BDF2W23: uses WENO23 for the interpolation and BDF2, as described above, for time integration;
- scheme BDF3W23: uses WENO23 for the interpolation and BDF2, as described above, for time integration;
- scheme BDF3W35: uses WENO35 for interpolation and BDF3 for time integration.

At variance with Runge–Kutta methods, BDF methods do not need to compute intermediate stage values, and the implicit step for the Maxwellian is solved only once during a time step. Moreover, we have to interpolate in fewer out-of-grid points (for instance, we have to perform only 3 interpolations in a time step using BDF3, versus 6 interpolations needed to advance one time step using a DIRK method of order 3). This makes BDF methods very efficient from a computational point of view.

4. Semi-Lagrangian schemes without interpolation

As we can observe, the cost of the schemes presented above is mainly due to the interpolation, especially when we use high order interpolation techniques.

In order to reduce the computational cost we look for schemes that avoid interpolation. The key idea is to choose discretization parameters in such a way that all the characteristics connect grid points in space. This is obtained, for example, by choosing $\Delta v \Delta t = \Delta x$ (see Figure 4.1).

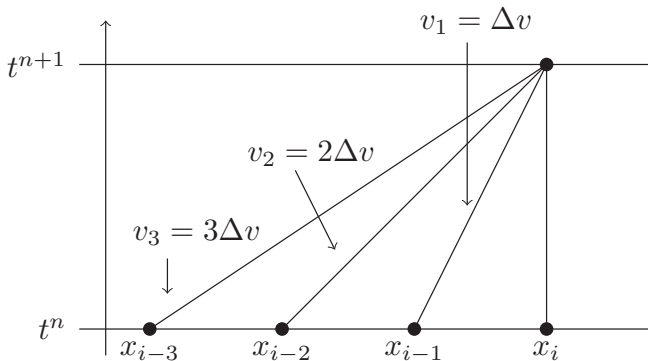


FIG. 4.1. *Implicit first order scheme without interpolation.*

This choice corresponds to solving the equation at each characteristics by implicit Euler, thus resulting in a first order method in time. In order to increase the order of

accuracy one can resort to BDF or RK time discretization. BDF2 and BDF3 can be easily applied in this setting.

The use of higher order RK schemes requires that the stage values lie on the grid as well. This is obtained by imposing $\Delta v \Delta t = s \Delta x$, $s \in \mathbb{N}$. In this case the coefficients of the vector \underline{c} must be multiples of $1/s$. Moreover we need a L-stable scheme. Imposing accuracy and stability constraints on the coefficients of the Butcher's table, together with the fact that coefficients of the vector \underline{c} must be multiples of $1/s$, we obtain some DIRK methods of second and third order, using, respectively, $s = 3$ and $s = 4$.

Using schemes that avoid the interpolation, the choice of the time step is determined by the others discretization steps, and the CFL number is fixed to sN_v . This means that we have very large time step. An example of a Runge-Kutta method that avoids interpolation is the following

$$\begin{array}{c|cc} 1/3 & 1/3 & 0 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4. \end{array}$$

This is a second order method, diagonally implicit and L-stable, because it is stiffly-accurate and A-stable, and allows us to avoid interpolation using $s = 3$.

These schemes are much simpler to implement and therefore each time step can be advanced very efficiently. However they require a very fine grid in space. A comparison with more standard semi-Lagrangian methods that make use of interpolation will be presented in the Section 6 on numerical results.

5. Chu reduction model

These methods have been extended to treat problem in slab geometry, which are 3D in velocity, 1D in space. The technique used is the Chu reduction [8], which, under suitable symmetry assumption, allows the transformation of a 3D equation (in velocity) into a system of two equations 1D (in velocity), to which the schemes previously introduced can be applied.

We consider the application of BGK equation to problems with axial symmetry with respect to an axis (say, $x_1 \equiv x$), in the sense that all transverse spatial gradients vanish, and the gas is drifting only in the axial direction. In such cases, distribution functions $f(x, \underline{v}, t)$ depend on the full velocity vector \underline{v} (i.e., molecular trajectories are three-dimensional) but dependence on the azimuthal direction around the symmetry axis is such that all transverse components of the macroscopic velocity \underline{u} vanish (i.e., $u_2 = u_3 = 0$).

Let us introduce the new unknowns

$$g_1(x, v, t) = \int_{\mathbb{R}^2} f(x, \underline{v}, t) dv_2 dv_3, \quad g_2(x, v, t) = \int_{\mathbb{R}^2} (v_2^2 + v_3^2) f(x, \underline{v}, t) dv_2 dv_3, \quad (5.1)$$

each depending only on one space and one velocity variable $v = v_1$. Multiplication of (1.1) by 1 and $(v_2^2 + v_3^2)$ and integration with respect to $(v_2, v_3) \in \mathbb{R}^2$ yields then the following system of BGK equations for the unknown vector $g = (g_1, g_2)$, coupled with initial conditions

$$\frac{\partial g_i}{\partial t} + v \frac{\partial g_i}{\partial x} = \frac{1}{\varepsilon} (M[f]_i - g_i), \quad (x, v, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+, \quad (5.2)$$

$$g_i(x, v, 0) = g_{i,0}(x, v), \quad i = 1, 2.$$

The BGK system (5.2) describes a relaxation process towards the vector function $(M[f]_1, M[f]_2)$, which is obtained by Chu transform of (1.2) with $N=3$ and has the form

$$(M[f]_1, M[f]_2) = (M[f]_1, 2RT M[f]_1),$$

where

$$M[f]_1 = \frac{\rho(t, x)}{\sqrt{2\pi RT(t, x)}} \exp\left(-\frac{(v - u(t, x))^2}{2RT(t, x)}\right).$$

The macroscopic moments of the distribution function f , needed to evaluate $M[f]_1$ are given in terms of g_1 and g_2 as:

$$\begin{aligned} \rho &= \int_{\mathbb{R}} g_1 dv, & u &= \frac{1}{\rho} \int_{\mathbb{R}} v g_1 dv, \\ 3RT &= \frac{1}{\rho} \left[\int_{\mathbb{R}} (v - u)^2 g_1 dv + \int_{\mathbb{R}} g_2 dv \right]. \end{aligned}$$

The following relation will be useful to solve the implicit step:

$$\int_{\mathbb{R}} (v - u)^2 (M[f]_1 - g_1) dv + \int_{\mathbb{R}} (M[f]_2 - g_2) dv = 0. \tag{5.3}$$

Indeed

$$3RT\rho = \int_{\mathbb{R}} (v - u)^2 M[f]_1 dv + 2RT \int_{\mathbb{R}} M[f]_1 dv,$$

and also

$$3RT\rho = \int_{\mathbb{R}} (v - u)^2 g_1 dv + \int_{\mathbb{R}} g_2 dv.$$

Taking the difference we obtain (5.3).

