



**HAL**  
open science

## An efficient numerical method for solving the Boltzmann equation in multidimensions

Giacomo Dimarco, Raphaël Loubère, Jacek Narski, Thomas Rey

► **To cite this version:**

Giacomo Dimarco, Raphaël Loubère, Jacek Narski, Thomas Rey. An efficient numerical method for solving the Boltzmann equation in multidimensions. *Journal of Computational Physics*, 2018, 353, pp.46-81. 10.1016/j.jcp.2017.10.010 . hal-01357112

**HAL Id: hal-01357112**

**<https://hal.science/hal-01357112>**

Submitted on 29 Aug 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# An efficient numerical method for solving the Boltzmann equation in multidimensions

Giacomo Dimarco

*Department of Mathematics and Computer Science, University of Ferrara, Via Machiavelli 30, 44122 Ferrara, Italy.*

Raphaël Loubère

*CNRS and Institut de Mathématiques de Toulouse (IMT) Université Paul-Sabatier, Toulouse, France.*

Jacek Narski

*CNRS and Institut de Mathématiques de Toulouse (IMT) Université Paul-Sabatier, Toulouse, France.*

Thomas Rey

*Laboratoire Paul Painlevé, Université Lille 1, Lille, France.*

---

## Abstract

In this paper we deal with the extension of the Fast Kinetic Scheme (FKS) [J. Comput. Phys., Vol. 255, 2013, pp 680-698] originally constructed for solving the BGK equation, to the more challenging case of the Boltzmann equation. The scheme combines a robust and fast method for treating the transport part based on an innovative Lagrangian technique supplemented with fast spectral schemes to treat the collisional operator by means of an operator splitting approach. This approach along with several implementation features related to the parallelization of the algorithm permits to construct an efficient simulation tool which is numerically tested against exact and reference solutions on classical problems arising in rarefied gas dynamic. We present results up to the 3D×3D case for unsteady flows for the Variable Hard Sphere model which may serve as benchmark for future comparisons between different numerical methods for solving the multidimensional Boltzmann equation. For this reason, we also provide for each problem studied details on the computational cost and memory consumption as well as comparisons with the BGK model or the limit model of compressible Euler equations.

*Keywords:* Boltzmann equation, kinetic equations, semi-Lagrangian schemes, spectral schemes, 3D/3D, GPU, CUDA, OpenMP, MPI.

*2000 MSC:* (82B40, 76P05, 65M70, 65M08, 65Y05, 65Y20)

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The Boltzmann equation</b>	<b>3</b>
<b>3</b>	<b>The Fast Kinetic Scheme and the Fast Spectral Scheme</b>	<b>5</b>
3.1	The Fast Kinetic Scheme (FKS) . . . . .	5

---

\*Corresponding author

*Email address:* giacomo.dimarco@unife.it (Giacomo Dimarco)

3.2	Conservation of macroscopic quantities . . . . .	7
3.3	Fast Spectral Scheme (FSS) to discretize the Boltzmann collision operator . . . . .	8
<b>4</b>	<b>Numerical implementation</b>	<b>11</b>
4.1	Algorithm for the transport part . . . . .	12
4.2	Profiling and parallelization strategy . . . . .	12
<b>5</b>	<b>Numerics</b>	<b>14</b>
5.1	Part 1. Numerical results for the space homogeneous case . . . . .	15
5.1.1	Test 1.1. Convergence to equilibrium for the Maxwell molecules in dimension two. . .	15
5.1.2	Test 1.2. Space homogeneous comparison between the BGK and the Boltzmann model. Maxwellian molecules. . . . .	15
5.1.3	Test 1.3. Convergence to equilibrium in dimension three. Hard sphere molecules. . .	17
5.2	Part 2. Numerical results for the one dimensional case in space. . . . .	18
5.2.1	Test 2.1. Numerical convergence of the Boltzmann equation. The two dimensional in velocity Maxwellian molecules case. . . . .	18
5.2.2	Test 2.2. Comparisons between the BGK model and the Boltzmann model. The two dimensional in velocity Maxwellian molecules case. . . . .	22
5.2.3	Test 2.3. Numerical convergence of the Boltzmann equation. The three dimensional in velocity hard sphere molecules case. . . . .	22
5.2.4	Performances . . . . .	22
5.3	Part 3. Numerical results for the space two dimensional case. . . . .	26
5.3.1	Test 3.1. Two dimensional vortex in motion. . . . .	28
5.3.2	Test 3.2. Re-entry test in two dimensions with changing angle of attack in time. . . .	29
5.4	Part 4. Numerical results for the space three dimensional case. . . . .	31
5.4.1	Test 4.1. Three dimensional re-entry test case. . . . .	31
<b>6</b>	<b>Conclusion and perspectives</b>	<b>37</b>

## 1. Introduction

Kinetic equations consider a representation of a gas as particles undergoing instantaneous collisions interspersed with ballistic motion [1, 2]. Nowadays, these models appear in a variety of sciences and applications such as astrophysics, aerospace and nuclear engineering, semiconductors, plasmas related to fusion processes as well as biology, medicine or social sciences. The common structure of such equations consists in a combination of a linear transport term with one or more interaction terms which furnishes the time evolution of the distribution of particles in the phase space. For its nature, the unknown distribution lives in a seven dimensional space, three dimensions for the physical space and three dimensions for the velocity space, plus the time. This makes the problem a real challenge from the numerical point of view, since the computational cost becomes immediately prohibitive for realistic multidimensional problems [2]. Aside from the curse of dimensionality problem, there are many other difficulties which are specific to kinetic equations. We recall two among the most important ones. The computational cost related to the evaluation of the collision operator involving multidimensional integrals which should be solved in each point of the physical space [3, 4]. The second challenge is represented by the presence of multiple scales which requires the development of adapted numerical schemes to avoid the resolution of the stiff dynamics [5, 6, 7, 8, 9, 10, 11] typically arising when dealing with problems with multiple regimes.

Historically, there exists two different approaches which are generally used to tackle kinetic equations from a numerical point of view: deterministic numerical methods such as finite volume, semi-Lagrangian and spectral schemes [2], and, probabilistic numerical methods such as Direct Simulation Monte Carlo (DSMC) schemes [12, 13]. Both methodologies have strengths and weaknesses. While the first could normally reach high order of accuracy, the second are often faster, especially for solving steady problems but, typically, exhibit lower convergence rate and difficulties in describing non stationary and slow motion flows.

In this work, we deal with deterministic techniques. In particular, we focus on semi-Lagrangian approaches [14, 15, 16, 17, 18, 19] for the transport part coupled with spectral methods [20] for the interaction part. Our main goal is to tackle the challenges related to the high dimensionality of the equations and with the difficulties related to the approximation of the collision integral. More in details, we deal with the extension of our recent works [21, 22, 23, 24] which were based on an innovative semi-Lagrangian technique for discretizing the transport part of a kinetic model (FKS method). This technique has been applied solely to the solution of a simple kinetic equation with a relaxation type collision operator, *i.e.* the BGK (Bhatnagar-Gross-Krook) operator [25]. Here we extend it to the case of the full Boltzmann operator [12, 1] and we test his performances up to the six dimensional case. The FKS is based on the classical discrete velocity models (DVM) approach [26, 27, 28, 29]. Successively, in order to overcome the problem of the excessive computational cost, we propose to use a Lagrangian technique which exactly solves the transport step on the entire domain, without reprojecting the solution on the grid at each time step. The FKS approach was shown to be an efficient way to solve linear transport equations, and, it has permitted the simulation of full six dimensions problems on a single processor machine [21]. Unfortunately the solutions computed with this method are limited to a first order in space and time precision. Extension of the method to high order reconstruction is under consideration [30]. Concerning the discretization of the collision operator we rely on Fourier techniques. For the resolution of the Boltzmann integral, these techniques have been first introduced independently by L. Pareschi and B. Perthame in [31] and by A. Bobylev and S. Rjasanow in [32]. Since then, this approach has been investigated by a many authors [3, 4, 33, 34, 35, 36, 32, 37, 38, 39, 40, 41, 42, 43]. In this work we will make use in particular of the fast method described in [44, 3] which has a complexity of the order of  $\mathcal{O}(N^{d_v} \log(N^{d_v}))$  where  $N$  is the number of point in which the velocity space is discretized in one direction and  $d_v$  the dimension of the velocity space. The method preserves mass, and approximates with spectral accuracy momentum and energy.

Combining opportunely the FKS method with the fast spectral approach we have developed a method for solving the Boltzmann equation up to the six dimensional case for unsteady flows. In order to obtain such result we have constructed a parallel version of our algorithm taking advantages of Graphical-Processor-Unit (GPU) under CUDA language. The results presented in this work show that we are nowadays ready and able to use kinetic equations to simulate realistic multidimensional flows. Up to our knowledge this is one of the first examples in literature of solution of the full multidimensional Boltzmann equation by means of deterministic schemes. Our main limitation to run extensive simulations remains at the present moment the lack of memory capacity due to the use of shared memory machines. However, in the present paper, we also report some preliminary results about the extension of our method to deal with distributed-memory computers but we postpone for a future research the detailed analysis of the Message Passive Interface (MPI) version of the algorithm as well as the analysis of its performances.

The article is organized as follows. In Section 2 we recall the Boltzmann equation. In Section 3 we present the Fast Kinetic Scheme and the fast spectral scheme. In Section 4 we detail the aspects related to the implementation of the resulting algorithm necessary to realize an efficient parallel numerical tool. Several tests starting from the 0D× 2D up to the 3D×3D case are studied in details in Section 5. These tests assess the validity of our approach as well as detail all the computational aspects. Conclusions and future works are finally exposed in Section 6.

## 2. The Boltzmann equation

In this section we briefly recall the Boltzmann equation and its main properties, we also recall some related models, *i.e.* the BGK model and the compressible Euler equations, which will be used for numerical comparisons. We refer the reader to [2, 1] for an exhaustive description.

In the kinetic theory of rarefied gases, the non-negative function  $f(x, v, t)$  characterizes the state of the system and it defines the density of particles having velocity  $v \in \mathbb{R}^{d_v}$  in position  $x \in \mathbb{R}^{d_x}$  at time  $t \in \mathbb{R}^+$ , where  $d_x$  is the physical dimension and  $d_v$  the dimension of the velocity space. The time evolution of the particle system is obtained through the equation

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = Q(f). \quad (1)$$

The operator  $Q(f)$ , on the right hand side in equation (1), describes the effects of particle interactions and its form depends on the details of the microscopic dynamic. Independently on the type of microscopic interactions considered, typically the operator is characterized by some conservation properties of the physical system. They are written as

$$\int_{\mathbb{R}^{d_v}} Q(f)\phi(v) dv = 0, \quad (2)$$

where  $\phi(v) = (1, v, |v|^2)$  are commonly called the collision invariants. We denote by

$$U(x, t) = \int_{\mathbb{R}^{d_v}} f(x, v, t)\phi(v) dv \in \mathbb{R}^{2+d_v}$$

the first three moments of the distribution function  $f$ , namely,  $U(x, t) = (\rho, \rho u, E)$ , where  $\rho$  is the density,  $\rho u$  the momentum and  $E$  the energy. Integrating (1) against  $\phi(v)$  yields a system of macroscopic conservation laws

$$\frac{\partial}{\partial t} \int_{\mathbb{R}^{d_v}} f\phi(v) dv + \int_{\mathbb{R}^{d_v}} v \cdot \nabla_x f\phi(v) dv = 0. \quad (3)$$

The above moment system is not closed since the second term involves higher order moments of the distribution function  $f$ . However, using the additional property of the operator  $Q(f)$  that the functions belonging to its kernel satisfy

$$Q(f) = 0 \quad \text{iff} \quad f = M[f], \quad (4)$$

where the Maxwellian distributions  $M[f] = M[f](x, v, t)$  can be expressed in terms of the set of moments  $U(x, t)$  by

$$M[f] = \frac{\rho}{(2\pi T)^{d_v/2}} e^{-\frac{(v-u)^2}{2T}}, \quad \frac{1}{2}d_v\rho T = E - \frac{1}{2}\rho|u|^2 \quad (5)$$

with  $T$  the temperature, one can get a closed system by replacing  $f$  with  $M[f]$  in (3). This corresponds to the set of compressible Euler equations which can be written as

$$\frac{\partial U}{\partial t} + \nabla_x \cdot F(U) = 0, \quad (6)$$

with  $F(U) = \int_{\mathbb{R}^{d_v}} M[f]v\phi(v) dv$ . The simplest operator satisfying (2) and (4) is the relaxation operator [25]

$$Q_{BGK}(f) = \nu(M[f] - f), \quad (7)$$

where  $\nu = \nu(x, t) > 0$  defines the so-called collision frequency. Its values will be specified in the numerical test Section in order for the model to be as close as possible to the Boltzmann model described next. The classical Boltzmann operator reads

$$Q_B(f) = \int_{\mathbb{R}^{d_v}} \int_{\mathbb{S}^{d_v-1}} B(|v - v_*|, \omega) (f(v')f(v'_*) - f(v)f(v_*)) dv_* d\omega, \quad (8)$$

where  $\omega$  is a vector of the unitary sphere  $\mathbb{S}^{d_v-1} \subset \mathbb{R}^{d_v}$ . The post-collisional velocities  $(v', v'_*)$  are given by the relations

$$v' = \frac{1}{2}(v + v_* + |q|\omega), \quad v'_* = \frac{1}{2}(v + v_* - |q|\omega), \quad (9)$$

where  $q = v - v_*$  is the relative velocity. The kernel  $B$  characterizes the details of the binary interactions, it has the form

$$B(|v - v_*|, \cos \theta) = |v - v_*| \sigma(|v - v_*|, \cos \theta) \quad (10)$$

where the scattering cross-section  $\sigma$ , in the case of inverse  $k$ -th power forces between particles, can be written as

$$\sigma(|v - v_*|, \cos \theta) = b_\alpha(\cos \theta)|v - v_*|^{\alpha-1}, \quad (11)$$

with  $\alpha = (k - 5)/(k - 1)$ . The special situation  $k = 5$  gives the so-called Maxwell pseudo-molecules model with

$$B(v, v_*, \omega) = b_0(\cos \theta). \quad (12)$$

For the Maxwell case the collision kernel is independent of the relative velocity. For numerical purposes, a widely used model is the variable hard sphere (VHS) model introduced by Bird [12]. The model corresponds to  $b_\alpha(\cos \theta) = C_\alpha$ , where  $C_\alpha$  is a positive constant, and hence

$$\sigma(|v - v_*|, \cos \theta) = C_\alpha |v - v_*|^{\alpha-1}. \quad (13)$$

In the numerical test Section we will consider the Maxwell molecules case when dealing with a velocity space of dimension  $d_v = 2$  and with the more realistic case of VHS molecules when dealing with the three dimensional case in velocity space:  $d_v = 3$ . For comparison purposes we will also consider the simplified BGK model (7) and the compressible Euler system case (6).

### 3. The Fast Kinetic Scheme and the Fast Spectral Scheme

In this section we detail the numerical scheme. In the first part we discuss the Fast Kinetic Scheme (FKS) and in the second part the Fast Spectral method for the Boltzmann equation. The two solvers are connected by using splitting in time approaches as the ones described in [45].

#### 3.1. The Fast Kinetic Scheme (FKS)

The Fast Kinetic Scheme [21, 22] belongs to the family of so-called semi-Lagrangian schemes [14, 15, 46] which are typically applied to a Discrete Velocity Model (DVM) [26, 29] approximation of the original kinetic equation.

In order to introduce the scheme, let us truncate the velocity space by fixing some given bounds and set a cubic grid in velocity space of  $N$  points with  $\Delta v$  the grid step which is taken equal in each direction. The continuous distribution function  $f$  is then replaced by a vector whose components are assumed to be approximations of the distribution function  $f$  at locations  $v_k$ :

$$\tilde{f}_k(x, t) \approx f(x, v_k, t). \quad (14)$$

The discrete velocity kinetic model consists then of a set of  $N$  evolution equations in the velocity space for  $\tilde{f}_k$ ,  $1 \leq k \leq N$ , of the form

$$\partial_t \tilde{f}_k + v_k \cdot \nabla_x \tilde{f}_k = Q(\tilde{f}_k), \quad (15)$$

where  $Q(\tilde{f}_k)$  is a suitable approximation of the collision operator  $Q(f)$  at location  $k$  discussed in Section (3.3). Observe that, due to the truncation of the velocity space and to the finite number of points with which  $f$  is discretized, the moments of the discrete distribution function  $\tilde{f}_k$  are such that

$$\tilde{U}(x, t) = \sum_k \phi_k \tilde{f}_k(x, t) \Delta v \neq U(x, t), \quad (16)$$

with  $\phi_k = (1, v_k, |v_k|^2)$  the discrete collision invariants, i.e. they are no longer those given by the continuous distribution  $f$ . This problem concerns all numerical methods based on the discrete velocity models and different strategies can be adopted to restore the correct macroscopic physical quantities [29, 27, 28, 34]. At the same time, the spectral approach does not preserve the energy and the momentum of the system even if it approximates these quantities with spectral accuracy [4]. In order to solve this problem of loss of conservation we adopt an  $L_2$  projection technique for the discretized distribution and the discretized collision operator which permits to project the discretized  $\tilde{f}_k$  and  $Q(\tilde{f}_k)$  in the space of the distributions for which the moments are exactly the continuous macroscopic quantities we aim to preserve. We detail this approach in the following paragraph. For lightening the notations we omit the tilde in the rest of the paper, we only use it in the next paragraph in order to introduce the  $L_2$  projection procedure. We will then suppose from now on that all the operations preserves the macroscopic moments if not differently stated.

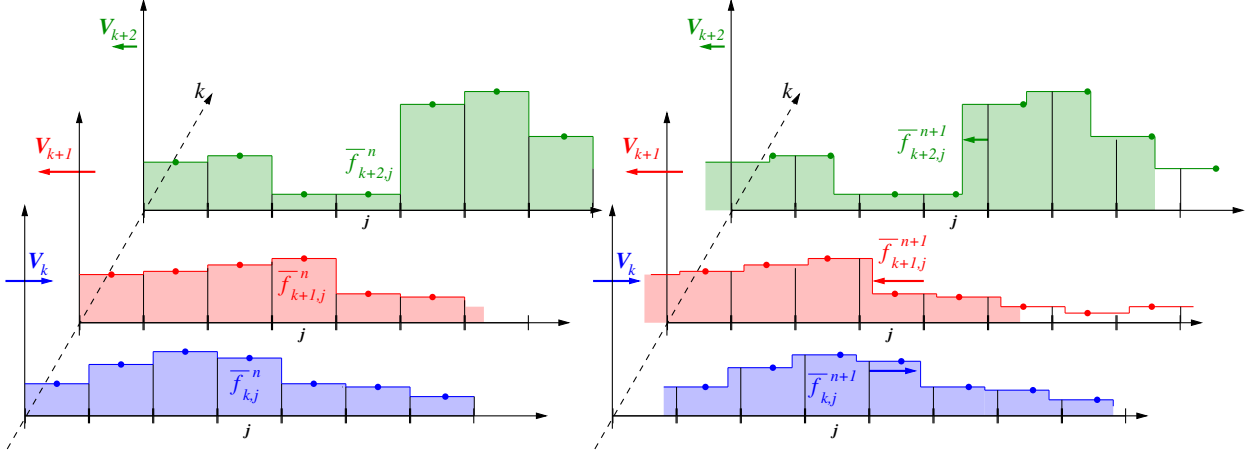


Figure 1: Illustration of the transport scheme for the FKS scheme. Left panel before transport step, right panel after transport step. Each discrete velocity (index  $k$ ) drives its own transport equation with velocity  $v_k$ . The representation of  $f$  is made by means of a piecewise constant function. The shape of the entire function has not changed during the transport but the cell-centered values (bullets) may have.