The discrete version of the first order implicit scheme (in a similar way one can extend to high order schemes) of (5.2) is

$$g_{s,ij}^{n+1} = \tilde{g}_{s,ij}^n + \frac{\Delta t}{\varepsilon} (M_{s,ij}^{n+1} - g_{s,ij}^{n+1}), \quad s = 1, 2. \tag{5.4}$$

To solve the implicit step we have to compute $m[g_{1,i}^{n+1}]$. The density ρ_i^{n+1} and the momentum $(\rho u)_i^{n+1}$ can be easily computed multiplying the first equation of (5.4) by 1 and v_j and summing over j . In this way we get

$$\rho_i^{n+1} = \Delta v \sum_j \tilde{g}_{1,ij}^n, \quad (\rho u)_i^{n+1} = \Delta v \sum_j v_j \tilde{g}_{1,ij}^n.$$

To obtain the temperature T_i^{n+1} , instead, we have to multiply by $(v_j - u_j)^2$ and by 1 respectively the first and the second equation of (5.4), and than summing over j .

Now, using the discrete analogue of (5.3):

$$\Delta v \sum_j (v_j - u_j)^2 (M_{1,ij}^{n+1} - g_{1,ij}^{n+1}) + \Delta v \sum_j (M_{2,ij}^{n+1} - g_{2,ij}^{n+1}) = 0,$$

one can compute the temperature T_i^{n+1} in this way:

$$3R\rho_i^{n+1}T_i^{n+1} = \Delta v \sum_j (v_j - u_j)^2 \tilde{g}_{1,ij}^n + \Delta v \sum_j \tilde{g}_{2,ij}^n.$$

Once the new moments ρ_i^{n+1} , $(\rho u)_i^{n+1}$, and T_i^{n+1} are computed, we can solve the implicit step and upgrade the numerical solution.

6. Numerical tests

We have considered two types of numerical tests with the purpose of verifying the accuracy (test 1) and the shock capturing properties (test 2) of the schemes. Different values of the Knudsen number, represented by the relaxation parameter ε , have been investigated in order to observe the behavior of the methods varying from the rarefied ($\varepsilon \simeq 1$) to the fluid ($\varepsilon \simeq 10^{-6}$) regime. We use units for temperature such that $R = 1$.

In the first part of the section we consider the 1D model and we explore the choice of the optimal CFL. A comparison between semi-Lagrangian schemes with and without interpolation is also presented. The second part of the section is devoted to the results obtained by the method applied to the 1D space-3D velocity case in slab geometry (Chu reduction).

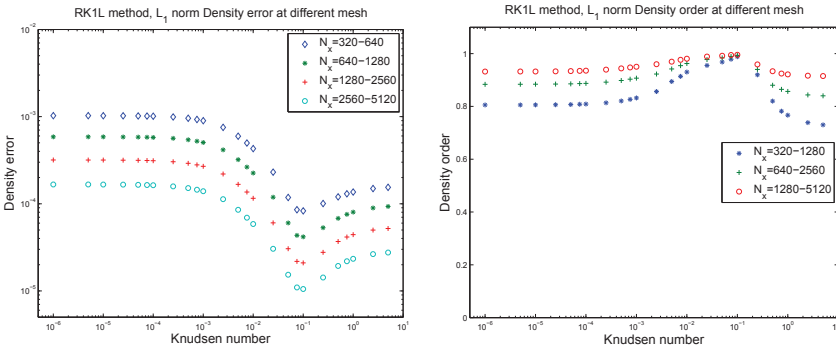


FIG. 6.1. L_1 error and accuracy order of implicit Euler methods coupled with linear interpolation, varying ε , using periodic boundary conditions.

6.1. Test 1: regular velocity perturbation. This test has been proposed in [22]. Initial velocity profile is given by

$$u = 0.1 \exp(-(10x - 1)^2) - 2 \exp(-(10x + 3)^2), \quad x \in [-1, 1].$$

Initial density and temperature profiles are uniform, with constant value $\rho = 1$ and $T = 1$.

The initial condition for the distribution function is the Maxwellian, computed by given macroscopic fields. To check the accuracy order the solution must be smooth. Using periodic or reflective boundary conditions, we observe that some shocks appear in the solution around the time $t = 0.35$, so the accuracy order has been tested using a final time $t_f = 0.32$, which is large enough to reach thermodynamic equilibrium. In all tests $N_v = 20$ velocity points have been used, uniformly spaced in $[-10, 10]$. For the time step, we set $\Delta t = \text{CFL} \Delta x / v_{max}$ and we have used $\text{CFL} = 4$. The spatial domain is $[-1, 1]$.

We have compared the following methods:

- First order implicit Euler coupled with linear interpolation (see Figure 6.1);

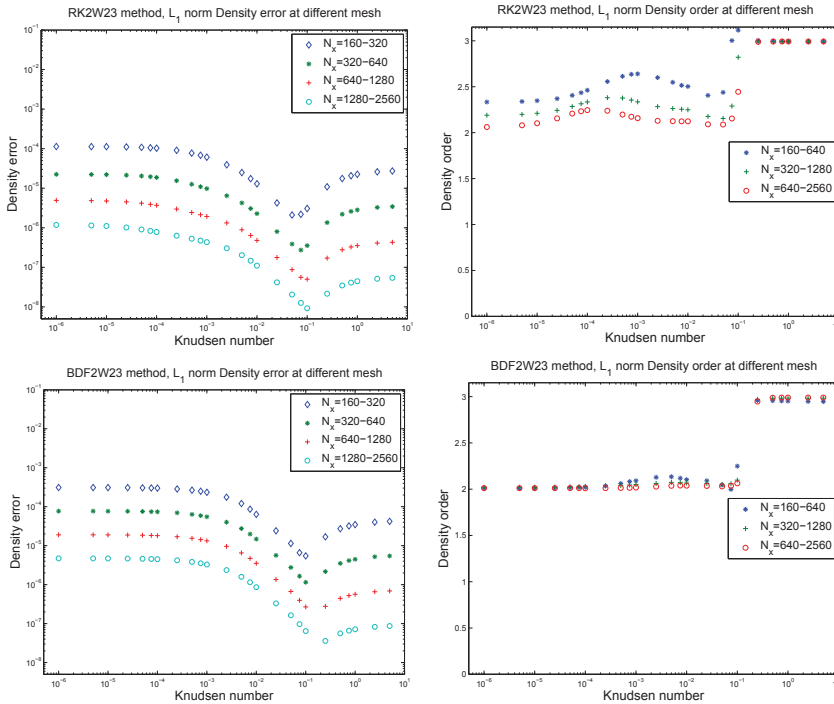


FIG. 6.2. L_1 error and accuracy order of RK2W23 and BDF2W23, varying ε , using periodic boundary conditions.

- RK2W23 and BDF2W23 as second order methods (see Figure 6.2);
- RK3W23 and RK3W35 (see Figure 6.3), BDF3W23 and BDF3W35 (see Figure 6.4) as third order methods;

Figures 6.1–6.4 show the L_1 error and the rate of convergence related to macroscopic density of the schemes using periodic boundary conditions. The same behavior is observed when monitoring the error in mean velocity and in energy.