Let us now introduce a Cartesian uniform grid in the three dimensional physical space made of  $M$  points with  $\Delta x$  a scalar which represents the grid step (the same in each direction) in the physical space. Further we define a time discretization  $t^{n+1} = t^n + \Delta t$  starting at  $t^0$ , where  $\Delta t$  is the time step defined by an opportune CFL condition discussed next. The time index  $n$  varies between 0 and  $N_t$  so that the final time is  $t_{\text{final}} = t^{N_t} = t^0 + N_t \Delta t$ .

Each equation of system (15) is solved by a time splitting procedure. We recall here a first order splitting approach: first a transport step exactly solves the left-hand side, whereas a collision stage solves the right-hand side using the solution from the transport step as initial data:

$$\text{Transport stage} \longrightarrow \partial_t f_k + v_k \cdot \nabla_x f_k = 0, \quad (17)$$

$$\text{Collisions stage} \longrightarrow \partial_t f_k = Q(f_k). \quad (18)$$

*Transport step.* Let  $f_{j,k}^0$  be the point-wise values at time  $t^0$  of the distribution  $f$ ,  $f_{j,k}^0 = f(x_j, v_k, t^0)$ . The idea behind the fast kinetic scheme is to solve the transport stage (17) continuously in space, see Fig. 1 for a sketch in the one dimensional setting. To this aim we define at the initial time the function  $\bar{f}_k^0(x)$  as a piecewise continuous function for all  $x \in \Omega_j$ , where  $\Omega_j = [x_{j-1/2}; x_{j+1/2}]$  and  $\Omega = \bigcup_j \Omega_j$ . Hence starting from data  $\bar{f}_{k,j}^0$  at time index 0, the exact solution of (17) is simply

$$\bar{f}_k^{*,1} = \bar{f}_k^0(x - v_k \Delta t), \quad \forall x \in \Omega. \quad (19)$$

In other words, the entire function  $\bar{f}_k^0$  is advected with velocity  $v_k$  during  $\Delta t$  unit of time and the  $*$  superscript indicates that only the transport step has been solved so far. The extension of this procedure to the generic time step  $n$  gives

$$\bar{f}_k^{*,n+1} = \bar{f}_k^n(x - v_k \Delta t), \quad \forall x \in \Omega, \quad (20)$$

where now, the key observation is that the discontinuities of the piecewise function  $\bar{f}_k^n(x)$  do not lie on the interfaces of two different cells. Instead, the positions of the discontinuities depend entirely on the previous advection step and thus they may be located anywhere in the physical space. This means that if only the linear transport equation has to be solved, this approach gives the exact solution to the equation if the initial data is truly a piecewise constant function initially centered on the spacial mesh.

*Collision step.* The effect of the collisional step is to change the amplitude of  $\bar{f}_k(x)$ . The idea is to solve the collision operator locally on the grid points, and, successively, extend these computed values to the full domain  $\Omega$ . Thus we need to solve the following ordinary differential equation

$$\partial_t f_{j,k} = Q(f_{j,k}), \quad (21)$$

where  $f_{j,k} = f(x_j, v_k, t)$ , for all velocities of the lattice  $k = 1, \dots, N$  and grid points  $j = 1, \dots, M$ . The initial data for solving this equation is furnished by the result of the transport step obtained by (20) at points  $x_j$  of the mesh at time  $t^{n+1} = t^n + \Delta t$ , i.e.  $\bar{f}_k^{*,n+1}(x_j)$ , for all  $k = 1, \dots, N$ , and  $j = 1, \dots, M$ . Then, the solution of (21), locally on the grid points, reads if, for simplicity, a forward Euler scheme is used as

$$f_{j,k}^{n+1} = f_{j,k}^{*,n+1} + \Delta t Q(f_{j,k}^{*,n+1}), \quad (22)$$

where  $f_{j,k}^{*,n+1} = \bar{f}_k^{*,n+1}(x_j)$ . Many different time integrators can be employed to solve this equation. In particular special care is needed in the case in which the equation becomes stiff, refer to [5, 6] for alternative strategies. Since the time integration of the collision term is not the issue considered in this paper, we considered the simplest possible scheme, but the FKS technique remains the same when other time integrators are employed. Equation (22) furnishes a new value for the distribution  $f$  at time  $t^{n+1}$  only in the cell centers of the spacial cells for each velocity  $v_k$ . However, one needs also the value of the distribution  $f$  in all points of the domain in order to perform the transport step at the next time step. Therefore, we define a new piecewise constant function  $\bar{Q}_k$  for each velocity of the lattice  $v_k$  as

$$\bar{Q}_k^{n+1}(x) = Q(f_{j,k}^{*,n+1}), \quad \forall x \text{ such that } \bar{f}_k^{*,n+1}(x) = \bar{f}_k^{*,n+1}(x_j). \quad (23)$$

Said differently we make the fundamental assumption that the shape of  $Q(f_k)$  in space coincides with the one of  $f_k$ . Thanks to the above choice one can rewrite the collision step in term of spatially reconstructed functions as

$$\bar{f}_k^{n+1}(x) = \bar{f}_k(x, t^n + \Delta t) = \bar{f}_k^{*,n+1}(x) + \Delta t \bar{Q}_k^{n+1}(x). \quad (24)$$

This ends one time step of the FKS scheme.

Concerning the transport part of the scheme, the time step  $\Delta t$  is constrained by a CFL like condition of type

$$\Delta t \max_k \left( \frac{|v_k|}{\Delta x} \right) \leq 1 = \text{CFL}. \quad (25)$$

The time step constraint for the collision step depends on the choice of the operator  $Q$ . Since in the numerical test section we used both a BGK and a Boltzmann operator, the time step constraint for the interaction part has been chosen as the minimum time step which gives stability in the solution of the ODE (22) independently on the type of collision operator employed. As observed in [21] the transport scheme is stable for every choice of the time step, being the solution for a given fixed reconstruction performed exactly. Nonetheless the full scheme being based on a time splitting technique, the error is of the order of  $\Delta t$  in the case of first order splitting or of order  $(\Delta t)^q$  for a splitting of order  $q$ . This suggests to take the usual CFL condition for the transport part in order to maintain the time error small enough.

To conclude this Section let us observe that time accuracy can be increased by high order time splitting methods, while spacial accuracy can be increased close to the fluid limit to a nominally second-order accurate scheme by the use of piecewise linear reconstructions of state variables, see the details in [22]. Spacial accuracy for all regime can also be increased by using high order polynomial reconstruction for the distribution  $f$ . This work is in progress and results are discussed in [30].

### 3.2. Conservation of macroscopic quantities

In order to preserve mass, momentum and energy in the scheme, we employ the strategy proposed in [34]. We consider one space cell, the same renormalization of  $f$  should be considered for all spatial cells. This step



is performed at the beginning, i.e.  $t = 0$ . for the distribution  $\tilde{f}(x_j, v_k, t = 0)$  and after each collision step (22) which causes loss of momentum and energy due to the spectral discretization. Let  $\tilde{f} = (\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_N)^T$  be the distribution function vector at  $t = t^n$  at the center of the cell and let  $f = (f_1, f_2, \dots, f_N)^T$  be the unknown corrected distribution vector which fulfills the conservation of moments. Let

$$C_{(d_v+2) \times N} = \begin{pmatrix} (\Delta v)_k^{d_v} \\ v_k (\Delta v)_k^{d_v} \\ |v_k|^2 (\Delta v)_k^{d_v} \end{pmatrix}$$

be a matrix of coefficients depending on the discretization parameters and  $U_{(d_v+2) \times 1} = (\rho \ \rho u \ E)^T$  be the vector of conserved quantities, namely density, momentum and energy. Conservation can be imposed solving a constrained optimization formulation:

$$\begin{aligned} &\text{Given } \tilde{f} \in \mathbb{R}^N, \ C \in \mathbb{R}^{(d_v+2) \times N}, \ \text{and } U \in \mathbb{R}^{(d_v+2) \times 1}, \\ &\text{find } f \in \mathbb{R}^N \text{ such that} \\ &\|\tilde{f} - f\|_2^2 \text{ is minimized subject to the constrain } Cf = U. \end{aligned} \quad (26)$$

The solution of this minimization problem can be analytically obtained by employing the Lagrange multiplier method. Let  $\lambda \in \mathbb{R}^{d_v+2}$  be the Lagrange multiplier vector. The corresponding scalar objective function to be minimized is given by

$$L(f, \lambda) = \sum_{k=1}^N |\tilde{f}_k - f_k|^2 + \lambda^T (Cf - U). \quad (27)$$

Then, by nullifying the derivative of  $L(f, \lambda)$  with respect to  $f_k$  we get

$$f = \tilde{f} + \frac{1}{2} C^T \lambda \quad (28)$$

while the Lagrange multipliers are obtained by solving

$$CC^T \lambda = 2(U - C\tilde{f}). \quad (29)$$

In particular, the above expression says that the value of  $\lambda$  is uniquely determined by  $\lambda = 2(CC^T)^{-1}(U - C\tilde{f})$ . Back substituting  $\lambda$  into (28) finally provides

$$f = \tilde{f} + C^T (CC^T)^{-1} (U - C\tilde{f}). \quad (30)$$

Note that matrix  $C^T(CC^T)^{-1}$  is pre-computed and stored as being constant for each simulation. If a discretization of the Maxwellian distribution  $M[f]$  is needed, as for example when using the BGK model, the same procedure should be applied to the function  $M(x_j, v_k, t^n)$  in order to assure conservation of the first three moments at each instant of time at which the Maxwellian function  $M(x_j, v_k, t^n)$  is invoked. In this case by defining with  $\mathcal{E}[f]$  the approximated Maxwellian and by  $\tilde{\mathcal{E}}[f] = M(x_j, v_k, t^n)$  the pointwise value of the equilibrium distribution, we get mimicking (30)

$$\mathcal{E}[U] = \tilde{\mathcal{E}}[f] + C^T (CC^T)^{-1} (U - C\tilde{\mathcal{E}}[f]). \quad (31)$$

This ends the description of the procedure which permits to conserve macroscopic quantities.

### 3.3. Fast Spectral Scheme (FSS) to discretize the Boltzmann collision operator

The fast spectral discretization of the Boltzmann operator employed in this work is described in this section. To this aim, we focus again on a given cell  $x_j$  at a given instant of time  $t^n$ . The same computation is repeated for all cells  $x_j$ ,  $j = 1, \dots, M$  and times  $t^n$ ,  $n = 0, \dots, N_t$ . Moreover, since the collision operator

acts only on the velocity variable, to lighten the notation in this paragraph, only the dependency on the velocity variable  $v$  is considered for the distribution function  $f$ , i.e.  $f = f(v)$ .

The first step to construct for our spectral discretization is to truncate the integration domain of the Boltzmann integral (8) as done for the distribution  $f$ . As a consequence, we suppose the distribution function  $f$  to have compact support on the ball  $\mathcal{B}_0(R)$  of radius  $R$  centered in the origin. Then, since one can prove that  $\text{supp}(Q(f)(v)) \subset \mathcal{B}_0(\sqrt{2}R)$ , in order to write a spectral approximation which avoid aliasing, it is sufficient that the distribution function  $f(v)$  is restricted on the cube  $[-T, T]^{d_v}$  with  $T \geq (2 + \sqrt{2})R$ . Successively, one should assume  $f(v) = 0$  on  $[-T, T]^{d_v} \setminus \mathcal{B}_0(R)$  and extend  $f(v)$  to a periodic function on the set  $[-T, T]^{d_v}$ . Let observe that the lower bound for  $T$  can be improved. For instance, the choice  $T = (3 + \sqrt{2})R/2$  guarantees the absence of intersection between periods where  $f$  is different from zero. However, since in practice the support of  $f$  increases with time, we can just minimize the errors due to aliasing [20] with spectral accuracy.

To further simplify the notation, let us take  $T = \pi$  and hence  $R = \lambda\pi$  with  $\lambda = 2/(3 + \sqrt{2})$  in the following. We denote by  $Q_B^R(f)$  the Boltzmann operator with cut-off. Hereafter, using one index to denote the  $d_v$ -dimensional sums, we have that the approximate function  $f_N$  can be represented as the truncated Fourier series by

$$f_N(v) = \sum_{k=-N/2}^{N/2} \hat{f}_k e^{ik \cdot v}, \quad (32)$$

$$\hat{f}_k = \frac{1}{(2\pi)^{d_v}} \int_{[-\pi, \pi]^{d_v}} f(v) e^{-ik \cdot v} dv. \quad (33)$$

We then obtain a spectral quadrature of our collision operator by projecting (8) on the space of trigonometric polynomials of degree less or equal to  $N$ , i.e.

$$\hat{Q}_k = \int_{[-\pi, \pi]^{d_v}} Q_B^R(f_N) e^{-ik \cdot v} dv, \quad k = -N/2, \dots, N/2. \quad (34)$$

Finally, by substituting expression (32) in (34) one gets after some computations

$$\hat{Q}_k = \sum_{\substack{l, m=-N/2 \\ l+m=k}}^{N/2} \hat{f}_l \hat{f}_m \hat{\beta}(l, m), \quad k = -N, \dots, N, \quad (35)$$

where  $\hat{\beta}(l, m) = \hat{B}(l, m) - \hat{B}(m, m)$  are given by

$$\hat{B}(l, m) = \int_{\mathcal{B}_0(2\lambda\pi)} \int_{\mathbb{S}^{d_v-1}} |q| \sigma(|q|, \cos \theta) e^{-i(l \cdot q^+ + m \cdot q^-)} d\omega dq. \quad (36)$$

with

$$q^+ = \frac{1}{2}(q + |q|\omega), \quad q^- = \frac{1}{2}(q - |q|\omega). \quad (37)$$

Let us notice that the naive evaluation of (35) requires  $O(n^2)$  operations, where  $n = N^3$ . This causes the spectral method to be computationally very expensive, especially in dimension three. In order to reduce the number of operations needed to evaluate the collision integral, the main idea is to use another representation of (8), the so-called Carleman representation [47] which is obtained by using the following identity

$$\frac{1}{2} \int_{\mathbb{S}^{d_v-1}} F(|u|\sigma - u) d\sigma = \frac{1}{|u|^{d-2}} \int_{\mathbb{R}^{d_v}} \delta(2x \cdot u + |x|^2) F(x) dx.$$

This gives in our context for the Boltzmann integral

$$Q_B(f) = \int_{\mathbb{R}^{d_v}} \int_{\mathbb{R}^{d_v}} \tilde{B}(x, y) \delta(x \cdot y) [f(v+y) f(v+x) - f(v+x+y) f(v)] dx dy, \quad (38)$$

with

$$\tilde{B}(|x|, |y|) = 2^{d_v-1} \sigma \left( \sqrt{|x|^2 + |y|^2}, \frac{|x|}{\sqrt{|x|^2 + |y|^2}} \right) (|x|^2 + |y|^2)^{-\frac{d_v-2}{2}}. \quad (39)$$

This transformation yields the following new spectral quadrature formula

$$\hat{Q}_k = \sum_{\substack{l, m = -N/2 \\ l+m=k}}^{N/2} \hat{\beta}_F(l, m) \hat{f}_l \hat{f}_m, \quad k = -N, \dots, N \quad (40)$$

where  $\hat{\beta}_F(l, m) = \hat{B}_F(l, m) - \hat{B}_F(m, m)$  are now given by

$$\hat{B}_F(l, m) = \int_{\mathcal{B}_0(R)} \int_{\mathcal{B}_0(R)} \tilde{B}(x, y) \delta(x \cdot y) e^{i(l \cdot x + m \cdot y)} dx dy. \quad (41)$$

Now, in order to reduce the number of operation needed to evaluate (40), we look for a convolution structure. The aim is to approximate each  $\hat{\beta}_F(l, m)$  by a sum

$$\hat{\beta}_F(l, m) \simeq \sum_{p=1}^A \alpha_p(l) \alpha'_p(m),$$

where  $A$  represents the number of finite possible directions of collisions. This finally gives a sum of  $A$  discrete convolutions and, consequently, the algorithm can be computed in  $O(A N \log_2 N)$  operations by means of standard FFT technique [20].

In order to get this convolution form, we make the decoupling assumption

$$\tilde{B}(x, y) = a(|x|) b(|y|). \quad (42)$$

This assumption is satisfied if  $\tilde{B}$  is constant. This is the case of Maxwellian molecules in dimension two, and hard spheres in dimension three, the two cases treated in this paper. Indeed, using kernel (10) in (39), one has

$$\tilde{B}(x, y) = 2^{d_v-1} C_\alpha (|x|^2 + |y|^2)^{-\frac{d_v-\alpha-2}{2}},$$

so that  $\tilde{B}$  is constant if  $d_v = 2$ ,  $\alpha = 0$  and  $d_v = 3$ ,  $\alpha = 1$ .

We start by dealing with dimension 2 and  $\tilde{B} = 1$ , i.e. Maxwellian molecules. Here we write  $x$  and  $y$  in spherical coordinates  $x = \rho e$  and  $y = \rho' e'$  to get

$$\hat{B}_F(l, m) = \frac{1}{4} \int_{\mathbb{S}^1} \int_{\mathbb{S}^1} \delta(e \cdot e') \left[ \int_{-R}^R e^{i\rho(l \cdot e)} d\rho \right] \left[ \int_{-R}^R e^{i\rho'(m \cdot e')} d\rho' \right] de de'.$$

Then, denoting  $\phi_R^2(s) = \int_{-R}^R e^{i\rho s} d\rho$ , for  $s \in \mathbb{R}$ , we have the explicit formula

$$\phi_R^2(s) = 2R \text{Sinc}(Rs),$$

where  $\text{Sinc}(x) = \frac{\sin(x)}{x}$ . This explicit formula is further plugged in the expression of  $\hat{B}_F(l, m)$  and using its parity property, this yields

$$\hat{B}_F(l, m) = \int_0^\pi \phi_R^2(l \cdot e_\theta) \phi_R^2(m \cdot e_{\theta+\pi/2}) d\theta.$$

Finally, a regular discretization of  $A$  equally spaced points, which is spectrally accurate because of the periodicity of the function, gives

$$\hat{B}_F(l, m) = \frac{\pi}{M} \sum_{p=1}^A \alpha_p(l) \alpha'_p(m), \quad (43)$$

with

$$\alpha_p(l) = \phi_R^2(l \cdot e_{\theta_p}), \quad \alpha'_p(m) = \phi_R^2(m \cdot e_{\theta_p + \pi/2}) \quad (44)$$

where  $\theta_p = \pi p/A$ .