Remarks.

- In most regimes the order of accuracy is the theoretical one. More precisely all schemes maintain the theoretical order of accuracy in the limit of small Knudsen number, except RK3-based scheme, whose order of accuracy degrades to 2, with both WENO23 and WENO35 interpolation. Some schemes (RK2W23, RK3W35, BDF2W23, BDF3W35) present a spuriously high order of accuracy for large Knudsen number. This is due to the fact that for such large Knudsen number and small final time most error is due to space discretization, which in such schemes is of order higher than time discretization. The most uniform accuracy is obtained by the BDF3W23 scheme.
- Most tests have been conducted with periodic boundary conditions. Similar results are obtained using reflecting boundary conditions (see Figure 6.5).

6.2. Optimal CFL. The semi-Lagrangian nature of the scheme allows us to avoid the classical CFL stability restriction. In this way, one can use large CFL numbers in order to obtain larger time steps, thus lowering the computational cost. How much can we increase the CFL number without degrading the accuracy?

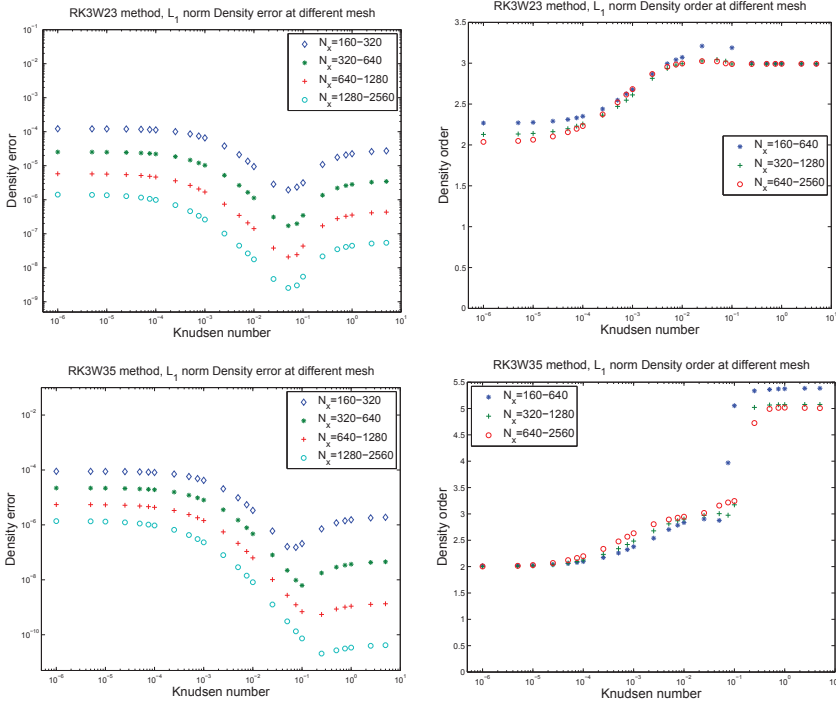


FIG. 6.3. L_1 error and accuracy order of RK3W23 and RK3W35, varying ε , using periodic boundary conditions.

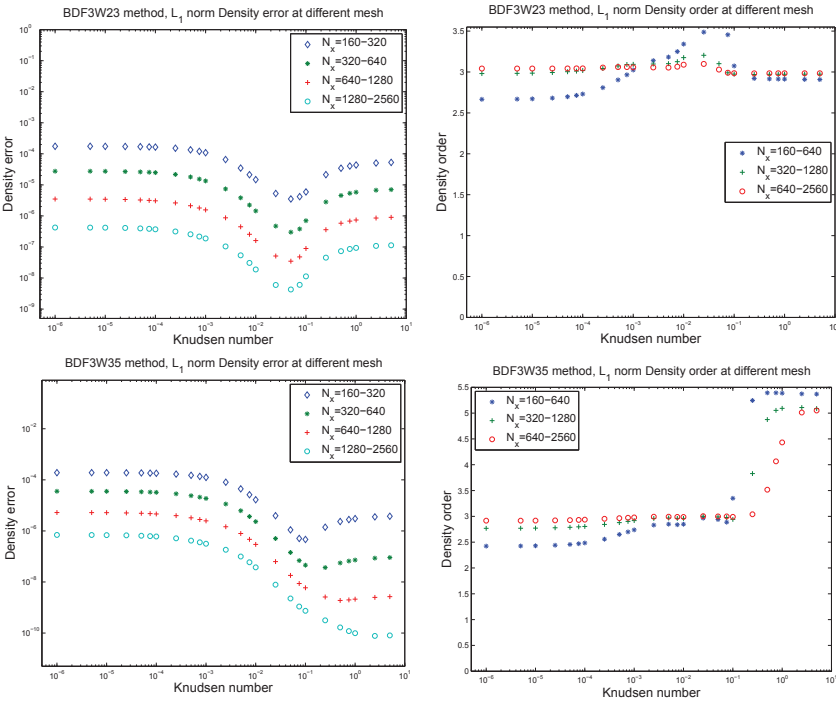


FIG. 6.4. L_1 error and accuracy order of BDF3W23 and BDF3W35, varying ε , using periodic boundary conditions.

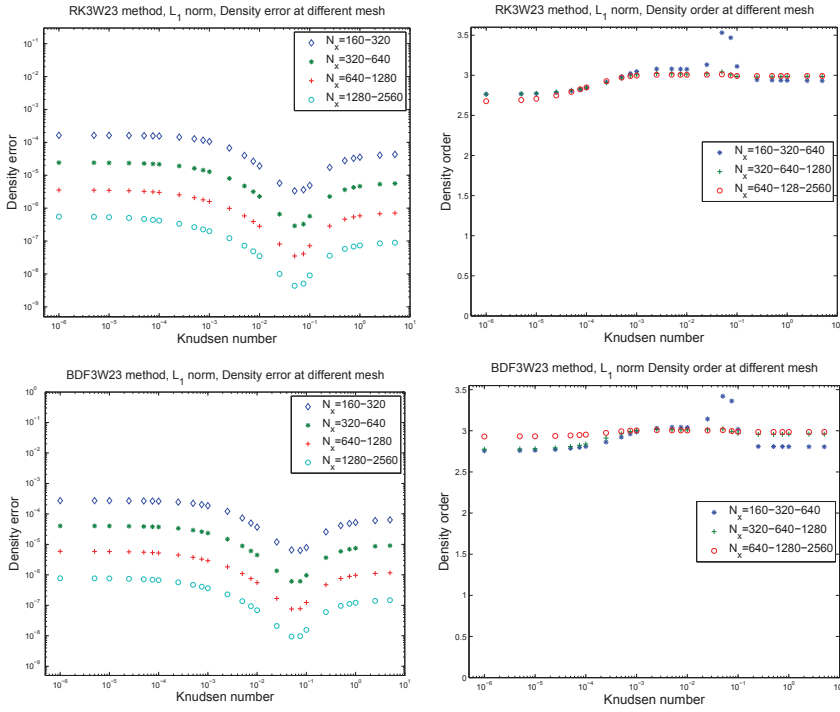


FIG. 6.5. L_1 error and accuracy order of RK3 and BDF3 methods coupled with WENO23, varying ε , using reflective boundary conditions.