Now let us deal with dimension  $d = 3$  with  $\tilde{B}$  satisfying the decoupling assumption (42). First we switch to the spherical coordinates for  $\hat{B}_F(l, m)$ :

$$\hat{B}_F(l, m) = \frac{1}{4} \int_{\mathbb{S}^2 \times \mathbb{S}^2} \delta(e \cdot e') \left[ \int_{-R}^R \rho a(\rho) e^{i\rho(l \cdot e)} d\rho \right] \left[ \int_{-R}^R \rho' b(\rho') e^{i\rho'(m \cdot e')} d\rho' \right] dede'.$$

Then, integrating first  $e'$  on the intersection of the unit sphere with the plane  $e^\perp$  gives

$$\hat{B}_F(l, m) = \frac{1}{4} \int_{e \in \mathbb{S}^2} \phi_{R,a}^3(l \cdot e) \left[ \int_{e' \in \mathbb{S}^2 \cap e^\perp} \phi_{R,b}^3(m \cdot e') de' \right] de$$

where

$$\phi_{R,a}^3(s) = \int_{-R}^R \rho a(\rho) e^{i\rho s} d\rho.$$

This leads to the following decoupling formula

$$\hat{B}_F(l, m) = \int_{e \in \mathbb{S}_+^2} \phi_{R,a}^3(l \cdot e) \psi_{R,b}^3(\Pi_{e^\perp}(m)) de$$

where  $\mathbb{S}_+^2$  denotes the half sphere and

$$\psi_{R,b}^3(\Pi_{e^\perp}(m)) = \int_0^\pi \sin \theta \phi_{R,b}(|\Pi_{e^\perp}(m)| \cos \theta) d\theta.$$

Now, in the particular case where  $\tilde{B} = 1$ , i.e. the hard sphere model, we can explicitly compute the functions  $\phi_R^3$ . These are

$$\phi_R^3(s) = R^2 [2 \text{Sinc}(Rs) - \text{Sinc}^2(Rs/2)], \quad \psi_R^3(s) = 2R^2 \text{Sinc}^2(Rs/2).$$

Taking a spherical parametrization  $(\theta, \varphi)$  of  $e \in \mathbb{S}_+^2$  and uniform grids of respective size  $A_1$  and  $A_2$  for  $\theta$  and  $\varphi$  (again spectrally accurate because of the periodicity of the function) leads to the following quadrature formula for  $\hat{B}_F(l, m)$

$$\hat{B}_F(l, m) = \frac{\pi^2}{A_1 A_2} \sum_{p,q=0}^{A_1, A_2} \alpha_{p,q}(l) \alpha'_{p,q}(m)$$

where

$$\begin{aligned} \alpha_{p,q}(l) &= \phi_{R,a}^3(l \cdot e_{(\theta_p, \varphi_q)}), & \alpha'_{p,q}(m) &= \psi_{R,b}^3\left(\Pi_{e_{(\theta_p, \varphi_q)}^\perp}(m)\right), \\ \phi_{R,a}^3(s) &= \int_{-R}^R \rho a(\rho) e^{i\rho s} d\rho, & \psi_{R,b}^3(s) &= \int_0^\pi \sin \theta \phi_{R,b}^3(s \cos \theta) d\theta, \end{aligned}$$

and for all  $p$  and  $q$

$$(\theta_p, \varphi_q) = \left( \frac{p\pi}{A_1}, \frac{q\pi}{A_2} \right).$$

#### 4. Numerical implementation

We discuss in this part some aspects relative to the numerical implementation of the method.

#### 4.1. Algorithm for the transport part

The method described in the previous sections can be resumed into two main actions: transport and interaction. In our implementation we have adopted a particle like interpretation of the FKS scheme which helps in reducing the computational effort and the memory requirement. In this interpretation, each point of the quadrature of the phase space is represented by a particle with a given mass, velocity and position. The transport phase causes the particle to move in the physical space, while the interaction phase causes the mass of each particle to change. In this setting, the distribution function  $f$  can be expressed as

$$f(x, v, t) = \sum_{i=1}^{N_p} \mathbf{m}_i(t) \delta(x - x_i(t)) \delta(v - v_i(t)), \quad v_i(t) = v_k, \quad (45)$$

where  $x_i(t)$  represents the particle position,  $v_i(t)$  the particle velocity,  $\mathbf{m}_i$  the particle mass. Moreover, the particle velocity corresponds to the quadrature point chosen to discretize the velocity space. The effect of the transport is a simple shift of particles to their new positions according to

$$x_i(t + \Delta t) = x_i(t) + v_i(t) \Delta t. \quad (46)$$

The collision step acts only locally and changes the velocity distribution. The particle interpretation of this part of the scheme consists in changing the mass of each particle through the formula (22) which reads

$$\mathbf{m}_i(t + \Delta t) = \mathbf{m}_i(t) + \Delta t Q(v_i), \quad (47)$$

where  $Q(v_i)$  corresponds to the approximation of the collision integral in the center of the cell evaluated at location  $v_i$  by means of the fast spectral method described before. Let observe that thanks to the uniform grid in velocity and physical space, the number of particles is the same in each cell and remains constant in time. This permits to consider the motion of only a fixed subset of particles which belongs to only one chosen cell. The relative motion of all other particles in their cells being the same [21]. This gives great computational advantage since the number of particle to track in time is greatly reduced. The only information which should be tracked remains the mass of each individual particle which is different for every velocity  $v_i$  and position  $x_i$ . This reinterpretation of the scheme permits to strongly reduce the computational cost related to the transport part as shown in the numerical test section.

#### 4.2. Profiling and parallelization strategy

The Fast Kinetic scheme for the BGK equation was shown to be very efficient and extremely parallelizable (with a shown acceleration very close to the ideal scaling) in [24] on mild parallel architecture, namely using OpenMP on maximum 64 threads and on two Graphical Processing Units (GPU) using CUDA framework. These light parallel infrastructures are unfortunately not sufficient when Boltzmann operator is to be simulated. To this aim, let us briefly discuss the profiling of the collision algorithm. As we have already seen in the description of the method in the previous section, the resolution of Boltzmann operator requires several passages from the physical space to the Fourier space, which implies several calls to (inverse) Fourier transforms within nested loops which are due to the convolutive form of the integral. This interaction step covers about 98% of the total cost of a 2D×2D simulation on a serial machine, see table 1. The situation is alike in the case of 3D×3D simulations, and, thus, results are not reported. From table 1 we deduce that the collision step is the part of the code which demands to be dealt with great attention if any gain in performance is expected. More precisely, if we give a closer look to the computational cost related to the computation of the collision term  $Q(f)$  in one fixed spacial cell, the major cost (beyond 90%) is spent by the Fourier transform routines. In our implementation, the collision is performed without any communication with the local neighborhood. In other words, once the  $N$  values of the distribution functions  $f_{j,k}$ ,  $k = 1, \dots, N$  are stored then  $Q(f)$  is computed locally. Because  $N$  is not too large, typically the number of points chosen to discretize the velocity space goes from  $N = 16^3$  to  $N = 64^3$ , then the memory requirement is not extremely demanding. Finally, as seen in the previous Section the computation of  $Q(f)$  is based on a loop over the number of angles  $\theta_p$  with  $p = 1, \dots, A$  in dimension two, and, a double loop over the angles  $\theta_p$  and

Main routines	Cost CPU %	Sub-routines	Sub-cost CPU %
Transport	$\simeq 2$ %	—	$\simeq 2$ %
Update moments	$\simeq 0$ %	—	$\simeq 0$ %
Collision	$\geq 98$ %	Compute $\mathcal{F}(f)$	$\simeq 5$ %
		$\theta_p$ -Loop: compute coeff.	$\simeq 5$ %
		$\theta_p$ -Loop: compute $\mathcal{F}^{-1}$	$\simeq 87$ %
		Update $Q$	$\simeq 3$ %
Total	100 %		100%

Table 1: Schematic table of the average cost for each routine of the code on a cylindrical Sod like problem in  $2D \times 2D$ . Simulation is run on a serial machine. The collision part demands almost all the CPU resources and, within this step, the calls to the (inverse) Fourier transform routine covers more than 85% of the total cost of one single step.

$\phi_q$  in dimension three with  $p = 1, \dots, A_1$  and  $q = 1, \dots, A_2$  with a typical number of chosen angles of eight. After a first Fourier transform of the distribution  $f$ , for each iterate of these loops, the coefficients of the quadrature formula are computed and then three calls to the inverse fast Fourier transform are done. The results of the FFT are gathered into temporary complex variables later arranged to form the solution of  $Q(f)$ . The costs related to the four main routines needed for the evaluation of the operator namely: Fourier transform, coefficients computation, inverse Fourier transform and sum of the obtained values to form  $Q(f)$ , are detailed in table 1.

We discuss now some aspects related to the parallelization strategy. The framework described is extremely well suited for parallelization. The strategy adopted is to divide the space domain in several subdomains because the interaction step, the most expensive one, does not demand any communication between spatial cells. Each cell can be therefore treated independently from the others. Then, the idea employed has been to use the OpenMP on shared memory systems and make each computational core responsible for a subset of a space mesh. This approach requires very little modification of a sequential code and gives strong scaling close to ideal as shown in [24] for the BGK case. A different possibility is to compute the collision kernel, and all the related Fast Fourier Transforms on a GPU. Even if, this involves substantial amount of slow communications between CPU and GPU, the computational complexity of this part is so large that the communication time does not play a fundamental role. This approach requires more programming effort, compared to the OpenMP parallelization strategy as several routines have to be completely rewritten. In particular, the code has to be adapted to the Single Instruction, Multiple Threads (SMT) parallel programming mode imposed by the GPU architecture. That is to say, parts of the code that can be run in parallel on large number of threads have to be identified and rewritten as so called CUDA kernels - functions that are executed multiple times in parallel by different CUDA threads. Moreover, the GPU-CPU memory data transfer has to be taken into account. The payoff is however higher and, at least for computations which does not require too large memory storage, this is probably the best choice.

Unfortunately, when the number of points needed to perform a simulation grows in the case the Boltzmann operator is used, the above implementation strategies on shared memory systems are not sufficient any more. This is true particularly in the full three dimensional case. Luckily, the method described in this work can be easily adapted for its application in distributed memory systems. The idea adopted in this case is to distribute spatial degrees of freedom over computational nodes, keeping on every node a complete set of velocity space points. The computational domain is then decomposed into slices along the  $Z$  direction (see Fig.2). The slices are successively distributed over different MPI processes. Every node will therefore compute the collision term for a part of computational domain. This can be done on each node using the OpenMP or GPU parallelization strategy, depending on the architecture at disposal. As the update of the primitive variables requires an exchange with neighboring spatial cells, some of the particle masses, which contain all the information related to the distribution function  $f$  in our implementation, need to be broadcast to other computational nodes. This data exchange, which is typically a bottleneck in the MPI computations, and, usually demands large efforts in order to be minimized the internodal communications,

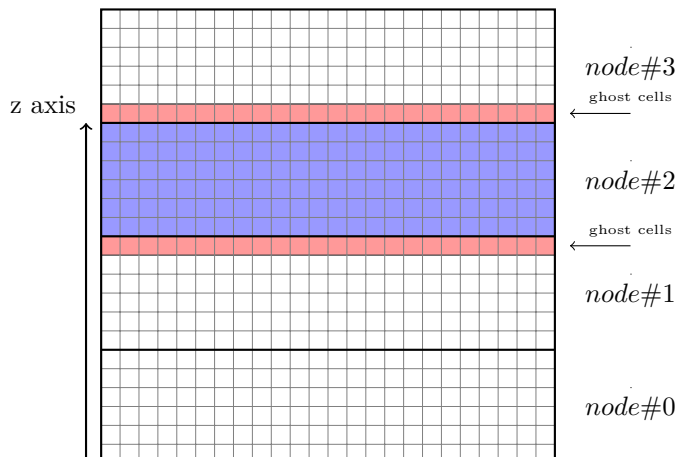


Figure 2: Domain decomposition for the MPI parallelization. The domain slices are distributed over computational nodes. After the collision step the new information brought by the particles on the cells located on the subdomain boundary are communicated to the ghost cells of the neighboring nodes.

in the case of the Boltzmann equation, takes only a small part of the total runtime, even if, in our first MPI implementation, the domain decomposition is far from being optimal. We postpone to a future work the discussion related to a more efficient MPI parallelization strategy and all details related to this type of implementation.

## 5. Numerics

In this section, we validate the proposed SFKS (Spectral-FKS) method which couples the fast semi-Lagrangian and the fast spectral scheme. The testing methodology is divided into four parts.

**Part 1.** Sanity checks. We only consider the space homogeneous Boltzmann equation in two and three dimensions in velocity space. Two problems from [4] for which the analytical solution is known are simulated. The results show that our implementation of the fast spectral discretization of the Boltzmann operator is correct.

**Part 2.** Using the SFKS method in one dimension of physical space and two or three dimensions in velocity space, we show on Riemann like problems the differences between the BGK and the Boltzmann models. The reported results justify the use of the more complex and costly collisional Boltzmann operator especially in situations far from the thermodynamical equilibrium. We also report the details of the computational costs as well as the numerical convergence of the method for an increasing number of points in the physical or velocity space.

**Part 3.** Using the SFKS method in the two dimensional case in space and velocity, we show that using the Boltzmann or the BGK operator leads to notably different results. We consider a regular vortex like problem and a reentry like problem with evolving angle of attack. Performances and profiling study of the SFKS approach supplemented with BGK or Boltzmann collisional operator are provided for refined grids in space and time. Details about the computational costs of the different part of the scheme, scalability with respect to the number of cells are also furnished in order to characterize the behavior of the method as precisely as possible.

**Part 4.** We test the SFKS method in the full three dimensional case in space and velocity for an unsteady test problem. The interaction of a flying object with the surrounding ambient gas is simulated. We compare the results obtained with the Boltzmann collisional operator with those given by a simpler

# of points	$L_1$ error	$L_2$ error
$8^2$	$6.3 \times 10^{-3}$	$3.8 \times 10^{-3}$
$16^2$	$5.7 \times 10^{-4}$	$2.9 \times 10^{-4}$
$32^2$	$6.5 \times 10^{-5}$	$7.2 \times 10^{-5}$

Table 2: Test 1.1. Sanity check —  $L_1$  and  $L_2$  relative errors for the fast spectral method for different number of points at  $t_{\text{final}} = 10$ . Maxwellian molecules in dimension two.

BGK model. Moreover, we measure the cost of such a 3D×3D simulation in terms of CPU, cost per degree of freedom and scalability with respect to the number of cells. Some code profiling is also provided to measure the cost of the main components of the scheme.

Finally, the ability of the whole numerical code to run on several parallel environments is tested: CPU (OpenMP, MPI) and GPU (CUDA) types of parallelism frameworks are considered.

### 5.1. Part 1. Numerical results for the space homogeneous case

In this part, we validate our spectral discretization implementation of the Boltzmann collisional operator. To this aim, we re-employ the test cases considered in [4] and [18] for the homogeneous two dimensional and three dimensional Boltzmann equation. For both situations, exact solutions are available and briefly recalled in the following.

#### 5.1.1. Test 1.1. Convergence to equilibrium for the Maxwell molecules in dimension two.

We consider two dimensional in velocity Maxwellian molecules and the following space homogeneous initial condition

$$f(v, t = 0) = \frac{v^2}{\pi} \exp(-v^2). \quad (48)$$

The analytical solution (the so-called BKW one) is given for all times  $t$  by [48, 49]:

$$f(v, t) = \frac{1}{2\pi S^2} \exp(-v^2/2S) \left[ 2S - 1 + \frac{1-S}{2S} v^2 \right], \quad \forall t > 0, \quad (49)$$

with  $S = 1 - \exp(-t/8)/2$ . The final time is set to  $t_{\text{final}} = 10$  and the time step is equal to  $\Delta t = 0.02$ . The test is performed for  $N = 8^2, 16^2, 32^2$  points and for eight discrete angles  $\theta_p$ . In order to keep the aliasing error smaller than the spectral error, the velocity domain is set to  $[-4, 4]^2$  for  $N = 8^2$ ,  $[-6, 6]^2$  for  $N = 16^2$  and to  $[-9, 9]^2$  for  $N = 32^2$  points. This choice helps fighting back the aliasing error behavior since it increases with the number of points. Therefore, to have comparable aliasing errors for the three meshes, we enlarge the size of the velocity space when the number of points increases. We report the discrete  $L_1$  and  $L_2$  norms of the error for the distribution function  $f$  in Table 2. Moreover, in Figure 3, we report the distribution function  $f(v_x, v_y = 0, t)$  at different times ( $t = 1, 2$  and  $10$ ) when  $N = 32$ . The solution is plotted versus the exact solution. The results clearly show the convergence of the method towards the exact solution. Last in Figure 3 right panel, we present the time evolution of the  $L_1$  error for the three configurations. Consistently the errors decrease by about one order of magnitude. The last recorded results, at  $t_{\text{final}} = 10$  corresponds to the Figures reported in table 2.

#### 5.1.2. Test 1.2. Space homogeneous comparison between the BGK and the Boltzmann model. Maxwellian molecules.

In Figure 4, we compare the convergence to equilibrium for the BGK and Boltzmann models on the test case described in the previous paragraph using  $N = 32^2$  points on a domain  $[-9, 9]^2$ . The  $L_1$  norm of the difference between the two distribution functions  $f_{\text{BGK}}(v, t)$  and  $f_{\text{Boltz}}(v, t)$  as a function of time is shown. As expected the differences are increasing at early stages of the relaxation towards the equilibrium to reach a maximum difference around time  $t \simeq 1.66$ . Then, the two models slowly converge towards the



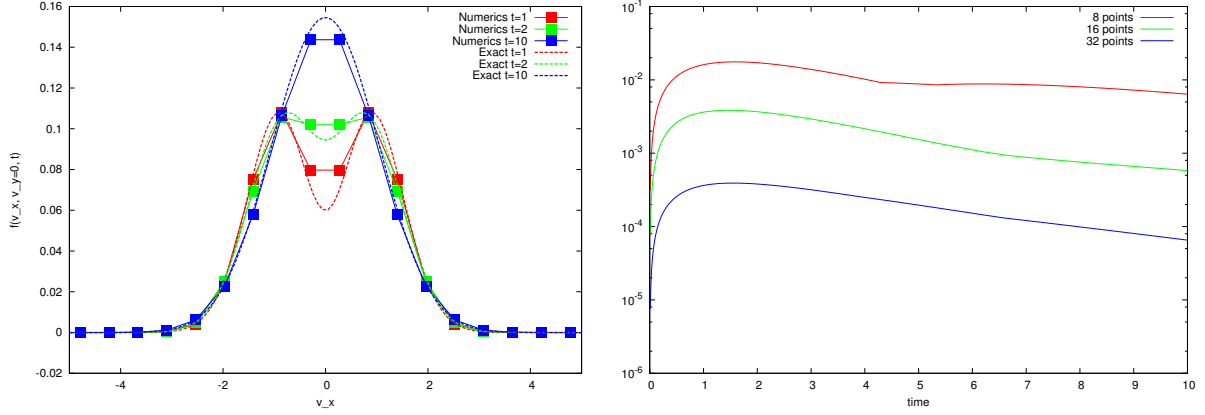


Figure 3: Test 1.1. Sanity check — Left:  $f(v_x, v_y = 0, t)$  (symbols and straight line) as a function of velocity  $v_x$  and time for  $t = 1, 2$  and  $t_{\text{final}} = 10$  versus the exact solution (dashed line). Right:  $L_1$  error as a function of time for 8, 16 and 32 points (or modes) in each direction. Maxwellian molecules in dimension two.

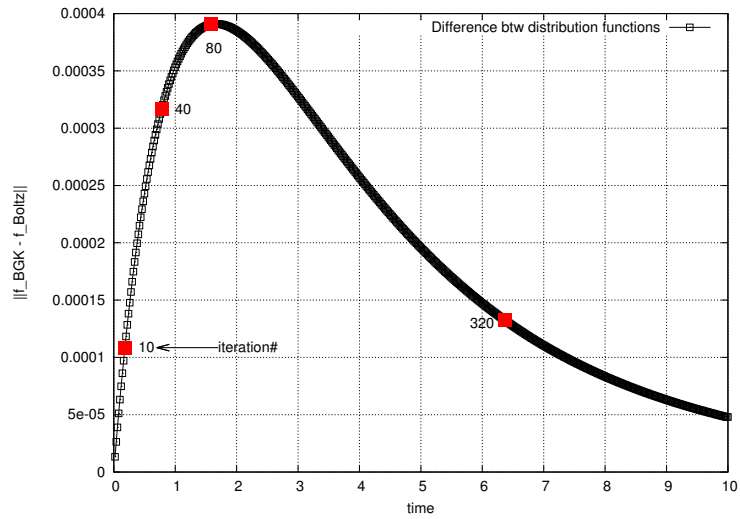


Figure 4: Test 1.2. Differences between the distribution functions obtained with the BGK and the Boltzmann models. Maxwellian molecules in two dimensions. The distribution function values for the red marked iterations are depicted in Figure 5.