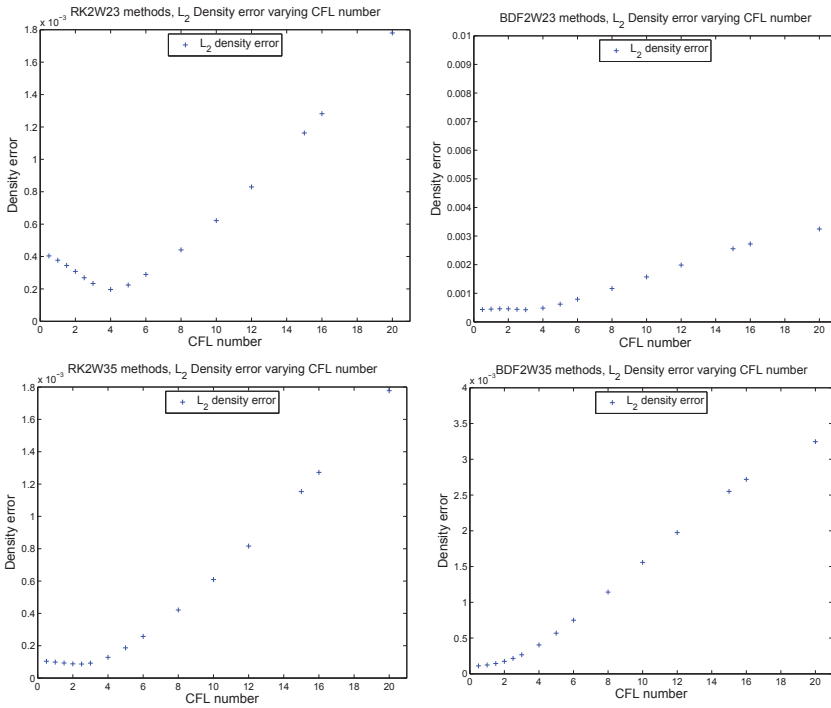


FIG. 6.6. Optimal CFL number. Left: RK2; Right: BDF2. From top to bottom: WENO23, WENO35.

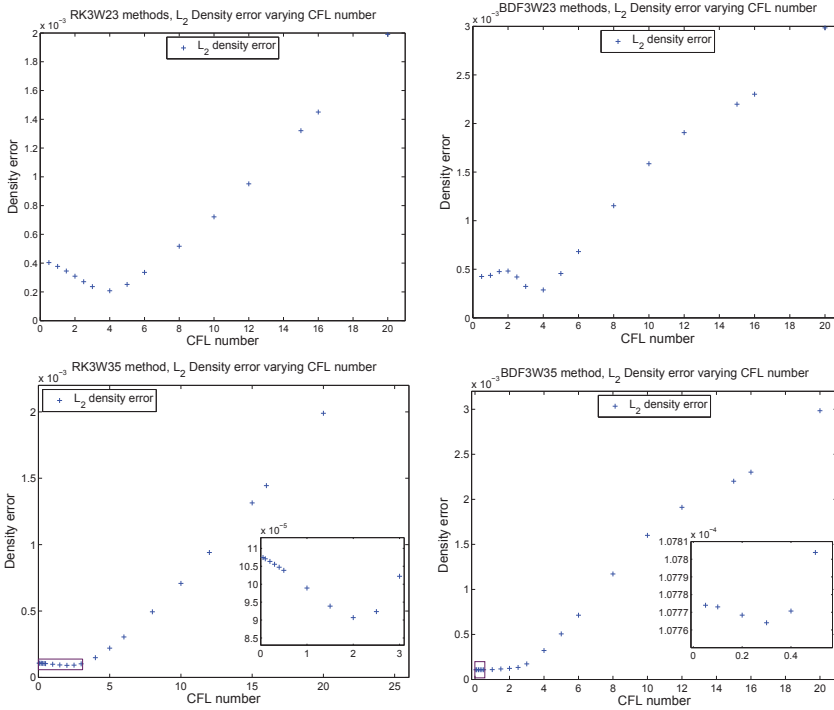


FIG. 6.7. Optimal CFL number. Left: RK3; Right: BDF3. From top to bottom: WENO23, WENO35.

Consistency analysis of semi-Lagrangian schemes [10] shows that the error is composed by two part: one depending on the time integration and the other depending on the interpolation. Therefore, if we use a small CFL number, the time step will be small and the error will be mainly due to the interpolation.

On the other hand, if we use a big CFL number, the error will be mainly due to the time integration. This argument leads us to think that there is an optimal value of the CFL number, that allows us to minimize the error. Figures 6.6 and 6.7 show this behavior. Each picture shows the L_2 error of the macroscopic density of the previous smooth initial data, varying the CFL number from 0.05 to 20. The grid of the CFL values is not uniform because we want to work with constant time step until the final time, that for this test is set to 0.3. The error is computed using two numerical solutions, obtained with $N_x = 160$ and $N_x = 320$. The Knudsen number is fixed to the value 10^{-4} .

When the accuracy in space is not much larger than in time, as in the case of RK2W23, RK3W23, BDF3W23, an evident optimal CFL number appears, since in such cases interpolation error and time discretization error balance out.

If space discretization is much more accurate than time discretization, the optimal CFL number decreases. Note that with the same formal order of accuracy, the optimal CFL number is larger for schemes based on RK than for schemes based on BDF, because RK based schemes have a smaller error constant.

6.3. Test 2: Riemann problem. This test allows us to evaluate the capability of our class of schemes in capturing shocks and contact discontinuities. In particular, we are interested in the behavior of the schemes in the fluid regime. Here we illustrate the results obtained for moments, i.e., density, mean velocity and temperature profiles, for

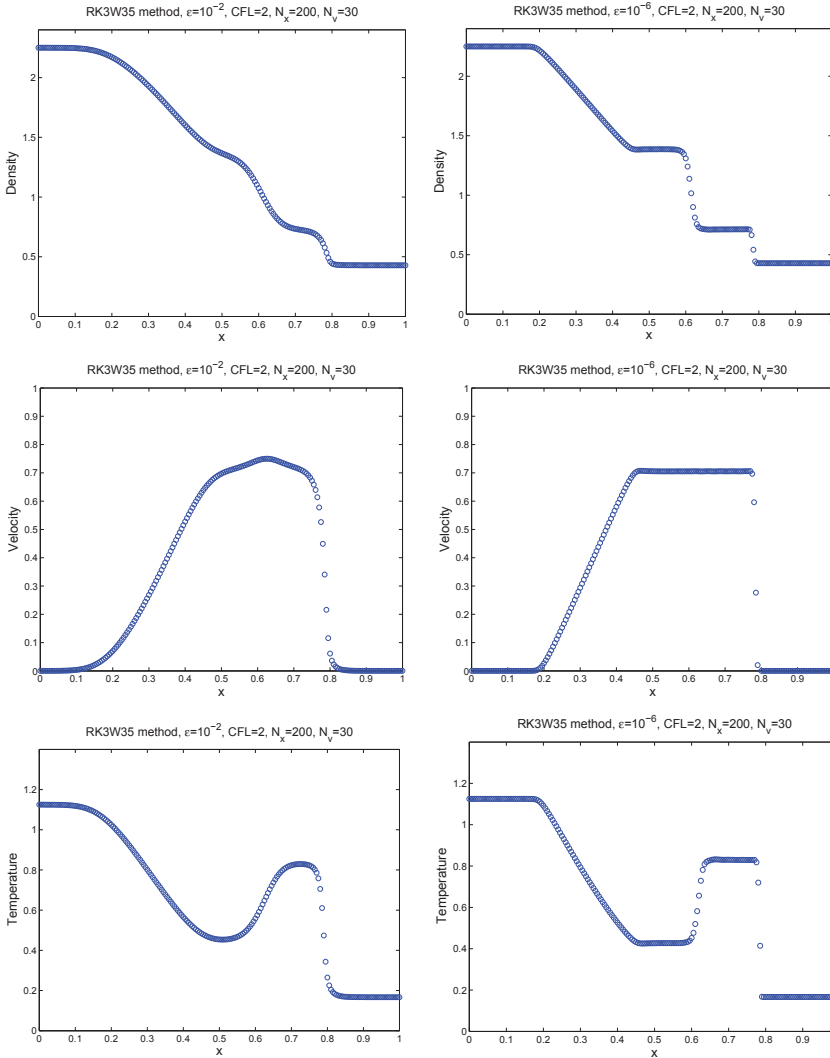


FIG. 6.8. *RK3W35* scheme. Riemann problem in 1D space and velocity case. Left: $\varepsilon = 10^{-2}$; Right: $\varepsilon = 10^{-6}$. From top to bottom: density, velocity and temperature.

$\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-6}$ (see Figures 6.8 and 6.9). The spatial domain chosen is $[0,1]$ and the discontinuity is taken at $x = 0.5$. The initial condition for the distribution function is the Maxwellian computed with the following moments: $(\rho_L, u_L, T_L) = (2.25, 0, 1.125)$, $(\rho_R, u_R, T_R) = (3/7, 0, 1/6)$. Free-flow boundary conditions are assumed. The final time is 0.16. These tests have been performed using $N_v = 30$ velocity nodes, uniformly spaced in $[-10, 10]$.

As shown in Figures 6.8 and 6.9, the schemes are able to capture the fluid dynamic limit for very small values of the relaxation time, relevant to regimes in which the evolution of the moments is governed by the Euler equations.

6.4. Semi-Lagrangian schemes without interpolation. As discussed in Section 4, semi-Lagrangian schemes avoiding interpolation are very advantageous from

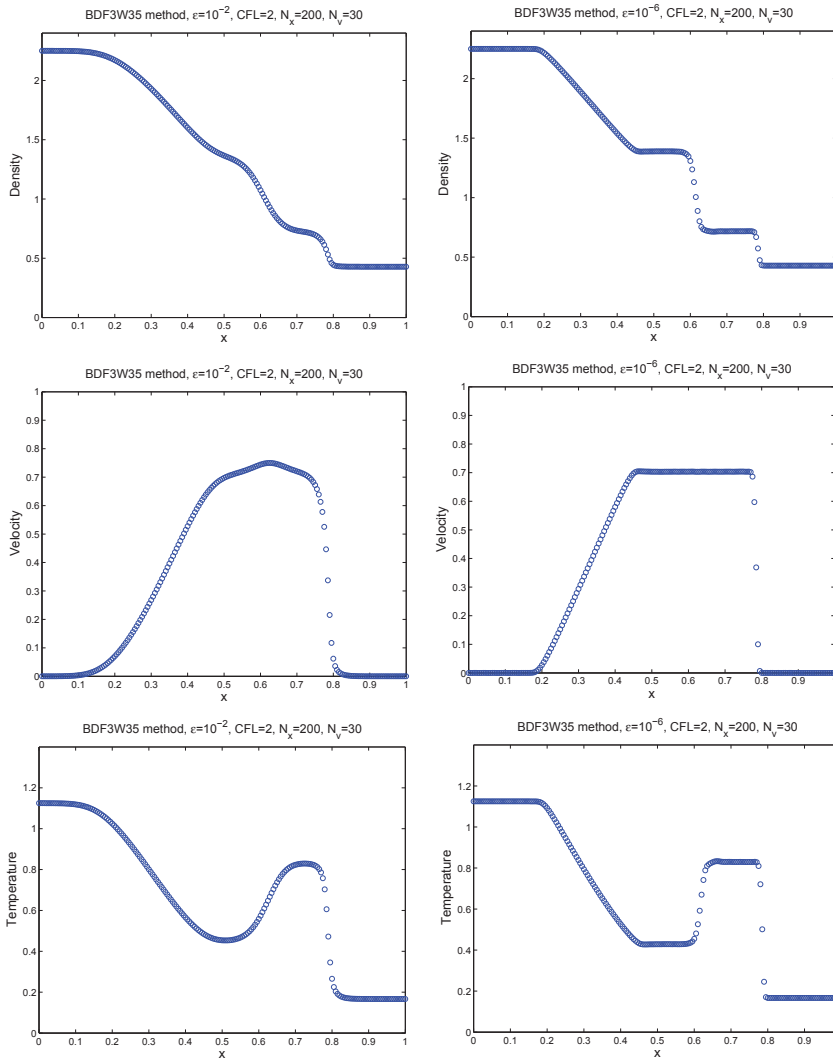


FIG. 6.9. *BDF3W35 Riemann problem in 1D space and velocity case. Left: $\varepsilon = 10^{-2}$; Right: $\varepsilon = 10^{-6}$. From top to bottom: density, velocity and temperature.*

a computational point of view. In Figure 6.10 we compare the CPU time and the L_1 error of the schemes with and without interpolation. At third order of accuracy the relation between CPU time and error is better for the scheme without interpolation using $N_v = 20$. The relative effectiveness of such schemes with respect to the ones that require interpolation decreases when increasing the number of velocities. However, these results are just indicative, as the schemes should be implemented efficiently.

6.5. Numerical results: Chu reduction. Also for the problem 3D in velocity we have considered two numerical tests, that are aimed at verifying the accuracy and the shock capturing properties of the schemes. Different values of the Knudsen number have been investigated in order to observe the behavior of the methods varying from the rarefied to the fluid regime.