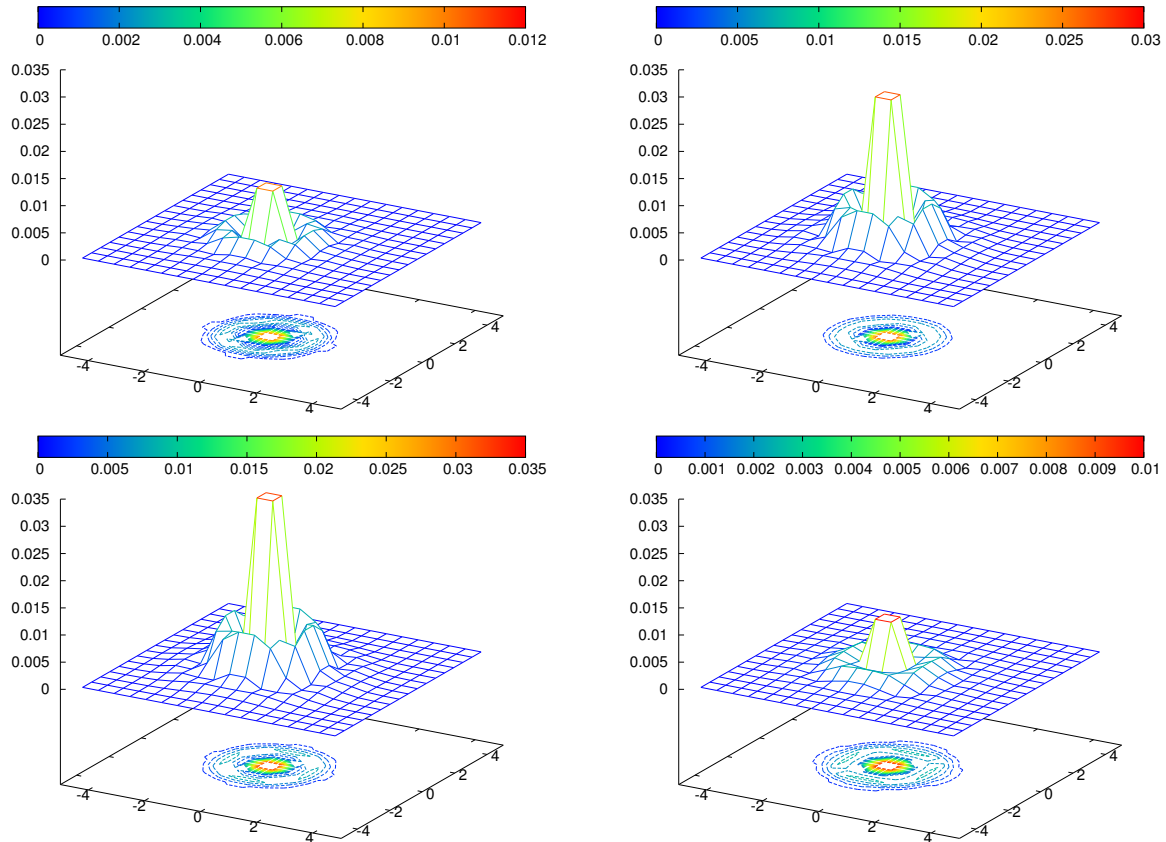


Figure 5: Test 1.2. Differences in the distribution functions obtained by using the BGK and the Boltzmann models at  $t = 0.2$ ,  $t = 1$ ,  $t = 2$  and  $t = 6.4$  from top-left to bottom-right. Maxwellian molecules in two dimensions.

same equilibrium solution. The red marked iterations in Figure 4 are further depicted in Figure 5, where we show the details of the difference in time between the two distribution functions. The same azimuthal scale is employed to ease the comparison.

### 5.1.3. Test 1.3. Convergence to equilibrium in dimension three. Hard sphere molecules.

The following initial condition is considered [18]

$$f(v, t = 0) = \frac{1}{2(2\pi\sigma^2)^{3/2}} \left[ \exp\left(-\frac{|v - v_1|^2}{2\sigma^2}\right) + \exp\left(-\frac{|v + v_1|^2}{2\sigma^2}\right) \right], \quad (50)$$

where  $\sigma^2 = 0.2$  and  $v_1$  is  $v_1 = (v_x, v_y, v_z) = (-1, -1, -0.25)$ . The final time is set to  $t_{\text{final}} = 2$ , the time step is  $\Delta t = 0.05$ , while the velocity domain is  $[-7; 7]^3$  discretized with  $N = 32^3$  points. In Figure 6, we report the initial data while in Figure 7, we report the relaxation to the equilibrium for the hard sphere model on a fixed plane passing through the points of coordinates  $\pm(-1, -1, -0.25)$  and of normal  $(-0.1, -0.1, 1)$ <sup>1</sup>. In Figure 8, we report as before the comparison between the Boltzmann and the BGK model measuring the  $L_1$  difference between the two distribution functions in time. Finally, in Figure 9 are presented the details of such differences between the distribution functions for the red marked iterations of figure 8 at  $v_z = 0$ . As for the Maxwellian molecules in two dimensions, at the beginning of the simulation we observe

<sup>1</sup>This plane passes through the center of the two spheres defined by (50)

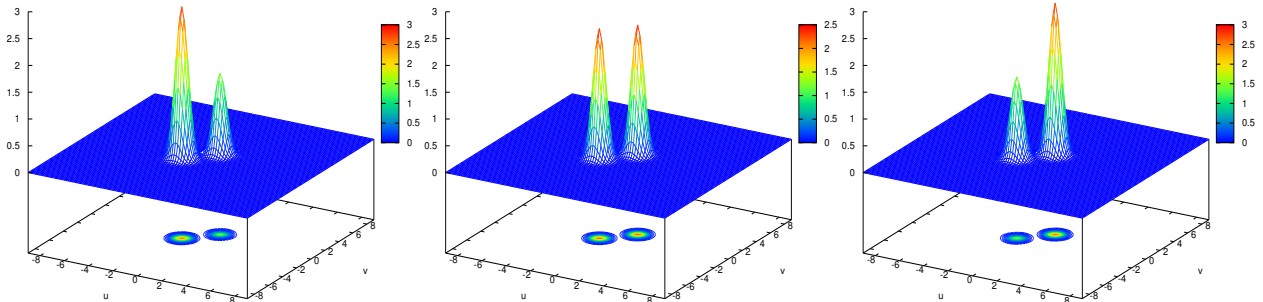


Figure 6: Test 1.3. Sanity check — Initial distribution function for  $f(v_x, v_y, 0)$  middle,  $f(v_x, v_y, 0.25)$  left and  $f(v_x, v_y, -0.25)$  right. Hard sphere molecules.

the larger deviations between the two models while at the end they both converge to the same limit solution as expected.

The results of these homogeneous tests (in two and in three dimensions in velocity) are meant to verify and validate our implementation of the spectral discretization of the Boltzmann collisional operator. Moreover, the differences observed between the two models justify the necessity of the Boltzmann operator for space non homogeneous situations. We can now investigate the coupling of the Boltzmann operator with the Fast semi-Lagrangian method for several space non homogeneous cases.

### 5.2. Part 2. Numerical results for the one dimensional case in space.

In this part, we focus on solving the one dimensional in space Boltzmann and BGK equations considering a two or three dimensional dimensional velocity space setting. The purposes are twofold. First, we numerically demonstrate that the FKS and the spectral accurate Boltzmann kernel solver may be appropriately coupled and that they provide a valid kinetic solver for Boltzmann equations. Second, we show that BGK and Boltzmann models provide different results, justifying the use of a more sophisticated model. For these tests, a classical Riemann problem with Sod like initial data is considered

$$\begin{aligned} \rho &= 1, & u &= 0, & T &= 2.5, & \text{if } x &\leq L/2, \\ \rho &= 0.125, & u &= 0, & T &= 0.25, & \text{if } x &> L/2, \end{aligned}$$

with  $\Omega = [0, 2]$ . Initial local thermodynamics equilibrium is considered for all tests:  $f(x, v, t = 0) = M[f](x, v, t = 0)$ . The velocity space is set to  $L_v = [-15, 15]$  for all cases. Dirichlet boundary conditions are set on the left/right boundaries of  $\Omega$ . The two kinetic models are solved by employing a time rescaling factor in order to put in evidence the role of the collisions in the solutions. The rescaled equations reads

$$\partial_t f + v \cdot \nabla_x f = \frac{1}{\tau} Q(f), \quad (51)$$

where  $\tau$  is the rescaled parameter (the frequency of relaxation) which plays the role of the non dimensional Knudsen number. Smaller is the relaxation frequency, faster is the relaxation of the distribution function towards the equilibrium state. However, the exact rate of convergence is dependent on the type of collision considered either BGK or Boltzmann (Maxwellian molecules or hard spheres).

#### 5.2.1. Test 2.1. Numerical convergence of the Boltzmann equation. The two dimensional in velocity Maxwellian molecules case.

The two dimensional velocity space is first considered leading to a space/velocity mesh of the form  $M \times N$  with  $N = 64^2$  and a varying number of space cells  $M$ . In Figure 10, we present the space convergence results

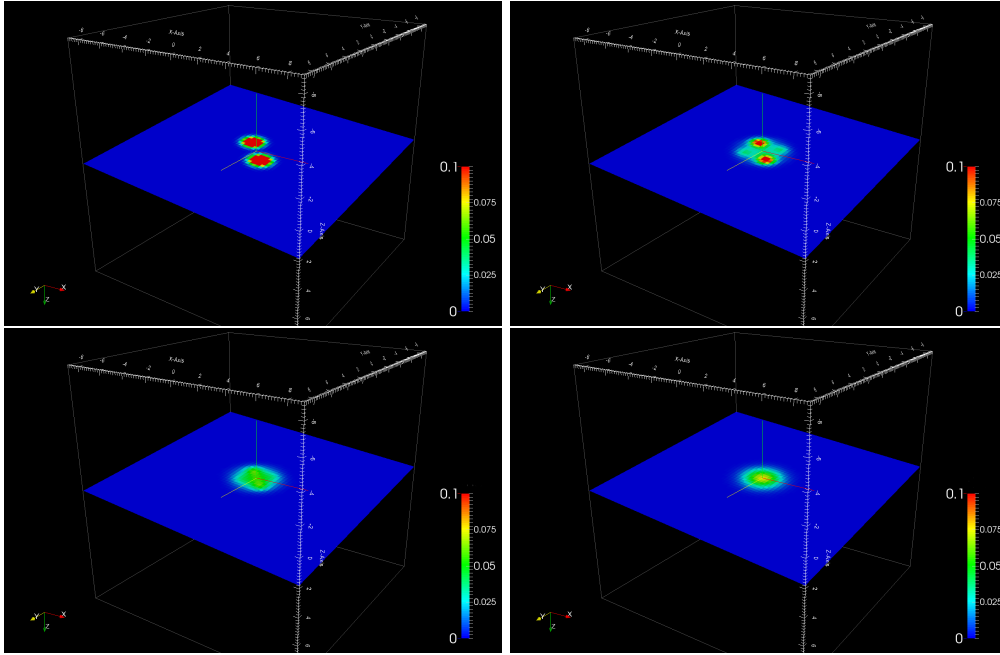


Figure 7: Test 1.3. Sanity check for the hard sphere case at time  $t = 0, 0.25, 0.5$  and  $t_{\text{final}} = 2$  for 32 points in each direction. The Figure shows the relaxation to the Maxwellian state for the distribution function  $f(v_x, v_y, v_z, t)$  on a plane which passes from the points  $\pm(-1, -1, -0.25)$  and of normal  $(-0.1, -0.1, 1)$ .

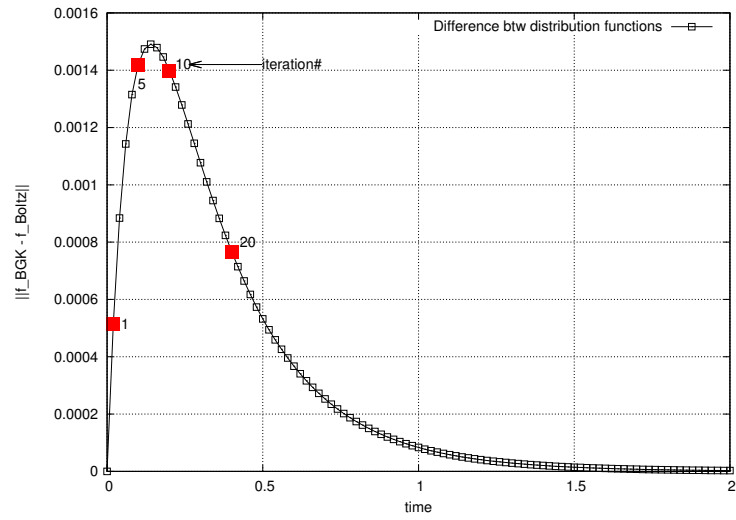


Figure 8: Test 1.3. Differences between the distribution functions obtained with the BGK and the Boltzmann models. Hard sphere molecules in three dimensions. The distribution function values for the red marked iterations are depicted in Figure 9.

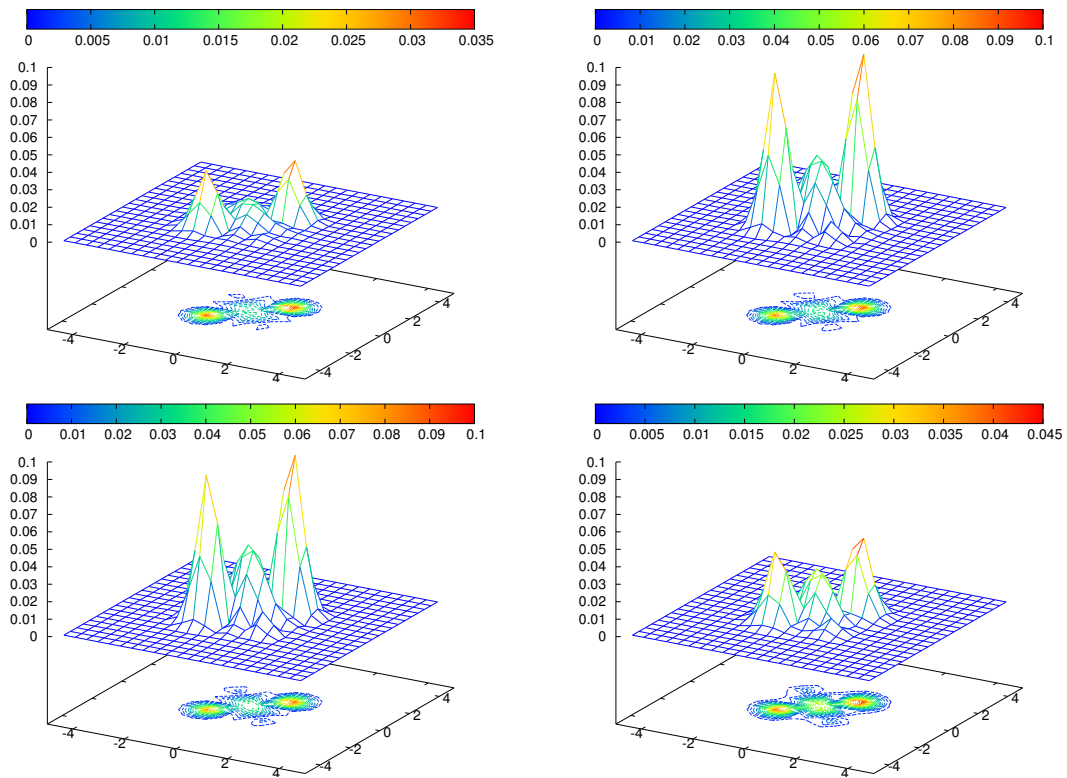


Figure 9: Test 1.3. Differences in the distribution functions obtained by using the BGK and the Boltzmann models at  $t = 0.05$ ,  $t = 0.25$ ,  $t = 0.5$  and  $t = 1$  from top-left to bottom-right for  $v_z = 0$ . Hard sphere molecules in three dimensions.

for the density, the velocity and the temperature when the Boltzmann operator is solved. Successively refined (doubled) spatial meshes are employed, from 50 to 400 up to final time  $t_{\text{final}} = 0.15$ . From these data we can observe that the simulation results seem to converge towards a given numerical solution in both cases  $\tau = 10^{-3}$  (left panels) and  $\tau = 10^{-4}$  (right panels). We can also observe that the increase in mesh resolution is profitable especially for smaller  $\tau$ . This is the same behavior observed in [21]. In fact, the scheme precision decreases as the equilibrium state is approached, being virtually exact in non collisional or almost non collisional regimes. The loss of precision observed in fluid dynamic regimes can be recovered with a similar technique as the one proposed in [22]. Here, however we do not consider this possibility. The CFL condition employed is the following

$$\Delta t \leq \min \left( \frac{\Delta x}{|v_{max}|}, \frac{\tau}{\rho} \right), \quad (52)$$

where the first term on the right hand side of the above equation comes from the will of keeping the error small enough in the splitting scheme, while the second term is due to the stability restriction in the solution of the space homogeneous problem when Maxwellian molecules are employed. In fact, in this case the loss part of the collision integral  $Q^-(f)$  can be estimated, giving  $Q^-(f) = \rho(f)f$ .

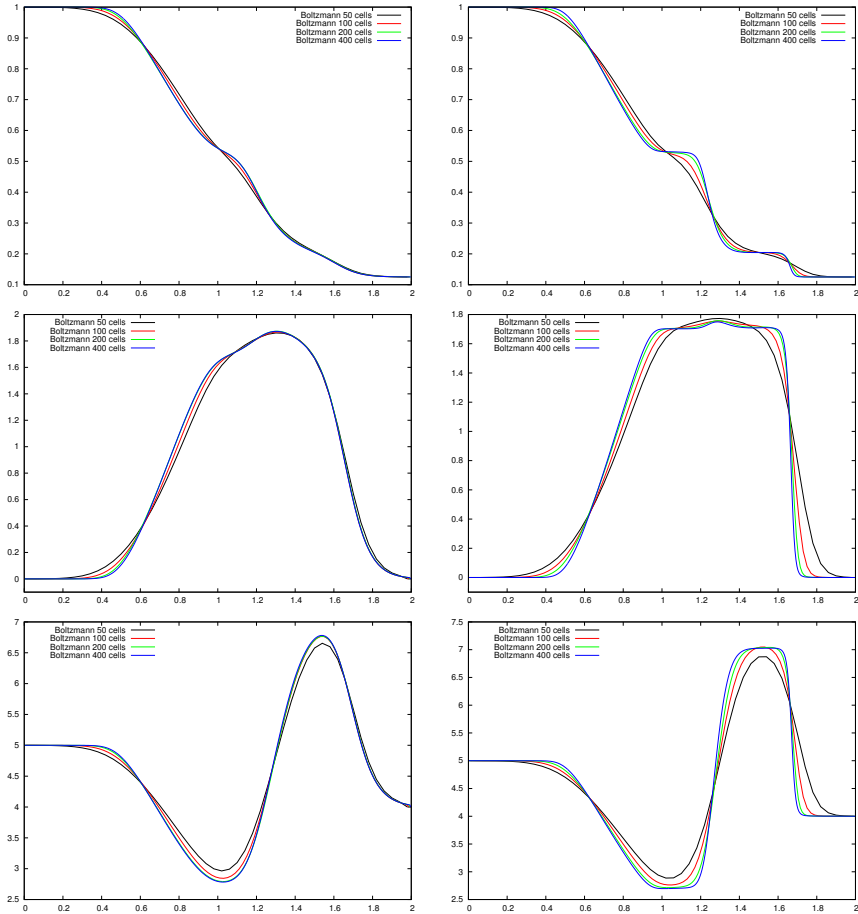


Figure 10: Test 2.1. One dimension in space and two dimension in velocity Boltzmann model with Maxwellian molecules for a Sod like test case at  $t_{\text{final}} = 0.15$ . Mesh convergence results for  $\tau = 10^{-3}$  (left) and  $\tau = 10^{-4}$  (right). Density (top), velocity (middle) and temperature (bottom) are shown for  $M = 50, 100, 200$ , and 400 cells and  $N = 64^2$  velocity cells.

*5.2.2. Test 2.2. Comparisons between the BGK model and the Boltzmann model. The two dimensional in velocity Maxwellian molecules case.*

Here, the BGK and Boltzmann models are simulated for the same Sod-like problem. In order to have fairest as possible comparisons between the two models, we choose  $\nu = \rho$  for the BGK model. This choice permits to have the same loss part for the two models since for Maxwellian molecules the loss part is close to  $\rho f$  as stated in the previous paragraph. We fix  $M = 800$  spatial cells and  $N = 64^2$  cells in velocity space. This permits to consider almost converged results. In Figure 11, we present the results when  $\tau = 10^{-3}$  (left) and  $\tau = 10^{-4}$  (right) for the density (top), the velocity (middle top), the temperature (middle bottom) and the heat flux (bottom). The very first observation is the relative large differences between the two solutions. The Boltzmann solution systematically presents more dissipated waves than BGK solution. However, the main waves are located in the same positions for the two models. The difference in term of the macroscopic quantities can be of the order of 10% – 25% of the solution. Systematically, the BGK model underestimates the values of the heat flux. In Figure 12 we report the absolute value of the difference between the two distribution functions  $f_{\text{BGK}}(x, v, t)$  and  $f_{\text{Boltz}}(x, v, t)$  at different locations  $x_j$  and at final time  $t_{\text{final}}$  as a function of the velocity variables  $v$ . The vertical scale is kept constant for all panels, only the color scale is adapted to the values reported.

*5.2.3. Test 2.3. Numerical convergence of the Boltzmann equation. The three dimensional in velocity hard sphere molecules case.*

Let us focus on the one dimension in space and three dimensions in velocity case. We consider the same Sod-like problem up to final time  $t_{\text{final}} = 0.5$  and two different collision frequencies  $\tau = 10^{-2}$  and  $\tau = 2 \cdot 10^{-3}$ . The space/velocity mesh chosen is of the form  $M \times N$  with  $N = 32^3$  uniformly spread on a velocity domain  $[-16; 16]^3$  and varying number of space cells  $M$ . In Figure 13, we present the space convergence results for the density, the velocity and the temperature for successively refined spatial meshes from 32 to 128. From these data we can observe that the simulation results seem to converge towards the same numerical solution in both cases  $\tau = 10^{-2}$  (left panels) and  $\tau = 2 \cdot 10^{-3}$  (right panels). The CFL condition employed in this case is the following

$$\Delta t \leq \min \left( \frac{\Delta x}{|v_{\text{max}}|}, \frac{\tau}{\mu} \right), \quad (53)$$

where the second term is due to the stability restriction in the solution of the space homogeneous problem when hard sphere molecules are employed. In this case, the loss part of the collision integral  $Q^-(f)$  can be only estimated, giving  $Q^-(f) = L(f)f \leq C_\alpha 4\pi(2\lambda\pi)^\alpha$  and thus  $\mu \geq C_\alpha 4\pi(2\lambda\pi)^\alpha$  in order to ensure that the gain part  $Q^+(f)$  is positive and monotone. We have also performed comparisons between the BGK and the Boltzmann models in this one dimensional setting. The results are close to the ones obtained for the two dimensional Maxwell molecules, i.e. the BGK model tends to over-relax the distribution function to the equilibrium state and for this reason we do not report them. However, since the computational costs involved in the approximation of the Boltzmann integral in the three dimensional setting are much larger than those obtained with the simpler two dimensional model, in the next paragraph we analyze the performances of the scheme proposed in both cases.

*5.2.4. Performances*

In this part, we analyze the performances of our scheme in the one dimensional in space setting. In Table 3 we monitor the CPU time for the Boltzmann and BGK operators when a successively refined spatial mesh is considered for a fixed two and three dimensional velocity mesh. The data reported are relative to the same Sod-like problem considered in the previous paragraphs. We employ  $64^2$  velocity cells in the two dimensional case and  $32^3$  in the three dimensional case. The number of spatial cells is  $M = 100, 200$  and  $400$  in the two dimensional case and  $M = 32, 64,$  and  $128$  in the three dimensional one. More precisely, we measure the CPU time  $T$  in seconds, the time per cycle as  $T_{\text{cycle}} = T/N_{\text{cycle}}$ , the time per cycle per spatial cell by  $T_{\text{cell}} = T/N_{\text{cycle}}/M$  and the time per cycle per degree of freedom (d.o.f)  $T_{\text{dof}} = T/N_{\text{cycle}}/(M \times N)$ . The OpenMP parallel version of the scheme is used on a laptop having 8 threads (HP ZBook with Intel 8 Core i7-4940MX CPU @ 3.10GHz on Ubuntu 15.10 (64 bits)). These simulations have been run in parallel

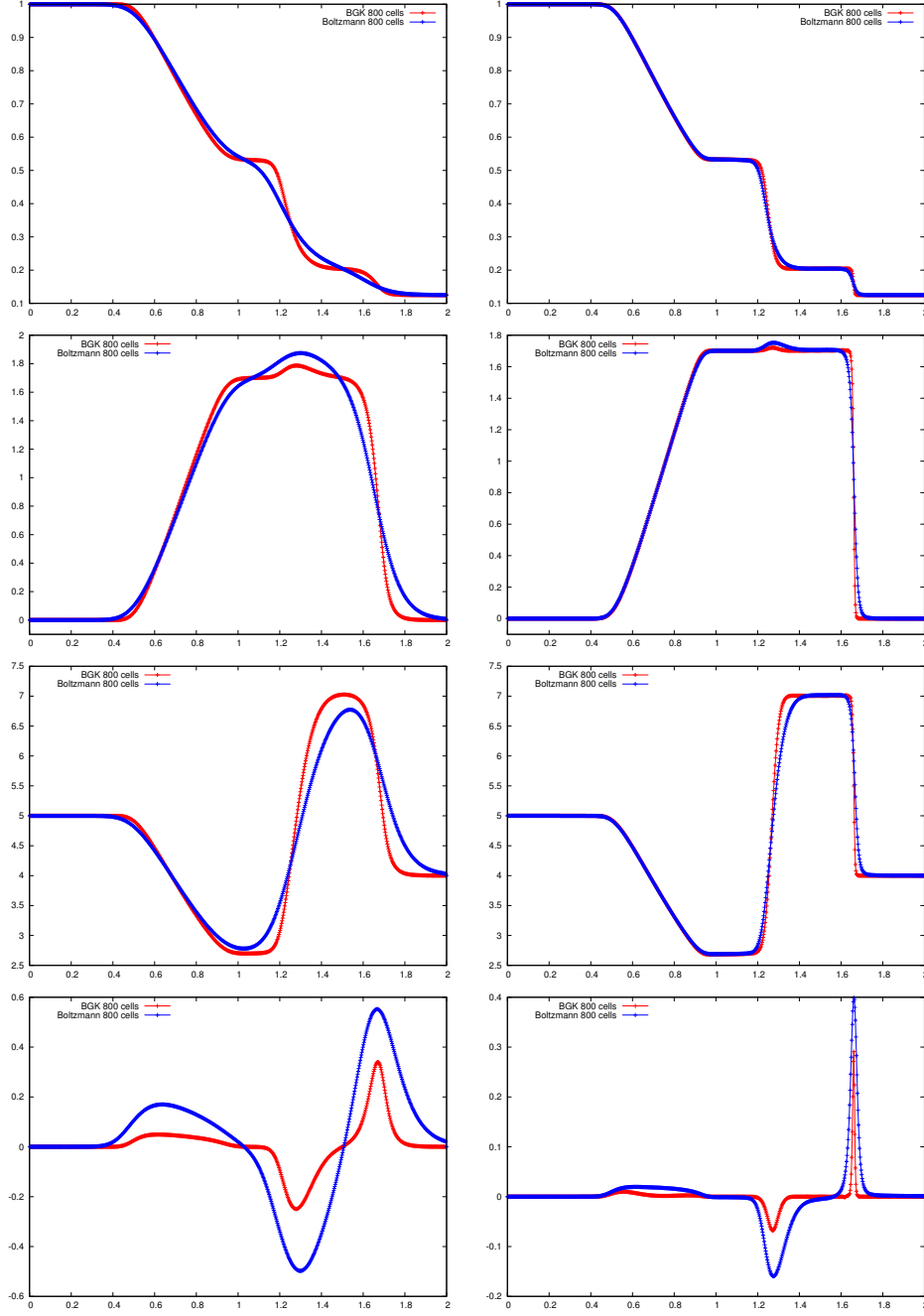


Figure 11: Test 2.2. One dimension in space and two dimension in velocity Boltzmann model with Maxwellian molecules (blue) and a BGK model (red) for a Sod like test case at  $t_{\text{final}} = 0.15$  for  $\tau = 10^{-3}$  (left) and  $\tau = 10^{-4}$  (right). Density (top), velocity (middle top), temperature (middle bottom) and heat flux (bottom) are shown for  $M = 800$  cells and  $N = 64^2$  velocity cells.



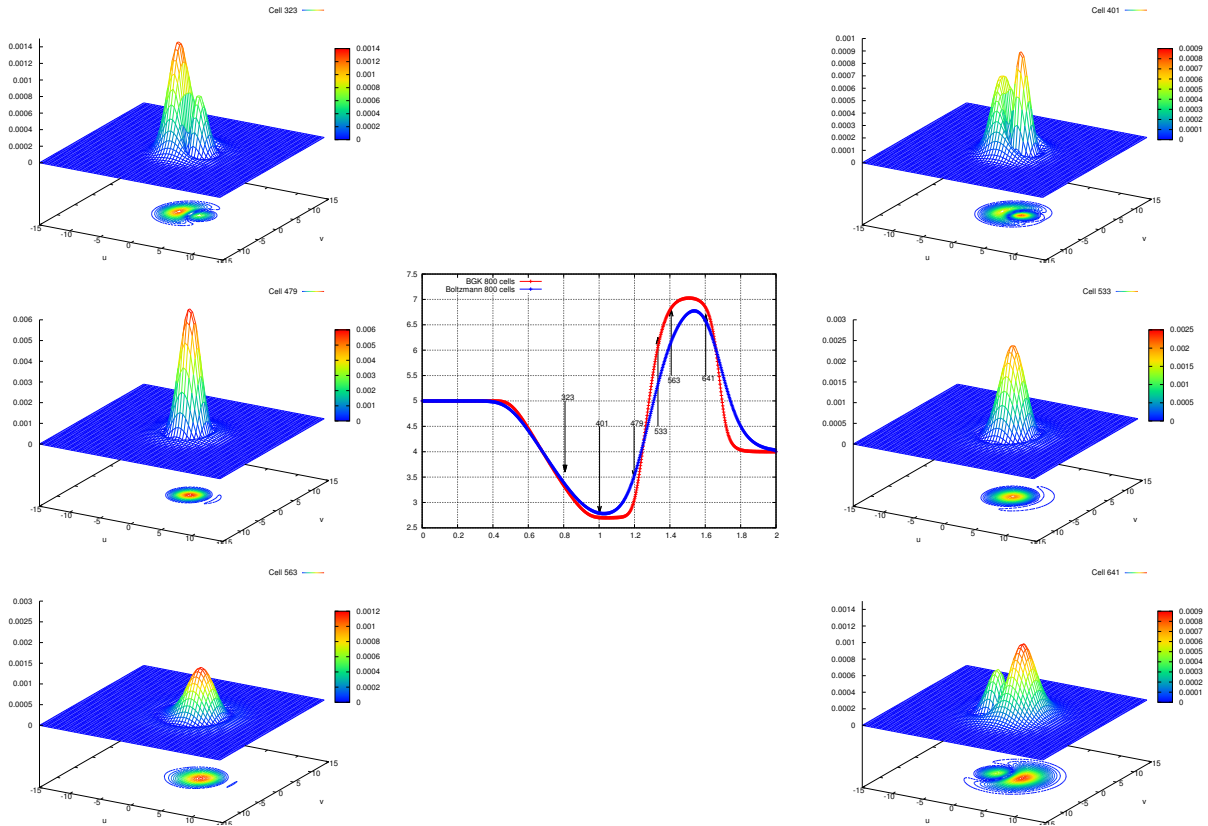


Figure 12: Test 2.2. One dimension in space and two dimension in velocity at  $t_{\text{final}} = 0.15$  for BGK and Boltzmann models for  $\tau = 10^{-3}$  with  $M = 800$  and  $N = 64^2$  velocity cells. Middle panel: temperature for the two models. The arrows indicate the regions for which the difference of the two distribution functions  $|f^{\text{BGK}}(x_i, v) - f^{\text{Boltz}}(x_i, v)|$  is reported. The vertical scale is kept constant, only the color scale is adapted to the values.

on non dedicated computer, as such the results are to be understood as rough estimates. By no mean we pretend that they cannot be improved. In the two dimensional case in velocity, the Boltzmann model results are about 15 times more expensive than the BGK model, while in three dimensions, it can reach about 250 times. Notice that, as expected, these ratio do increase when the collision frequency  $\tau$  becomes smaller, for instance when  $\tau = 2 \times 10^{-3}$  in three dimensions then the ratio in CPU time between the two models is of the order of 500. Let us notice that the number of cycles for hard sphere molecules is fixed for the reported simulations. This is due to the fact that the hard sphere model considered has a stability condition which is more restrictive than the CFL condition chosen for the transport part as opposite to the Maxwellian molecule model for the two dimensional case. For fair comparison, the number of cycles of the two models has been kept constant in the computation of the costs, even if, the space homogeneous BGK model does not present stability requirements. Next in table 4 we monitor the CPU time of the simulation when the number of cells is fixed in space to  $M = 100$ . The number of velocity cells increases from  $N = 8^2$  to  $64^2$  in two and from  $N = 8^3$  to  $64^3$  in three dimensions. The same Sod test case as in table 3 is simulated. Because the CPU times have been obtained by parallel simulations, the analysis of such table must take into account the fact that the dimension of the mesh plays a role in the performance. For instance, in two dimensions, the smaller mesh ( $100 \times 8^2$ ) usually presents a larger CPU time per degree of freedom. This is due to the fact that a decent amount of cells is needed to observe the benefit of using a parallel machine. For the Boltzmann simulations, in three dimensions, 8 velocity cells in each direction is not large enough to have a stable simulation. This is probably due to the large loss of energy caused by the spectral method

Sod like Riemann problem in 1D×2D $\tau = 10^{-3}$								
Model	Velocity		Cell #	Cycle	Time	T/cycle	T/cell	T/d.o.f
	$N$	Vel.	$M \times N$	$N_{\text{cycle}}$	T (s)	$T_{\text{cycle}}$ (s)	$T_{\text{cell}}$ (s)	$T_{\text{dof}}$ (s)
BGK	$64^2$	[-15, 15]	$100 \times 64^2$ $\approx 4.1 \times 10^5$	111	0.935	$8.42 \times 10^{-3}$	$8.42 \times 10^{-5}$	$2.06 \times 10^{-8}$
			$200 \times 64^2$ $\approx 8.2 \times 10^5$	222	2.738	$1.23 \times 10^{-2}$	$6.17 \times 10^{-5}$	$1.51 \times 10^{-8}$
			$400 \times 64^2$ $\approx 16.4 \times 10^5$	436	7.822	$1.77 \times 10^{-2}$	$4.41 \times 10^{-5}$	$1.08 \times 10^{-8}$
Boltzmann	$64^2$	[-15, 15]	$100 \times 64^2$ $\approx 4.1 \times 10^5$	111	8.174	$7.36 \times 10^{-2}$	$7.36 \times 10^{-4}$	$5.64 \times 10^{-8}$
			$200 \times 64^2$ $\approx 8.2 \times 10^5$	218	28.095	$1.27 \times 10^{-1}$	$6.33 \times 10^{-4}$	$4.42 \times 10^{-8}$
			$400 \times 64^2$ $\approx 16.4 \times 10^5$	443	113.495	$2.56 \times 10^{-1}$	$6.40 \times 10^{-4}$	$1.56 \times 10^{-7}$
Sod like Riemann problem in 1D×3D $\tau = 10^{-2}$								
Model	Velocity		Cell #	Cycle	Time	T/cycle	T/cell	T/d.o.f
	$N$	Vel.	$M \times N$	$N_{\text{cycle}}$	T (s)	$T_{\text{cycle}}$ (s)	$T_{\text{cell}}$ (s)	$T_{\text{dof}}$ (s)
BGK	32	[-16, 16]	$32 \times 32^2$ $\approx 1.05 \times 10^6$	395	12.64	$3.20 \times 10^{-2}$	$1.00 \times 10^{-3}$	$3.05 \times 10^{-8}$
			$64 \times 32^2$ $\approx 2.10 \times 10^6$	395	19.55	$4.95 \times 10^{-2}$	$7.73 \times 10^{-4}$	$2.36 \times 10^{-8}$
			$128 \times 32^2$ $\approx 4.19 \times 10^6$	395	40.75	$1.03 \times 10^{-1}$	$8.06 \times 10^{-4}$	$2.46 \times 10^{-8}$
Boltzmann	32	[-16, 16]	$32 \times 32^2$ $\approx 1.05 \times 10^6$	395	2104.17 $\sim 35\text{mn}$	5.33	$1.66 \times 10^{-1}$	$5.08 \times 10^{-6}$
			$64 \times 32^2$ $\approx 2.10 \times 10^6$	395	4799.19 $\sim 1.3\text{h}$	12.10	$1.90 \times 10^{-1}$	$5.79 \times 10^{-6}$
			$128 \times 32^2$ $\approx 4.19 \times 10^6$	395	9114.56 $\sim 2.5\text{h}$	23.10	$1.80 \times 10^{-1}$	$5.50 \times 10^{-6}$

Table 3: One dimensional in space, two dimensional in velocity Maxwellian molecules and three dimensional in velocity hard sphere molecules simulations. Comparisons between the BGK and Boltzmann models for spatial mesh variation. Monitoring of CPU time. Time per cycle is obtained by  $T_{\text{cycle}} = T/N_{\text{cycle}}$ , time per cycle per cell by  $T_{\text{cell}} = T/N_{\text{cycle}}/M$  and time per cycle per degree of freedom  $T_{\text{dof}} = T/N_{\text{cycle}}/(M \times N)$ .