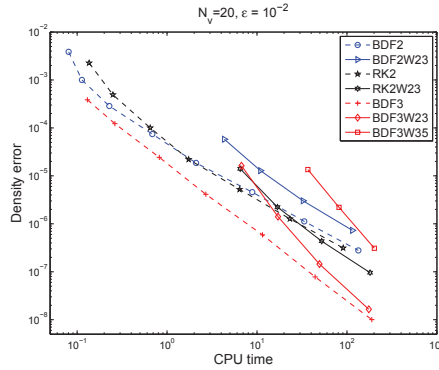


FIG. 6.10. CPU time and L_1 error varying N_x .

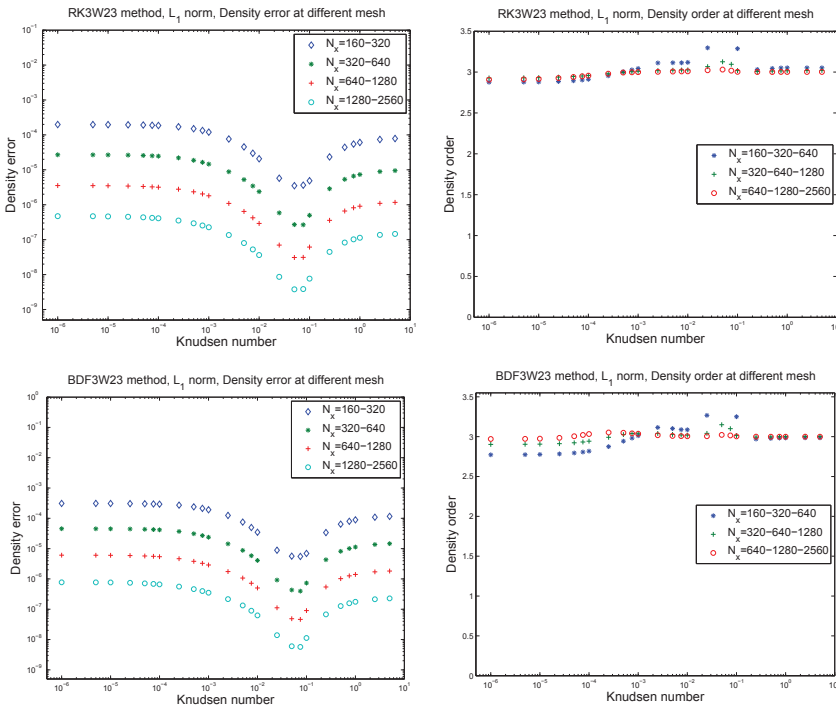


FIG. 6.11. L_1 error and accuracy order of RK3 and BDF3 methods coupled with WENO23, varying ε , using reflective boundary conditions related to the 1D space-3D velocity problem.

The initial data for test 1 is the same as the corresponding test problem 1D in velocity, whereas the data for the second one is different. For the Riemann problem in this case, the initial condition for the distribution function is again a Maxwellian, having now the following initial macroscopic moments: $(\rho_L, u_L, T_L) = (1, 0, 5/3)$, $(\rho_R, u_R, T_R) = (1/8, 0, 4/3)$. As in the previous cases, free-flow boundary conditions are imposed. The final time is 0.25. This test has been performed using $N_v = 30$ velocity nodes uniformly spaced in $[-10, 10]$.

We will show only the order of accuracy related to the schemes RK3W23 and BDF3W23 (Figure 6.11) using reflective boundary conditions, since we get essentially the same

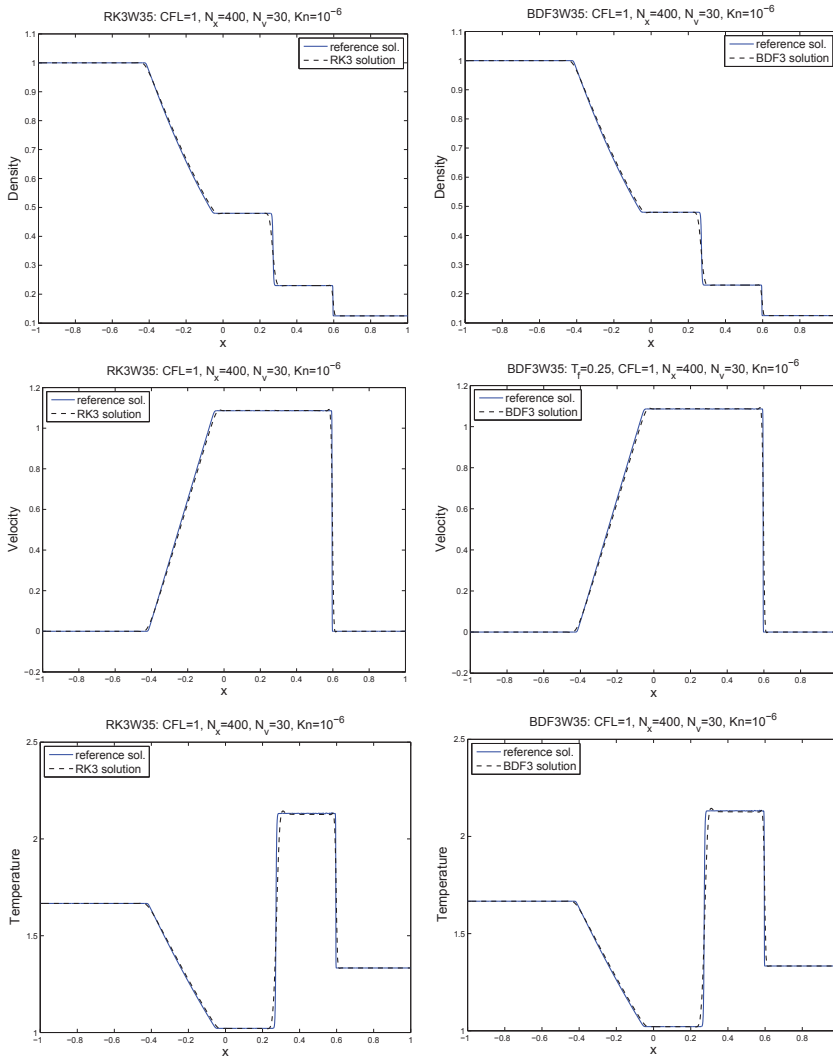


FIG. 6.12. 1D space - 3D velocity case. Comparison with the gas dynamics solution. Left: RK3W35; Right: BDF3W35. From top to bottom: density, velocity and temperature.

results of the 1D case. In this test $CFL=2$ and the final time is 0.4. Regarding the Riemann problem we will show a comparison with the solution of the gas dynamics, for $\varepsilon=10^{-6}$, see Figure 6.12. As shown by the results, also in this case the scheme is able to capture the fluid dynamic limit for very small values of the relaxation time, relevant to regimes in which the evolution of the moments is governed by the Euler equations.

7. Conclusions

This paper presents high order shock capturing semi-Lagrangian methods for the numerical solutions of BGK-type equations.

The methods are based on L-stable schemes for solution of the BGK equations along the characteristics, and our numerical tests show that they are asymptotic preserving, namely they capture the solution of the underlying fluid dynamic limit. This is evident

for instance in Figure 6.12, where the macroscopic moments of f for small values of ε ($\varepsilon = 10^{-6}$) and the numerical solution of Euler equations are indistinguishable.

The semi-Lagrangian nature of these schemes allows us to avoid the classical CFL restriction on the time step. In this sense the schemes are competitive with the IMEX approach [22], even if, in general, interpolation is required.

Two families of schemes are presented, which differ for the choice of the time integrator: Runge–Kutta or BDF. A further distinction concerns space discretization: some schemes are based on high order reconstruction, while others are constructed on the lattice in phase space, thus requiring no space interpolation.

Numerical experiments show that schemes without interpolation can be cost-effective, especially for problems that do not require a fine mesh in velocity. In particular, BDF3 without interpolation appears to have the best performance in most tests. Such schemes can be seen as an evolution, towards high accuracy, of the methods used in [12], where the authors studied applications of GPUs to simulations of rarefied gases using BGK models. Schemes with simple patterns are suitable for implementation by GPUs. In this regard, semi-Lagrangian schemes avoiding interpolation seem very suitable for this kind of simulations.

Future plans include extension of such schemes to problems in several space dimensions and treatment of more general boundary conditions. Moreover, applications of these schemes to the numerical approximation of BGK equations for mixtures of rarefied gases, either inert [2] or reactive [13, 5], are in progress.

Appendix A. In order to obtain high order accuracy and to ensure the shock capturing properties of the proposed schemes near the fluid regime, a suitable non-linear reconstruction technique for the computation of \tilde{f}_{ij}^n is required. ENO (essentially non-oscillatory) and WENO (weighted ENO) methods [28] provide the desired high accuracy and non-oscillatory properties. Both methods are based on the reconstruction of piecewise smooth functions by choosing the interpolation points on smooth the side of the function. In ENO methods these points are chosen according to the magnitude of the divided differences evaluated by two candidate stencils. In WENO methods the different polynomials defined on the stencils are weighted in such a way that in a smooth region high order is recovered, and in presence of singularity the smooth side is taken. In the next two paragraphs we briefly recall the pointwise WENO reconstruction described in [6].

A.1. A general form of WENO interpolation. Given a set of values $V = \{v_j\}_{j \in I}$ of a function v on a space grid $x_j, j \in I$, the aim is to obtain an accurate reconstruction of v at any point of the space interval.

To construct a WENO interpolation of degree $2n - 1$ on the interval $[x_j, x_{j+1}]$, we start from the Lagrange polynomial built on the stencil $S = \{x_{j-n+1}, \dots, x_{j+n}\}$ and written in the form:

$$Q(x) = \sum_{k=1}^n C_k(x) P_k(x), \quad (\text{A.1})$$

where the linear weights C_k are polynomials of degree $n - 1$ and the P_k are polynomials of degree n interpolating V on the stencil $S_k = \{x_{j-n+k}, \dots, x_{j+k}\}, k = 1, \dots, n$. The linear weights C_k have to satisfy, due to (A.1) and to the definition of the polynomials P_k , the following properties: they should vanish at the nodes outside S_k , and in the nodes of S the non-zero weights should have unit sum [6]. In the final reconstruction, in order to guarantee non-oscillatory behavior, the linear weights are replaced by the non-linear

weights ω_k , which are constructed in order to obtain an approximation of the highest degree according to the smoothness of v . The regularity of v is checked by means of the smoothness indicators β_k , whose expression is:

$$\beta_k = \sum_{l=1}^n \int_{x_j}^{x_{j+1}} \Delta x^{2l-1} (P_k^{(l)})^2 dx. \tag{A.2}$$

The non-linear weights are defined as

$$\omega_k(x) = \frac{\alpha_k(x)}{\sum_l \alpha_l(x)}, \tag{A.3}$$

where

$$\alpha_k(x) = \frac{C_k(x)}{(\beta_k + \delta)^2}, \tag{A.4}$$

with δ a properly small parameter, usually of the order of 10^{-6} . Last, the WENO reconstruction of the values $V = \{v_j\}_{j \in I}$ reads

$$I[V](x) = \sum_{k=1}^n \omega_k(x) P_k(x).$$

A.2. Second-third order WENO interpolation (WENO23). To construct a third order interpolation we start from two polynomials of degree two, such that

$$I[V](x) = \omega_L P_L(x) + \omega_R P_R(x),$$

where $P_L(x)$ and $P_R(x)$ are second order polynomials relevant to nodes x_{j-1}, x_j, x_{j+1} and x_j, x_{j+1}, x_{j+2} , respectively. The two linear weights C_L and C_R are first degree polynomials in x , and according to the general theory outlined thus far, they read as

$$C_L = \frac{x_{j+2} - x}{3\Delta x}, \quad C_R = \frac{x - x_{j-1}}{3\Delta x};$$

the expressions of α_L , α_R , ω_L , and ω_R may be easily recovered from the general form.

The smoothness indicators have the following explicit expressions

$$\beta_L = \frac{13}{12} v_{j-1}^2 + \frac{16}{3} v_j^2 + \frac{25}{12} v_{j+1}^2 - \frac{13}{3} v_{j-1} v_j + \frac{13}{6} v_{j-1} v_{j+1} - \frac{19}{3} v_j v_{j+1},$$

$$\beta_R = \frac{13}{12} v_{j+2}^2 + \frac{16}{3} v_{j+1}^2 + \frac{25}{12} v_j^2 - \frac{13}{3} v_{j+2} v_{j+1} + \frac{13}{6} v_{j+2} v_j - \frac{19}{3} v_j v_{j+1},$$

and α_k and ω_k are computed as in (A.4) and in (A.3), respectively.