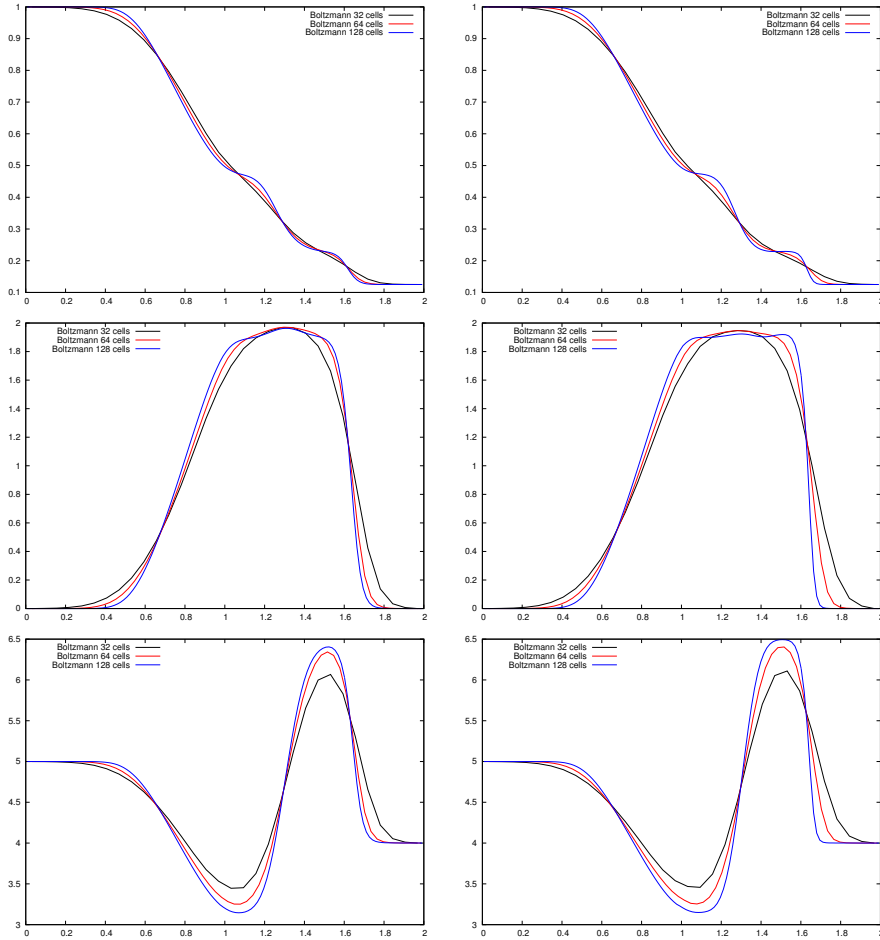


Figure 13: Test 2.3: One dimension in space and three dimension in velocity for Sod like test case at  $t_{\text{final}} = 0.5$  for the hard sphere molecules case. Mesh convergence results for  $\tau = 10^{-2}$  (left) and  $\tau = 2 \cdot 10^{-3}$  (right). Density (top), velocity (middle) and temperature (bottom) are shown for  $M = 32, 64$  and  $128$  cells and  $N = 32^3$  velocity cells.

which can be only partly cured by the  $L_2$  projection technique detailed in Section 3.2. In fact, even if such renormalization permits to keep the correct energy after the collision step, it may transform the distribution function in an unphysical manner. This operation repeated multiple times give rise to instability in the spectral scheme. However, for completeness using the first completed iterations we report in the table an approximation of the CPU time that would be needed using this small amount of points. The Hard Sphere molecule simulations, as can be seen on the Figures reported, are extremely time consuming already in the one dimensional case in space. In two dimensions, the cost of the Boltzmann model is about 3 to 8 times (from the smallest to largest mesh) more expensive than the BGK model. In three dimensions, this ratio ranges between 40 and 700 times.

### 5.3. Part 3. Numerical results for the space two dimensional case.

In this part, we focus on solving the two dimensional in space and velocity Boltzmann and BGK equations. The purposes are twofold. First, we want to show the differences which arises between the two models. Second, we want to analyze the performances of the method in the two dimensional setting by monitoring the cost of such simulations and by performing a profiling of the scheme in terms of the main routines in order to highlight the eventual bottlenecks. This permits to understand in which part of the scheme one

Sod like Riemann problem in 1D×2D $\tau = 10^{-3}$								
Model	Velocity		Cell #	Cycle	Time	T/cycle	T/cell	T/d.o.f
	$N$	Vel.	$M \times N$	$N_{\text{cycle}}$	T (s)	$T_{\text{cycle}}$ (s)	$T_{\text{cell}}$ (s)	$T_{\text{dof}}$ (s)
BGK	8 <sup>2</sup>	[-15, 15]	$100 \times 8^2$ $\simeq 6.4 \times 10^4$	105	0.126	$1.20 \times 10^{-3}$	$1.20 \times 10^{-5}$	$1.87 \times 10^{-7}$
	16 <sup>2</sup>		$100 \times 16^2$ $\simeq 2.56 \times 10^4$	106	0.265	$2.50 \times 10^{-3}$	$2.50 \times 10^{-5}$	$9.78 \times 10^{-8}$
	32 <sup>2</sup>		$100 \times 32^2$ $\simeq 1.024 \times 10^5$	109	0.291	$2.67 \times 10^{-3}$	$2.67 \times 10^{-5}$	$2.61 \times 10^{-8}$
	64 <sup>2</sup>		$100 \times 64^2$ $\simeq 4.096 \times 10^5$	111	0.935	$8.42 \times 10^{-3}$	$8.42 \times 10^{-5}$	$2.06 \times 10^{-8}$
Boltzmann	8 <sup>2</sup>	[-15, 15]	$100 \times 8^2$ $\simeq 6.4 \times 10^4$	105	0.387	$3.69 \times 10^{-3}$	$3.69 \times 10^{-5}$	$5.76 \times 10^{-7}$
	16 <sup>2</sup>		$100 \times 16^2$ $\simeq 2.56 \times 10^4$	106	0.733	$3.69 \times 10^{-3}$	$3.69 \times 10^{-5}$	$5.76 \times 10^{-7}$
	32 <sup>2</sup>		$100 \times 32^2$ $\simeq 1.024 \times 10^5$	109	2.302	$2.11 \times 10^{-2}$	$2.11 \times 10^{-4}$	$2.06 \times 10^{-7}$
	64 <sup>2</sup>		$100 \times 64^2$ $\simeq 4.096 \times 10^5$	111	7.834	$7.06 \times 10^{-2}$	$7.06 \times 10^{-4}$	$1.72 \times 10^{-7}$
Sod like Riemann problem in 1D×3D $\tau = 10^{-2}$								
Model	Velocity		Cell #	Cycle	Time	T/cycle	T/cell	T/d.o.f
	$N$	Vel.	$M \times N$	$N_{\text{cycle}}$	T (s)	$T_{\text{cycle}}$ (s)	$T_{\text{cell}}$ (s)	$T_{\text{dof}}$ (s)
BGK	8 <sup>3</sup>	[-15, 15]	$100 \times 8^3$ $\simeq 5.12 \times 10^4$	395	0.126	$1.20 \times 10^{-3}$	$1.20 \times 10^{-5}$	$1.87 \times 10^{-7}$
	16 <sup>3</sup>		$100 \times 16^3$ $\simeq 4.10 \times 10^5$	395	0.265	$2.50 \times 10^{-3}$	$2.50 \times 10^{-5}$	$9.78 \times 10^{-8}$
	32 <sup>3</sup>		$100 \times 32^3$ $\simeq 3.28 \times 10^6$	395	0.291	$2.67 \times 10^{-3}$	$2.67 \times 10^{-5}$	$2.61 \times 10^{-8}$
	64 <sup>3</sup>		$100 \times 64^3$ $\simeq 2.62 \times 10^7$	395	0.935	$8.42 \times 10^{-3}$	$8.42 \times 10^{-5}$	$2.06 \times 10^{-8}$
Boltzmann	8 <sup>3</sup>	[-15, 15]	$100 \times 8^3$ $\simeq 5.12 \times 10^4$	395	FAIL ( $\sim 25$ )	FAIL ( $\sim 6.50 \times 10^{-2}$ )	FAIL ( $\sim 6.50 \times 10^{-4}$ )	FAIL ( $\sim 1.27 \times 10^{-6}$ )
	16 <sup>3</sup>		$100 \times 16^3$ $\simeq 4.10 \times 10^5$	395	241 $\sim 4\text{mn}$	$6.11 \times 10^{-1}$	$6.11 \times 10^{-3}$	$1.49 \times 10^{-6}$
	32 <sup>3</sup>		$100 \times 32^3$ $\simeq 3.28 \times 10^6$	395	6438 $\sim 1.8\text{h}$	$1.63 \times 10^1$	$1.63 \times 10^{-1}$	$4.97 \times 10^{-6}$
	64 <sup>3</sup>		$100 \times 64^3$ $\simeq 2.62 \times 10^7$	395	79099 $\sim 22\text{h}$	$2.00 \times 10^2$	$2.00 \times 10^0$	$7.64 \times 10^{-6}$

Table 4: One dimensional in space, two dimensional in velocity Maxwellian molecules and three dimensional in velocity hard sphere molecules simulations. Simulations are performed using the OpenMP version of the scheme run on 8 threads. Comparisons between the BGK and Boltzmann models for velocity mesh variation. Monitoring of CPU time. Time per cycle is obtained by  $T_{\text{cycle}} = T/N_{\text{cycle}}$ , time per cycle per cell by  $T_{\text{cell}} = T/N_{\text{cycle}}/M$  and time per cycle per degree of freedom  $T_{\text{dof}} = T/N_{\text{cycle}}/(M \times N)$ .

should concentrate to improve the efficiency in the future. For all reported simulations Maxwellian molecules are considered for the Boltzmann model. The CFL condition employed is as for the one dimensional case and for all tests the following

$$\Delta t \leq \min \left( \frac{\Delta x}{|v_{max}|}, \frac{\tau}{\rho} \right). \quad (54)$$

### 5.3.1. Test 3.1. Two dimensional vortex in motion.

The test case consists of an isentropic vortex in motion initially introduced for the collisional regime, i.e. the compressible Euler equations, in two dimensions in [50]. This problem has an exact smooth solution expressed analytically in the fluid regime. The computational domain is  $\Omega = [0, 10]^2$ . The ambient flow is characterized by a gas with density, mean velocity and temperature respectively of  $\rho_\infty = 1.0$ ,  $u_{x,\infty} = 1.0$ ,  $u_{y,\infty} = 1.0$ ,  $T_\infty = 1.0$ . A vortex is centered at  $(x_v, y_v) = (5, 5)$  and supplemented at the initial time  $t = 0$  with conditions  $u_x(t = 0) = u_{x,\infty} + \delta u$ ,  $u_y(t = 0) = u_{y,\infty} + \delta v$ ,  $T(t = 0) = T_\infty + \delta T$  with

$$\delta u_x = -y' \frac{\beta}{2\pi} \exp \left( \frac{1-r^2}{2} \right), \quad \delta u_y = x' \frac{\beta}{2\pi} \exp \left( \frac{1-r^2}{2} \right), \quad \delta T = -\frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp(1-r^2),$$

where  $r$  is the Euclidean distance in the two dimensional space, i.e.  $r = \sqrt{x'^2 + y'^2}$ , and  $x' = x - x_v$ ,  $y' = y - y_v$ . The vortex strength depends on the value  $\beta$  fixed here to 5.0. The initial density is given by

$$\rho(t = 0) = \rho_\infty \left( \frac{T(t = 0)}{T_\infty} \right)^{\frac{1}{\gamma-1}} = \left( 1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp(1-r^2) \right)^{\frac{1}{\gamma-1}}. \quad (55)$$

Periodic boundary conditions are prescribed everywhere. At the final time chosen:  $t_{\text{final}} = 10$ , the vortex is back to its original position and, in the collisional regime, the initial and final conditions are alike. In a rarefied regime, the exact solution is not known, but at least the cylindrical symmetry of the problem must be retrieved.  $M = 100 \times 100$  uniform spatial cells are considered on domain  $[0; 10]^2$  with  $N = 32 \times 32$  uniform velocity cells on a velocity domain  $[-7.5; 7.5]^2$ . The relaxation frequency is fixed to  $\tau = 10^{-1}$ . The initial data are presented in Figure 14.

This problem is simulated using both the BGK and Boltzmann models with  $\nu = \rho$  for the BGK case. The results are presented in Figure 15 where density, temperature and the velocity are plotted. The top line presents BGK model results while the bottom line presents Boltzmann ones. The same scale is used to ease the comparison. The velocity fluctuation ( $\delta u_x, \delta u_y$ ) is represented with the same scaling as to observe that the vortex is more dissipated by the Boltzmann model than by BGK one. In Figure 16 we replot the initial density along with the final BGK and Boltzmann results using the same color scale used for showing the initial data. From the Figures we can observe that both models reproduce a vortex at the correct final location but with different rates of dissipation. The BGK model furnishes results closer to the compressible Euler solution results which means over-relaxation.

We analyze now the performances of the scheme. In order to obtain the previous results both models used 1400 time steps to reach  $t_{\text{final}}$ . This result is due to the choice of the time integrator: the same for both models. The Open MP version of the code is run on 8 cores and the total CPU times on a laptop HP ZBook Intel Core i7-4940MX CPU @ 3.10GHz $\times$ 8 operated by Ubuntu 15.10 64bits, are of the order  $\text{CPU}_{\text{BGK}} \simeq 102\text{s}$ , and  $\text{CPU}_{\text{Boltz}} \simeq 2008\text{s}$ . We split the code in several distinct and conceptually important parts which are denoted as `Transport`, `ToConservative`, `ToPrimitive`, and `Collision`. The first routine implements the transport phase, the last routine implement the collision phase, while the two routines in the middle reconstruct the conserved macroscopic and primitive variables from the kinetic distribution. These two routines are necessary for defining the Maxwellian distribution for performing the collision step in the BGK model, while they only serve to show the results for the Boltzmann model. Moreover, since the macroscopic quantities change only due to the transport phase, one can think to associate the cost of these routines to the transport part. All routines are monitored during a simulation of the vortex problem using the same mesh reported above. The results are reported in Table 5. The Boltzmann model is 20 times more expensive than BGK one. More in details, the relative cost of the collision steps jumps from 70% for

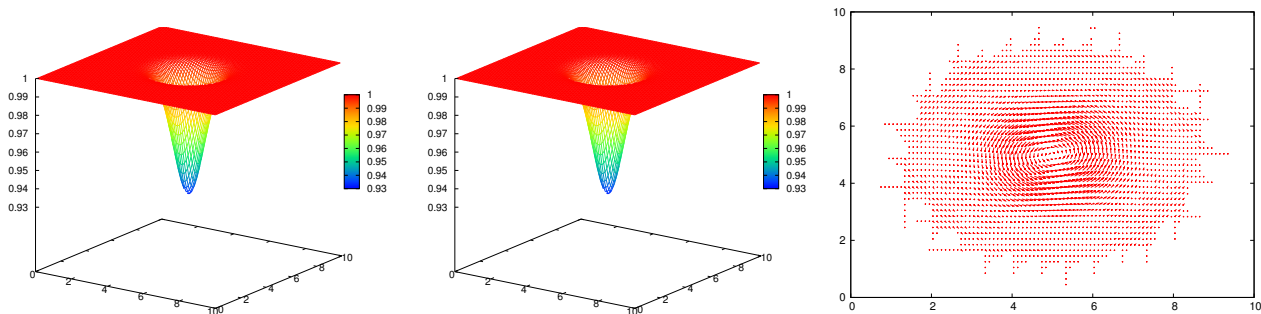


Figure 14: Test 3.1. Two dimensional vortex test case for  $\tau = 10^{-1}$  with  $M = 100 \times 100$  spatial cells and  $N = 32^2$  velocity cells. Initial density, temperature and velocity vector  $(\delta u_x, \delta u_y)$ .

	Cycle	CPU (s)	Main routines	Cost CPU vs total (s)	Cost vs total (%)
<b>BGK</b>	1400	102	Transport	0.36	0.35%
			ToConservative	30.26	29.77%
			ToPrimitive	0.19	0.19%
			Collision	70.83	69.69%
			=	101.64	100%
<b>Boltzmann</b>	1400	2008	Transport	1.06	0.053%
			ToConservative	66.75	3.325%
			ToPrimitive	0.29	0.015%
			Collision	1939.63	96.61%
			=	2007.73	100%

Table 5: Profiling of the average cost for each routine of the code on the test 3.1. (two dimensional vortex problem) simulated on a laptop using  $M = 100^2$  and  $N = 32^2$  points.

BGK model to 97% for Boltzmann model. Indeed, while for the BGK model, the collision routine and the routine recomputing the conservative variables have some impact on the total CPU time, for the Boltzmann model the collision part is the only one participating to the global cost at least for the mesh used. We finally gather in Table 6 the results in terms of computational costs when the number of spatial cell in each direction doubles from  $M = 25^2$  to  $M = 200^2$  keeping the number of velocity cell fixed to  $N = 32 \times 32$  on velocity domain  $[-7.5; 7.5]^2$ . We monitor the number of cycles, the CPU time  $T$  in second(s), and the time per cycle is obtained by  $T_{\text{cycle}} = T/N_{\text{cycle}}$ , the time per cycle per cell by  $T_{\text{cell}} = T/N_{\text{cycle}}/M$  and the time per cycle per degree of freedom  $T_{\text{dof}} = T/N_{\text{cycle}}/(M^d \times N^{d_v})$ . Doubling the number of spatial cells leads to an increase in CPU time by a factor 7 to 8 for both models and the number of time cycles is exactly doubled. Boltzmann model demands on average 20 times more CPU resources than the BGK one. The time per cycle per cell  $T_{\text{cell}}$  is rather constant ( $7.5 \times 10^{-6}$  for BGK and  $1.4 \times 10^{-4}$  for Boltzmann (thanks to the almost linear complexity of the fast spectral solver), it becomes relatively easy to estimate the cost of further refined simulations keeping the velocity mesh fixed.

### 5.3.2. Test 3.2. Re-entry test in two dimensions with changing angle of attack in time.

This test is inspired from re-entry test cases described in [51, 23]. The computational domain is set to  $\Omega = [0; 4] \times [0; 4]$ . Within this domain we initiate three static objects, two small rectangles upfront ( $[x_0; x_1] \times [y_0; y_1]$  and  $[x_0; x_1] \times [y'_0; y'_1]$ ) and one larger one behind ( $[x'_0; x'_1] \times [(y_0 + y'_0)/2; (y_1 + y'_1)/2]$ ) where  $x_0 = 1.5$ ,  $x_1 = 1.7$ ,  $x'_0 = 1.8$ ,  $x'_1 = 2$ ,  $y_0 = 1.7$ ,  $y_1 = 1.95$ ,  $y'_0 = 2.05$ ,  $y'_1 = 2.3$ . The computational mesh in physical space is made of  $800 \times 800$  square cells. The velocity space is  $[-10, 10]$  and is discretized with  $32^2$

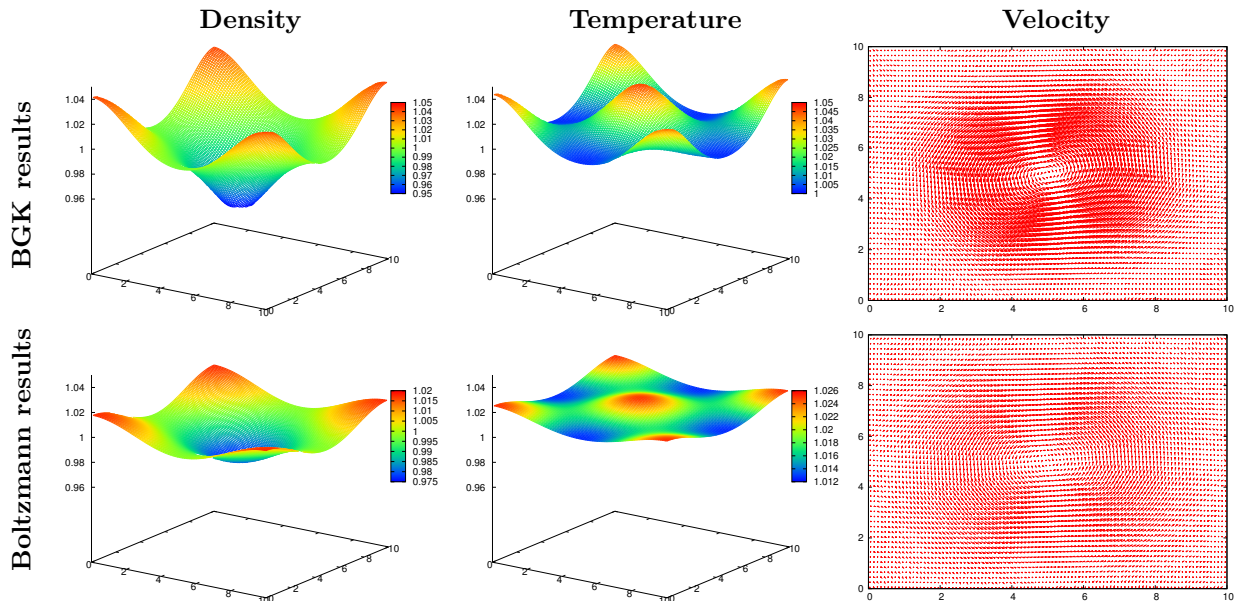


Figure 15: Test 3.1. Two dimensional vortex test case for  $\tau = 10^{-1}$  with  $M = 100 \times 100$  spatial cells and  $N = 32^2$  velocity cells. Top/bottom line: results for BGK/Boltzmann models at  $t_{\text{final}} = 10.0$ . Left to right: density, temperature, velocity vector  $(\delta u_x, \delta u_y)$ .

points. We report the solution obtained in the fluid regime, i.e.  $\tau = 0$ , obtained by projecting the distribution  $f$  over the equilibrium state  $M$  after each transport phase and the results obtained with  $\tau = 10^{-2}$  using both BGK and Boltzmann models. The initial density is set to  $\rho(t=0) = 1$ , the velocity  $(u_x, u_y)(t=0) = (3, 0)$  and the temperature to  $T(t=0) = 1$  everywhere. The final time is set to  $t_{\text{final}} = 10$ . On the boundaries with the objects reflective boundary conditions are employed. Inflow boundary conditions are imposed to the west boundary whereas outflow boundary conditions are set elsewhere. The inflow boundary conditions are evolving in time and equal to

$$(u_x, u_y)_{BC}(t) = \begin{cases} (3, 0) & \text{if } 0 \leq t \leq t_1 \\ (\sqrt{9 - g^2(t)}, g(t)) & \text{if } t_1 \leq t \leq t_2 \\ (\frac{3\sqrt{2}}{2}, \frac{3\sqrt{2}}{2}) & \text{if } t_2 \leq t \leq t_{\text{final}} \end{cases} \quad (56)$$

where  $t_1 = 3/2$ ,  $g(t) = t - t_1$  and  $t_2 = 3\sqrt{2}/2 + t_1$ . Given these initial data, we expect a detached shock wave to occur upfront the objects and some complex wave pattern behind them. Moreover, setting the inflow boundary conditions to (56) splits the simulation into three stages. The first stage consists in the inflow boundary conditions facing the objects up to  $t_1$ . For this stage the upfront detached shock and the complex flow structure behind the objects are formed but they are not yet steady. Next, for the second stage, the inflow boundary condition is changing its direction by smoothly increasing the  $y$  component of  $(u_x, u_y)_{BC}(t)$  up to  $t = t_2$ . Note that this mimics a modification of the angle of attack of the objects with time. This change modifies the entire flow structure. Last, for the third stage the inflow boundary condition is fixed to  $(u_x, u_y)_{BC} = (u_0, v_0)$  up to the final time. As such the flow reaches an almost steady state.