A.3. Third-fifth order WENO interpolation (WENO35). To construct a fifth order interpolation we start from three polynomials of third degree:

$$I[V](x) = \omega_L P_L(x) + \omega_C P_C(x) + \omega_R P_R(x),$$

where the third order polynomials $P_L(x)$, $P_C(x)$, and $P_R(x)$ are constructed, respectively, on $x_{j-2}, x_{j-1}, x_j, x_{j+1}$, on $x_{j-1}, x_j, x_{j+1}, x_{j+2}$, and on $x_j, x_{j+1}, x_{j+2}, x_{j+3}$. The weights C_L , C_C , and C_R are second degree polynomials in x , and have the form

$$C_L = \frac{(x - x_{j+2})(x - x_{j+3})}{20\Delta x^2}, \quad C_C = -\frac{(x - x_{j-2})(x - x_{j+3})}{10\Delta x^2},$$

$$C_R = \frac{(x - x_{j-2})(x - x_{j-1})}{20\Delta x^2},$$

while the smoothness indicators β_C and β_R have the expressions

$$\begin{aligned} \beta_C &= \frac{61}{45}v_{j-1}^2 + \frac{331}{30}v_j^2 + \frac{331}{30}v_{j+1}^2 + \frac{61}{45}v_{j+2}^2 - \frac{141}{20}v_{j-1}v_j + \frac{179}{30}v_{j-1}v_{j+1} \\ &\quad - \frac{293}{180}v_{j-1}v_{j+2} - \frac{1259}{60}v_jv_{j+1} + \frac{179}{30}v_jv_{j+2} - \frac{141}{20}v_{j+1}v_{j+2}, \\ \beta_R &= \frac{407}{90}v_j^2 + \frac{721}{30}v_{j+1}^2 + \frac{248}{15}v_{j+2}^2 + \frac{61}{45}v_{j+3}^2 - \frac{1193}{60}v_jv_{j+3} + \frac{439}{30}v_jv_{j+2} \\ &\quad - \frac{683}{180}v_jv_{j+3} - \frac{2309}{60}v_{j+1}v_{j+2} + \frac{309}{30}v_{j+1}v_{j+3} - \frac{553}{60}v_{j+2}v_{j+3}, \end{aligned}$$

and β_L can be obtained using the same set of coefficients of β_R in a symmetric way (that is, replacing the indices $j-2, \dots, j+3$ with $j+3, \dots, j-2$) and α_k and ω_k are computed as in (A.4) and in (A.3).

Acknowledgements. This work was supported by MIUR, by the National Group of Scientific Computing (GNCS-INdAM), by the National Group of Mathematical Physics (GNFM-INdAM), and by the Universities of Catania and Parma (Italy).

REFERENCES

- [1] A. Alaia and G. Puppo, *A hybrid method for hydrodynamic-kinetic flow - Part II - Coupling of hydrodynamic and kinetic models*, J. Comp. Phys., 231(16), 5217–5242, 2012.
- [2] P. Andries, K. Aoki, and B. Perthame, *A consistent BGK-type model for gas mixtures*, J. Stat. Phys., 106, 993–1018, 2002.
- [3] U. M. Ascher, S.J. Ruuth, and R.J. Spiteri, *Implicit-Explicit Runge–Kutta methods for time-dependent partial differential equations*, Appl. Numer. Math., 25, 151–167, 1997.
- [4] P.L. Bhatnagar, E.P. Gross, and K. Krook, *A model for collision processes in gases*, Phys. Rev., 94, 511–525, 1954.
- [5] M. Bisi, M. Groppi, and G. Spiga, *Kinetic Bhatnagar-Gross-Krook model for fast reactive mixtures and its hydrodynamic limit*, Phys. Rev. E 81 036327, 1–9, 2010.
- [6] E. Carlini, R. Ferretti, and G. Russo, *A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 27, 1071–1091, 2005.
- [7] C. Cercignani, *The Boltzmann Equation and its Applications*, Springer, New York, 1988.
- [8] C.K. Chu, *Kinetic-theoretic description of the formation of a shock wave*, Phys. Fluids, 8, 12–21, 1965.
- [9] G. Dimarco and R. Loubere, *Towards an ultra efficient kinetic scheme. Part II: The high order case*, J. Comput. Phys., 255, 699–719, 2013.
- [10] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, SIAM, Philadelphia, 2014.
- [11] F. Filbet and G. Russo, *Semi-Lagrangian schemes applied to moving boundary problems for the BGK model of rarefied gas dynamics*, Kinet. Relat. Models, 2, 231–250, 2009.
- [12] A. Frezzotti, G. Ghiroldi, and L. Gibelli, *Solving model kinetic equations on GPUs*, Comput. Fluids, 50, 136–146, 2011.
- [13] M. Groppi and G. Spiga, *A Bhatnagar-Gross-Krook type approach for chemically reacting gas mixtures*, Phys. Fluids, 16, 4273–4284, 2004.
- [14] S. Jin, *Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations*, SIAM J. Sci. Comput., 21, 441–452, 1999.
- [15] E. Hairer and G. Warner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, Springer, Berlin, 14, 1996.

- [16] L.H. Holway, *New statistical models for kinetic theory: methods of construction*, Phys. Fluids, 9, 1658–1673, 1966.
- [17] L. Mieussens, *Discrete-velocity models and numerical schemes for the Boltzmann-BGK equation in plane and axisymmetric geometries*, J. Comput. Phys., 162(2), 429–466, 2000.
- [18] L. Mieussens, *Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics*, Math. Models Meth. Appl. Sci., 10, 1121–1149, 2000.
- [19] L. Pareschi and G. Russo, *Implicit-explicit Runge–Kutta methods and applications to hyperbolic systems with relaxation*, J. Sci. Comput., 25, 129–155, 2005.
- [20] L. Pareschi and G. Russo, *Efficient asymptotic preserving deterministic methods for the Boltzmann equation*, AVT-194 RTO AVT/VKI, Models and Computational Methods for Rarefied Flows, Lecture Series held at the von Karman Institute, Rhode St. Genèse, Belgium, January, 24–28, 2011.
- [21] B. Perthame, *Global existence to the BGK model of Boltzmann equation*, J. Diff. Eqs., 82, 191–205, 1989.
- [22] S. Pieraccini and G. Puppo, *Implicit-explicit schemes for BGK kinetic equations*, J. Sci. Comput., 32, 1–28, 2007.
- [23] S. Pieraccini and G. Puppo, *Microscopically implicit-macroscopically explicit schemes for the BGK equation*, J. Comput. Phys., 231, 299–327, 2012.
- [24] G. Russo, P. Santagati, and S.-B. Yun, *Convergence of a semi-Lagrangian scheme for the BGK model of the Boltzmann equation*, SIAM J. Numer. Anal., 50, 1111–1135, 2012.
- [25] G. Russo and P. Santagati, *A new class of large time step methods for the BGK models of the Boltzmann equation*, arXiv:1103.5247v1, 2011.
- [26] L. Saint-Raymond, *From the BGK model to the Navier–Stokes equations*, Ann. Scient. Ec. Norm. Sup., 36, 271–317, 2003.
- [27] P. Santagati, *High order semi-Lagrangian schemes for the BGK model of the Boltzmann equation*, Department of Mathematics and Computer Science, University of Catania, PhD. thesis, 2007.
- [28] C.W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Lecture Notes in Math. 1697, Papers from the C.I.M.E. Summer School held in Certraro, June 23-28, 1997, A. Quarteroni (Ed.), Springer-Verlag, Berlin, 1998.
- [29] P. Welander, *On the temperature jump in a rarefied gas*, Ark. Fys., 7, 507–553, 1954.
- [30] S. Yun, *Cauchy problem for the Boltzmann-BGK model near a global Maxwellian*, J. Math. Phys., 51, 123514, 2010.