In Figure 17 we present the results for nine intermediate times when  $\tau = 0$  for the density. These results correspond to the fluid limit model that could be obtained when solving compressible Euler equations. Next, in Figures 18 and 19, we compare the BGK results (top row) with the Boltzmann ones (middle row) for intermediate iterations 4500, 12000 and 20000 for  $\tau = 10^{-2}$  for respectively the density and the temperature. In the bottom row, it is shown the differences between the two models in terms of density and temperature. The color legend is the same for the first two rows for both Figures and it can be found in Figure 17, while for the bottom rows they are shown in the pictures.

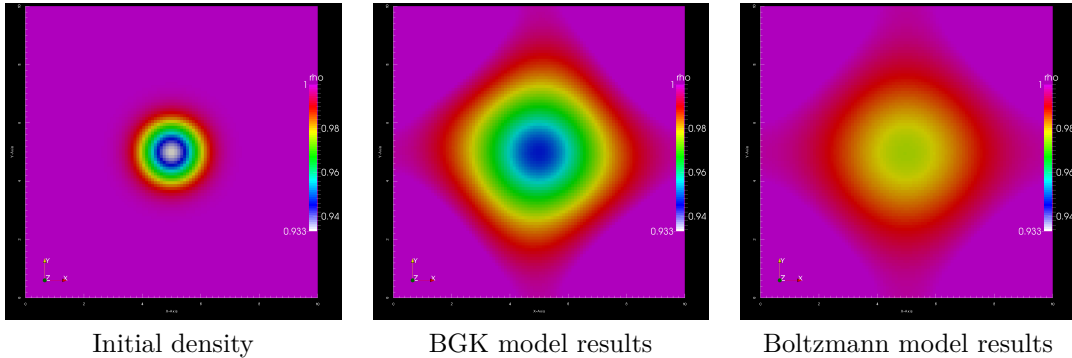


Figure 16: Test 3.1. Two dimensional vortex test case for  $\tau = 10^{-1}$  with  $M = 100 \times 100$  spatial cells and  $N = 32^2$  velocity cells. Left: initial density. Middle/right: results for BGK/Boltzmann model at  $t_{\text{final}} = 10.0$  using the same color scale of the initial density.

In order to conclude this part, let us present some performance data related to those simulations. The total amount of CPU time needed to compute the  $N_{\text{cycle}} = 26000$  cycles for the BGK solution is 14.5h, while for Boltzmann model is 304h= 12.67d. The ratio is of the order 21 in favor of BGK consistently with the previous simulations. However, even if Boltzmann results demand a large amount of CPU time, we have seen that discrepancies do exist with respect to BGK model and in some cases, especially far from equilibrium, they cannot be ignored.

#### 5.4. Part 4. Numerical results for the space three dimensional case.

In this last part, we present one numerical test in which we compare the relaxation model with the Boltzmann model in three space and velocity dimensions in a kinetic regime. Solving the full Boltzmann equation in three dimensions is extremely resource consuming, even if the fast spectral methods is used and consequently shared memory systems are not sufficient for this kind of problems. For this reason, only for this last case we adapted the method to distributed memory systems by employing MPI architecture as already stated and described in Section 4. The results reported for this situation are not to be intended as optimal since we are adapting the scheme to this kind of architecture and improvements in terms of efficiency are attended in the next future.

##### 5.4.1. Test 4.1. Three dimensional re-entry test case.

The computational domain is set to  $\Omega = [0, 2]^3$  with a static cuboid placed in the center (see Fig.20). The velocity space is  $[-10, 10]^3$  and discretized with  $32^3$  points. The relaxation parameter is set to  $\tau = 0.3$ . The initial density  $\rho$  is set to 1, the temperature  $T = 1$  while the initial velocity is given by  $(u_x, u_y, u_z) = (2, 0, 0)$ . The final time is set to  $t_{\text{final}} = 0.6$  leading to 379 time steps. The inflow boundary conditions are imposed on the left boundary ( $x = 0$ ) while outflow boundary conditions on the remaining part of the boundary are imposed. Hard sphere molecules are considered for Boltzmann while for the BGK model the frequency  $\nu$  is taken equal to  $\mu = C_\alpha 4\pi(2\lambda\pi)^\alpha$ . For both models the CFL condition considered is consequently given by

$$\Delta t \leq \min \left( \frac{\Delta x}{|v_{max}|}, \frac{\tau}{\mu} \right). \quad (57)$$

The results are shown for the temperature and the density in Figure 20 while the discrepancies between the two in Figure 21. From the analysis of such results it clearly emerge a difference in the profiles of the macroscopic quantities between the two models. We now analyze the performances. This test case was run on the EOS supercomputer at CALMIP, Toulouse France (<https://www.calmip.univ-toulouse.fr/>). The supercomputer is equipped with 612 computational nodes, each of them containing two Intel® Ivybridge 2.8GHz 10 core CPUs and 64 GB of RAM. Each CPU is equipped with 25MB of cache memory. The code



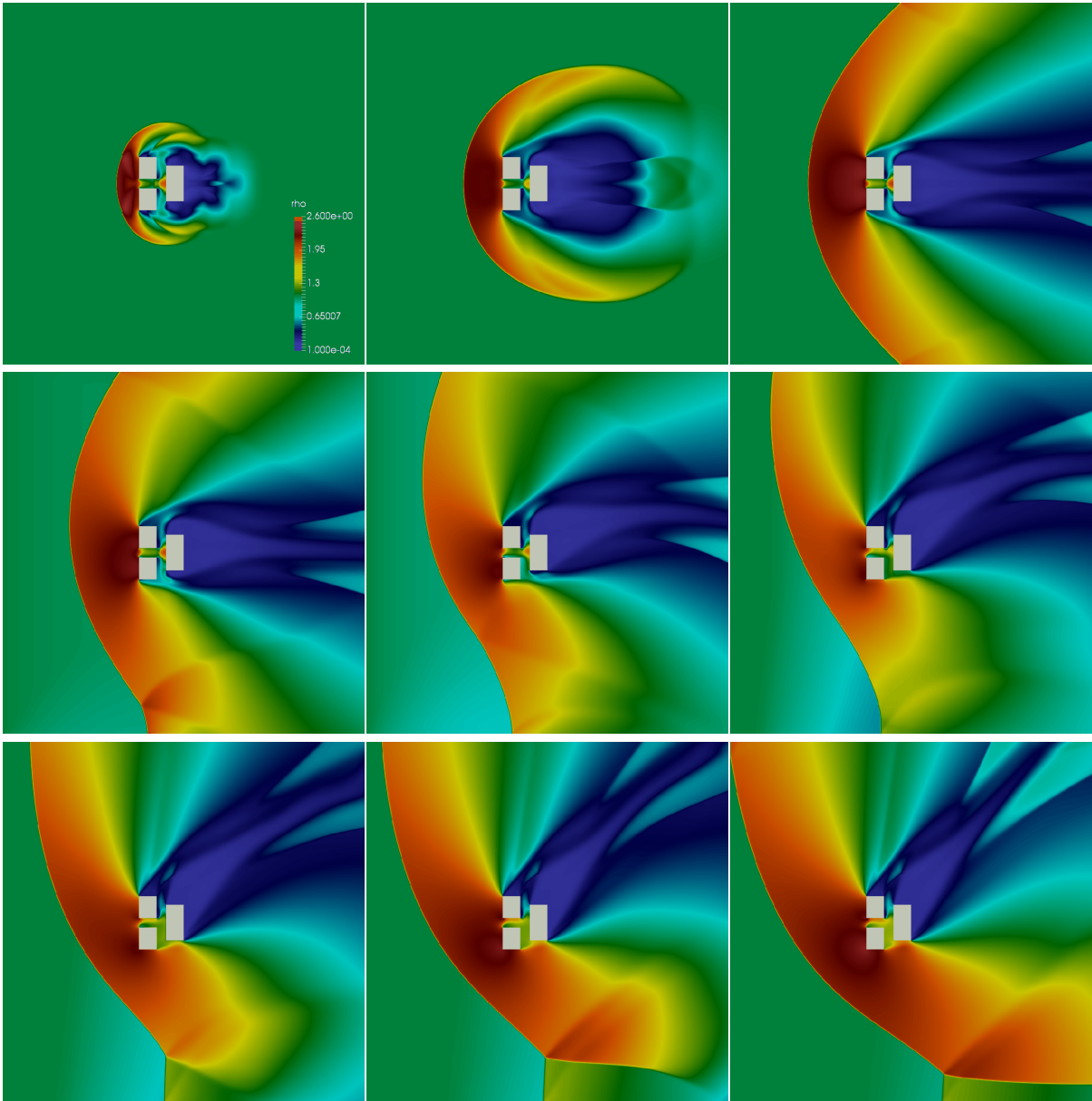


Figure 17: Test 3.2. Two dimensional re-entry test case for  $\tau = 0$  with  $M = 800 \times 800$  spatial cells and  $N = 32^2$  velocity cells. Density profile. Top-left to bottom-right iterations 500, 1400, 4500, 8000, 10000, 12000, 14000, 16000, 20000.

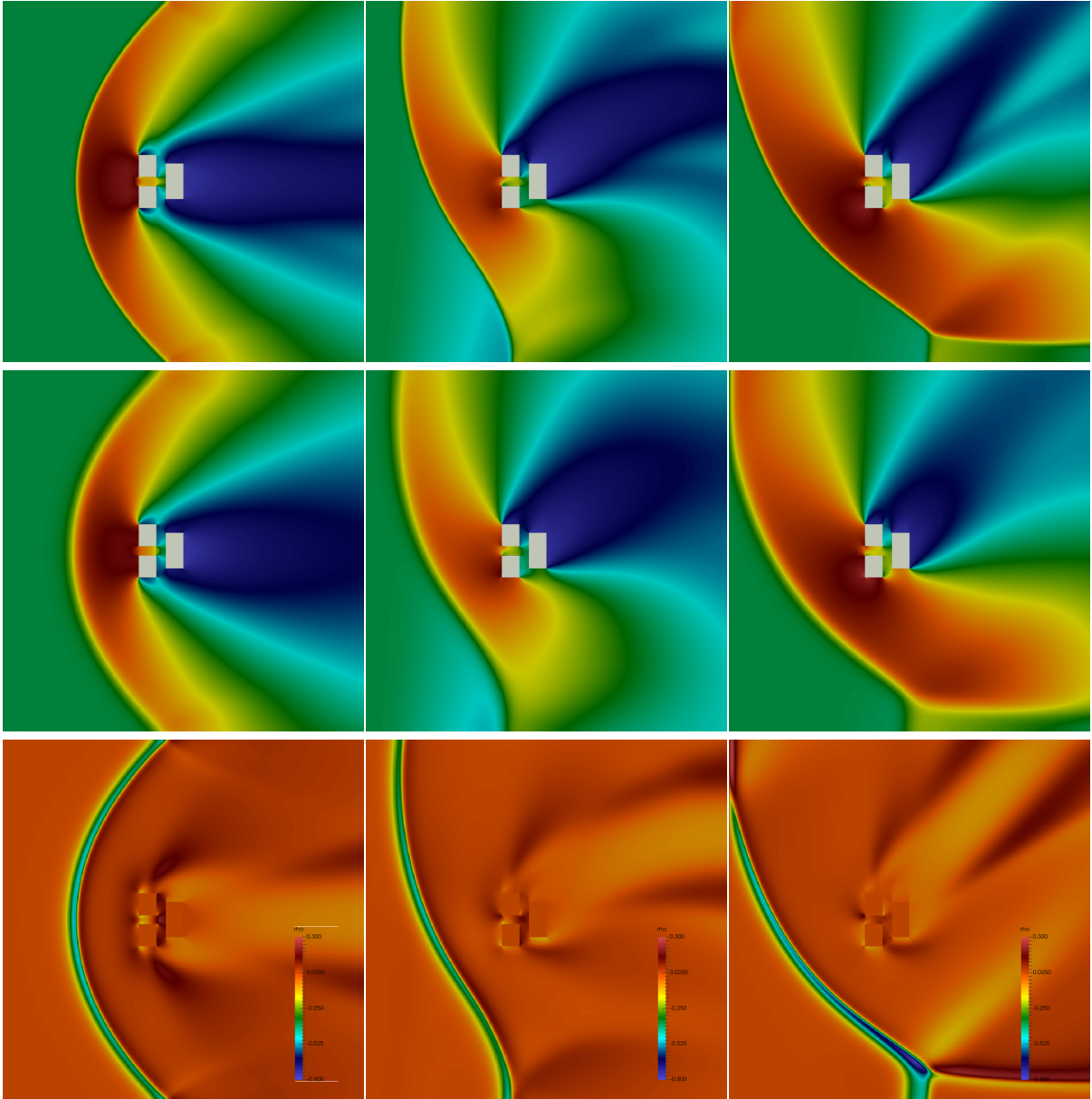


Figure 18: Test 3.2. Two dimensional re-entry test case for  $\tau = 10^{-2}$  with  $M = 800 \times 800$  spatial cells and  $N = 32^2$  velocity cells. BGK model (top row) Boltzmann model (middle row), difference between the two models (bottom row) at iterations 4500, 12000 and 20000. The color legend for the first two rows can be found in figure 17. Density profile.

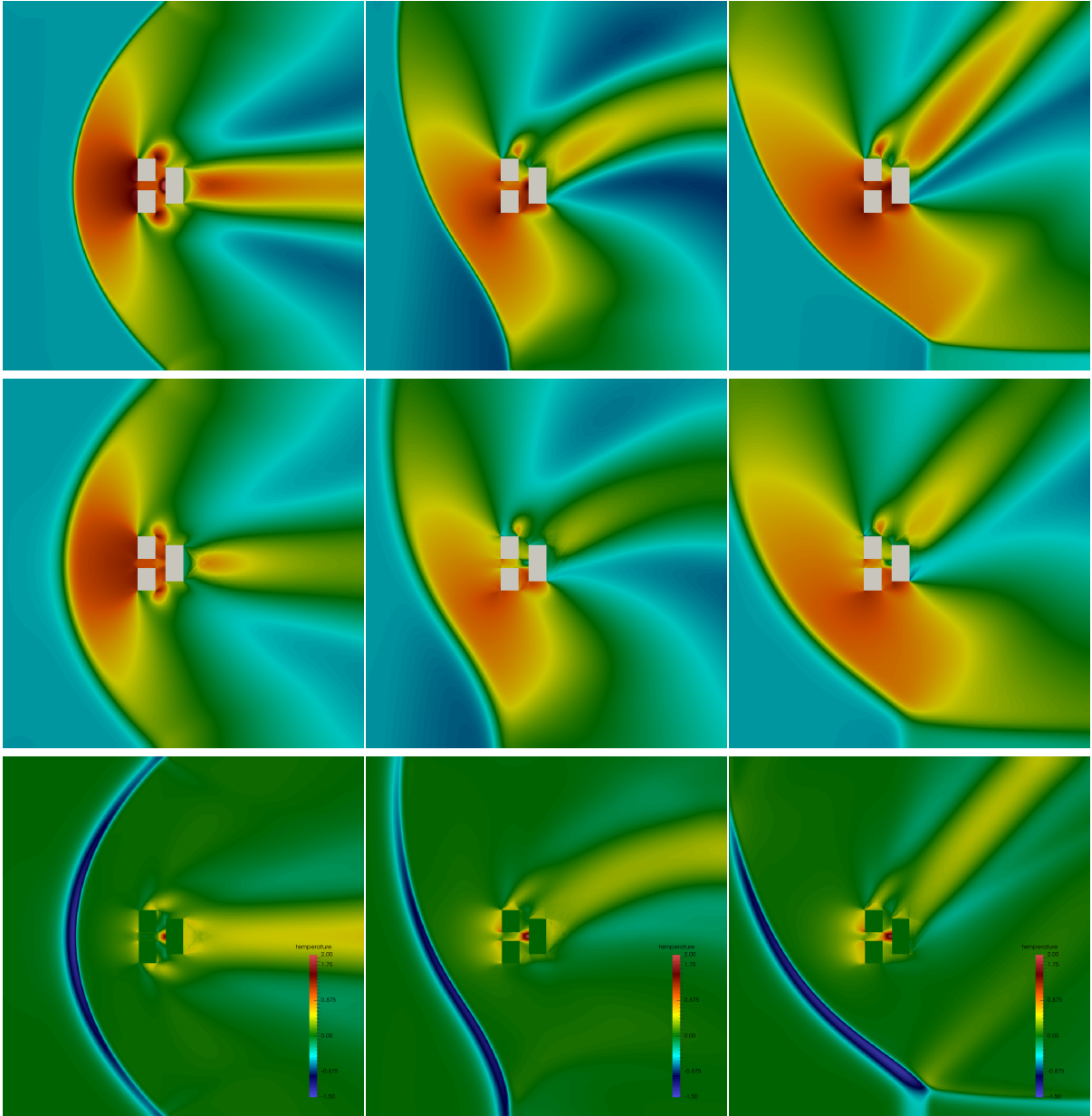


Figure 19: Test 3.2. Two dimensional re-entry test case for  $\tau = 10^{-2}$  with  $M = 800 \times 800$  spatial cells and  $N = 32^2$  velocity cells. BGK model (top row) Boltzmann model (middle row), difference between the two models (bottom row) at iterations 4500, 12000 and 20000. The color legend for the first two rows can be found in figure 17. Temperature profile.

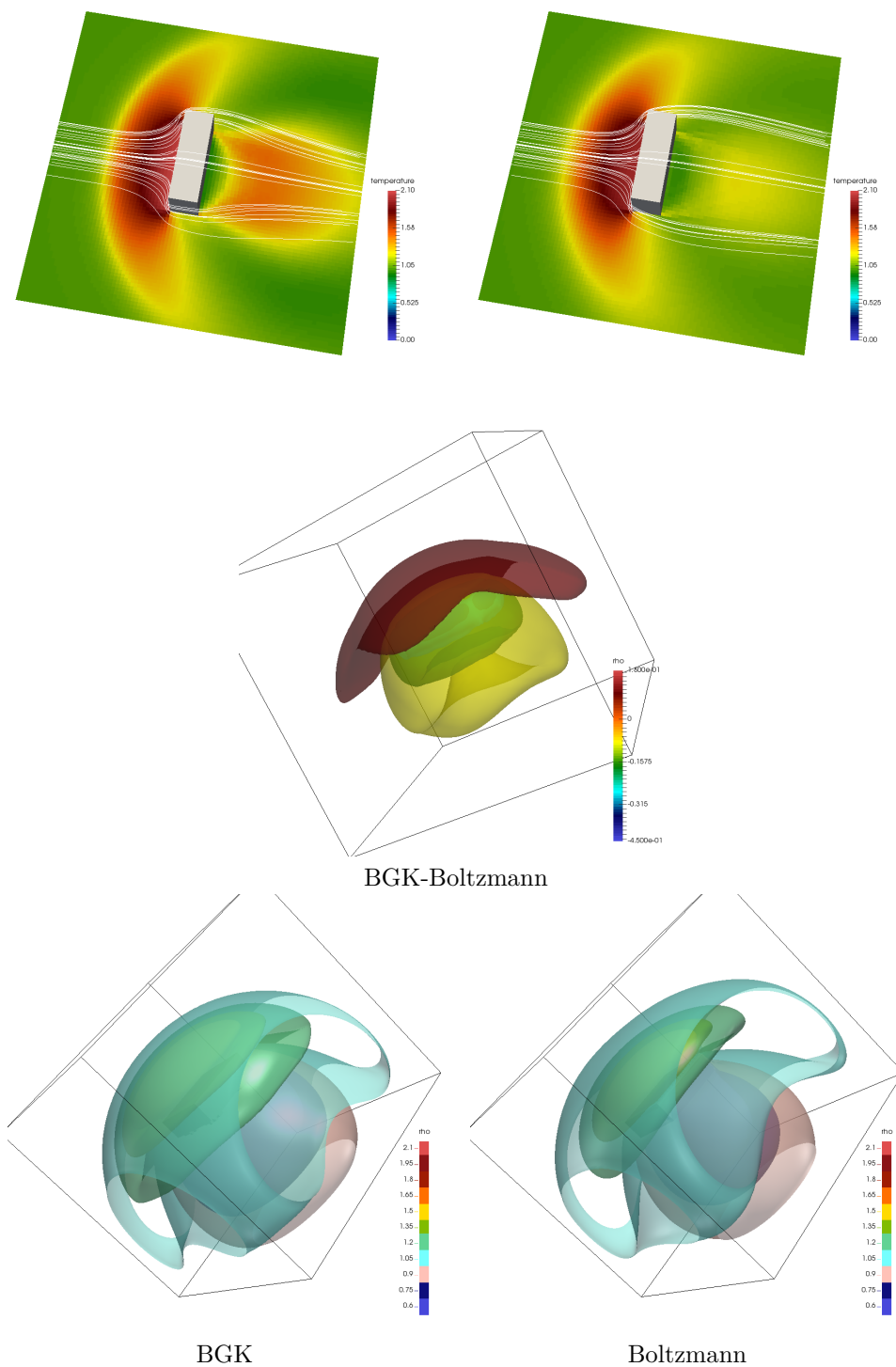


Figure 20: Test 4.1. Three dimensional re-entry test case for  $\tau = 0.3$  with  $M = 90 \times 90 \times 90$  spatial cells and  $N = 32^3$  velocity cells. BGK model (left column), Boltzmann model (right column) at time  $t_{\text{final}} = 0.6$ . Top row: temperature field with velocity streamlines, bottom row: isosurfaces of the density. Middle row: isosurfaces of the density difference between BGK and Boltzmann models.

Vortex problem in two dimension								
Model	Velocity		Cell #	Cycle	Time	T/cycle	T/cell	T/d.o.f
	$N$	Vel.	$M^d \times N^{d_v}$	$N_{\text{cycle}}$	T (s)	$T_{\text{cycle}}$ (s)	$T_{\text{cell}}$ (s)	$T_{\text{dof}}$ (s)
BGK	$32^2$	[-7.5, 7.5]	$25^2 \times 32^2$ $= 64 \times 10^4$	351	2.61	0.0035	$5.60 \times 10^{-6}$	$5.47 \times 10^{-9}$
			$50^2 \times 32^2$ $= 256 \times 10^4$	701	13.77	0.0196	$7.84 \times 10^{-6}$	$7.66 \times 10^{-9}$
			$100^2 \times 32^2$ $= 1024 \times 10^4$	1400	102.42	0.0732	$7.32 \times 10^{-6}$	$7.15 \times 10^{-9}$
			$200^2 \times 32^2$ $= 4096 \times 10^4$	2800	785.54	0.2806	$7.02 \times 10^{-6}$	$6.85 \times 10^{-9}$
Boltzmann	$32^2$	[-7.5, 7.5]	$25^2 \times 32^2$ $= 64 \times 10^4$	351	32.92	0.0938	$1.50 \times 10^{-4}$	$1.47 \times 10^{-7}$
			$50^2 \times 32^2$ $= 256 \times 10^4$	701	245.03	0.350	$1.40 \times 10^{-4}$	$1.37 \times 10^{-7}$
			$100^2 \times 32^2$ $= 1024 \times 10^4$	1400	2008.56	1.435	$1.44 \times 10^{-4}$	$1.40 \times 10^{-7}$
			$200^2 \times 32^2$ $= 4096 \times 10^4$	2800	15762	5.630	$1.41 \times 10^{-4}$	$1.37 \times 10^{-7}$

Table 6: Test 3.1. Two dimensional vortex problem simulations with spatial mesh variation. Monitoring of CPU time. Time per cycle is obtained by  $T_{\text{cycle}} = T/N_{\text{cycle}}$ , time per cycle per cell by  $T_{\text{cell}} = T/N_{\text{cycle}}/M$  and time per cycle per degree of freedom  $T_{\text{dof}} = T/N_{\text{cycle}}/(M^d \times N^{d_v})$ .

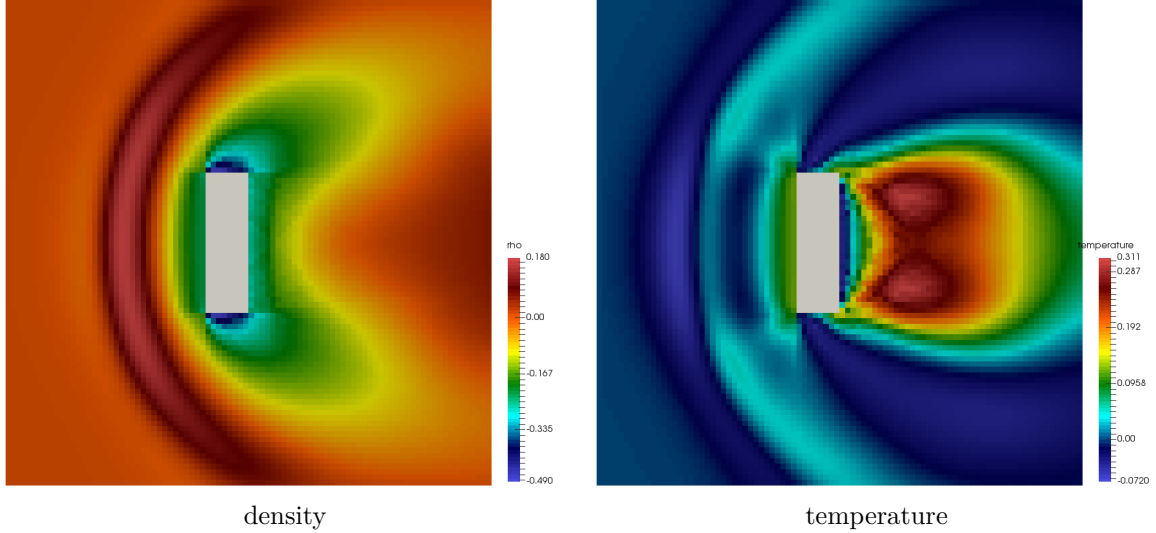


Figure 21: Test 4.1. Three dimensional re-entry test case for  $\tau = 0.3$  with  $M = 90 \times 90 \times 90$  spatial cells and  $N = 32^3$  velocity cells. Discrepancies between the BGK solution and the Boltzmann solution at time  $t_{\text{final}} = 0.6$ . Density on the left and temperature on the right.

	Cycle	CPU (s)	Main routines	Cost CPU vs total (s)	Cost vs total (%)
<b>BGK</b>	379	2174	Transport	0.03	<0.1%
			ToConservative	1126	52%
			ToPrimitive	0.06	<0.1%
			Collisions	217	10%
			MPI comm.	825	38%
			=	2168	100%
<b>Boltzmann</b>	379	93713	Transport	0.5	<0.1%
			ToConservative	1127	1.2%
			ToPrimitive	0.07	<0.1%
			Collisions	89396	95.4%
			MPI comm.	3190	3.4%
			=	93713	100%

Table 7: Profiling of the average cost for each routine of the code on the test 4.1. (three dimensional re-entry test case) simulated using  $M = 90^3$  and  $N = 32^3$  points.

was compiled with gcc-5.3.0 and executed on 90 computational nodes. That is to say, on 1800 computational cores in parallel. In the case of the Boltzmann collision operator the Fast Fourier Transforms were computed by means of the fftw library, version 3.3.4. The total runtime was equal to  $t = 93713s$  (26h) for the Boltzmann model. This is equivalent to 46000 computational hours (3.25 years) on a sequential machine. On the other hand, the runtime for the BGK model was only  $t = 2174s$  (0.6h) with a ratio 43 in favor of the simpler relaxation model. The profiling data are summarized in Table 7. The communications and MPI synchronization take 38% of the computational time for BGK equation. On the other hand, for the Boltzmann model the time spent on communications and synchronization amounts merely to 3.4% of the total runtime. This is due to extreme computational complexity of the three dimensional Boltzmann collision kernel. As discussed in the previous paragraphs, the costs relatives to the routines `ToConservative`, `ToPrimitive` which are relative to the computation of the macroscopic variables from the distribution function can be imputed to the transport phase, since this is the only part of the scheme which modifies these quantities, since collisions preserves density, momentum and energy in the cell.

## 6. Conclusion and perspectives

In this paper we have generalized the Fast Kinetic Scheme [21, 22, 24, 30] to the challenging case of the Boltzmann collision integral. We have shown that it is possible to solve the full unsteady three dimensional Boltzmann equation with variable hard sphere kernel in a reasonable amount of time by using parallel architectures. Up to the author knowledge, the results reported in this paper represent one of the very first attempts of solving the seven dimensional Boltzmann model with deterministic numerical schemes. This has been made possible by combining a fast semi-Lagrangian approach for the transport part with a fast spectral method for the collision dynamic.

We have performed several numerical tests with the aim of detailing the behavior of the method in different situations in order to understand its strengths and weaknesses. A side scope of the paper has been to show the differences that arise between the Boltzmann model and the popular BGK relaxation model.

Differences have been observed to be large far from equilibrium situations.

In the future, we aim in working in the direction of additionally improving the fast spectral method since we have observed it to be one of the main bottleneck of the scheme by using for instance different grids during the transport and the collision phases. Another fundamental direction we aim to pursue is the optimization of the MPI parallelization which is necessary for considering more realistic applications as

well as the development of techniques for treating complex boundaries. Finally, the extension of the present scheme to plasmas is under study.

## Acknowledgments

The authors would like to thank Professor Francis Filbet from the University of Toulouse III, Professor Rémi Abgrall and Dott. Tulin Kaman from the institute of mathematics in Zürich, Switzerland for their suggestions and help.

This work has been supported by the Galileo project G14 (Fast Asymptotic-Preserving and semi-Lagrangian schemes for High Performance Computing : applications to plasmas) from the Franco-Italian University and by the ANR project MOONRISE (MOdels, Oscillations and NUMerical SchEmes, 2015-2019). Thomas Rey was partially funded by Labex CEMPI (ANR-11-LABX-0007-01).

This work was granted access to the HPC resources of CALMIP supercomputing center under the allocation 2016-P1542 and the authors acknowledge the help from CALMIP.

## References

- [1] C. Cercignani, The Boltzmann equation and its applications, Vol. 67 of Applied Mathematical Sciences, Springer-Verlag, New York, 1988.
- [2] G. Dimarco, L. Pareschi, Numerical methods for kinetic equations, *Acta Numer.* 23 (2014) 369–520.
- [3] F. Filbet, C. Mouhot, L. Pareschi, Solving the Boltzmann equation in  $N \log^2 N$ , *SIAM J. Sci. Comput.* 28 (3) (2007) 1029–1053.
- [4] L. Pareschi, G. Russo, Numerical solution of the Boltzmann equation I: Spectrally accurate approximation of the collision operator, *SIAM J. Numer. Anal.* 37 (4) (2000) 1217–1245.
- [5] G. Dimarco, L. Pareschi, High order asymptotic-preserving schemes for the Boltzmann equation, *C. R. Math. Acad. Sci. Paris* 350 (9-10) (2012) 481–486.
- [6] G. Dimarco, L. Pareschi, Asymptotic preserving implicit-explicit Runge-Kutta methods for nonlinear kinetic equations, *SIAM J. Numer. Anal.* 51 (2) (2013) 1064–1087.
- [7] S. Jin, Asymptotic preserving (AP) schemes for multiscale kinetic and hyperbolic equations: a review, *Riv. Math. Univ. Parma (N.S.)* 3 (2) (2012) 177–216.
- [8] S. Jin, Runge-Kutta methods for hyperbolic conservation laws with stiff relaxation terms, *J. Comp. Phys.* 122 (1995) 51–67.
- [9] S. Jin, Efficient asymptotic-preserving (ap) schemes for some multiscale kinetic equations, *SIAM J. Sci. Comput.* 21 (1999) 441–454.
- [10] M. Bennoune, M. Lemou, L. Mieussens, Uniformly stable numerical schemes for the Boltzmann equation preserving the compressible Navier-Stokes asymptotics, *J. Comp. Phys.* 227 (2008) 3781–3803.
- [11] P. Degond, Asymptotic-preserving schemes for fluid models of plasmas, *Panoramas et Synthèses SMF*.
- [12] G. Bird, Molecular gas dynamics and the direct simulation of gas flows, 2nd Edition, Oxford University Press, 1994.
- [13] R. E. Caflisch, Monte carlo and quasi-monte carlo methods, *Acta numerica* 7 (1998) 1–49.
- [14] N. Crouseilles, T. Respaud, E. Sonnendrücker, A forward semi-lagrangian method for the numerical solution of the vlasov equation, *Computer Physics Communications* 180 (10) (2009) 1730–1745.
- [15] N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, Conservative semi-Lagrangian schemes for Vlasov equations, *Journal of Computational Physics* 229 (6) (2010) 1927–1953.
- [16] Y. Güçlü, W. N. G. Hitchon, A high order cell-centered semi-Lagrangian scheme for multi-dimensional kinetic simulations of neutral gas flows, *J. Comput. Phys.* 231 (8) (2012) 3289–3316.
- [17] M. Shoucri, G. Knorr, Numerical integration of the Vlasov equation, *J. Computational Phys.* 14 (1) (1974) 84–92.
- [18] F. Filbet, G. Russo, Accurate numerical methods for the Boltzmann equation, in: *Modeling and computational methods for kinetic equations*, Springer, 2004, pp. 117–145.
- [19] F. Filbet, G. Russo, High order numerical methods for the space non-homogeneous Boltzmann equation, *J. Comput. Phys.* 186 (2) (2003) 457–480.
- [20] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, *Spectral methods in fluid dynamics*, Springer Series in Computational Physics, Springer-Verlag, New York, 1988.
- [21] G. Dimarco, R. Loubère, Towards an ultra efficient kinetic scheme. Part I: Basics on the BGK equation, *Journal of Computational Physics* 255 (2013) 680–698.
- [22] G. Dimarco, R. Loubère, Towards an ultra efficient kinetic scheme. Part II: The high order case, *Journal of Computational Physics* 255 (2013) 699–719.
- [23] G. Dimarco, R. Loubère, V. Rispoli, A multiscale fast semi-Lagrangian method for rarefied gas dynamics, *Journal of Computational Physics* 291 (2015) 99–119.



- [24] G. Dimarco, R. Loubère, J. Narski, Towards an ultra efficient kinetic scheme. Part III: High-performance-computing, *Journal of Computational Physics* 284 (2015) 22–39.
- [25] P. Bhatnagar, E. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (3) (1954) 511–525.
- [26] A. V. Bobylev, A. Palczewski, J. Schneider, On approximation of the Boltzmann equation by discrete velocity models, *C. R. Acad. Sci. Paris Sér. I Math.* 320 (5) (1995) 639–644.
- [27] A. Palczewski, J. Schneider, A. V. Bobylev, A consistency result for a discrete-velocity model of the Boltzmann equation, *SIAM journal on numerical analysis* 34 (5) (1997) 1865–1883.
- [28] A. Palczewski, J. Schneider, Existence, stability, and convergence of solutions of discrete velocity models to the Boltzmann equation, *Journal of statistical physics* 91 (1-2) (1998) 307–326.
- [29] 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.
- [30] G. Dimarco, C. Hauck, R. Loubère, A new class of high order semi-Lagrangian schemes for rarefied gas dynamics, Submitted.
- [31] L. Pareschi, B. Perthame, A Fourier spectral method for homogeneous Boltzmann equations, *Transport Theory Statist. Phys.* 25 (3) (1996) 369–382.
- [32] A. V. Bobylev, S. Rjasanow, Difference scheme for the Boltzmann equation based on the fast Fourier transform, *Eur. J. Mech. B Fluids* 16 (2) (1997) 293–306.
- [33] I. M. Gamba, S. H. Tharkabhushanam, Shock and boundary structure formation by spectral-Lagrangian methods for the inhomogeneous Boltzmann transport equation, *J. Comput. Math.* 28 (4) (2010) 430–460.
- [34] I. M. Gamba, S. H. Tharkabhushanam, Spectral-Lagrangian methods for collisional models of non-equilibrium statistical states, *J. Comput. Phys.* 228 (6) (2009) 2012–2036.
- [35] I. M. Gamba, J. R. Haack, A conservative spectral method for the Boltzmann equation with anisotropic scattering and the grazing collisions limit, *J. Comput. Phys.* 270 (2014) 40–57.
- [36] L. Wu, C. White, T. J. Scanlon, J. M. Reese, Y. Zhang, Deterministic numerical solutions of the Boltzmann equation using the fast spectral method, *Journal of Computational Physics* 250 (2013) 27–52.
- [37] A. V. Bobylev, S. Rjasanow, Numerical solution of the Boltzmann equation using a fully conservative difference scheme based on the fast Fourier transform, in: *Proceedings of the Fifth International Workshop on Mathematical Aspects of Fluid and Plasma Dynamics* (Maui, HI, 1998), Vol. 29, 2000, pp. 289–310.
- [38] L. Pareschi, G. Toscani, C. Villani, Spectral methods for the non cut-off Boltzmann equation and numerical grazing collision limit, *Numer. Math.* 93 (3) (2003) 527–548.
- [39] F. Filbet, C. Mouhot, Analysis of spectral methods for the homogeneous Boltzmann Equation, *Trans. Amer. Math. Soc.* 363 (2011) 1947–1980.
- [40] L. Wu, H. Liu, Y. Zhang, J. M. Reese, Influence of intermolecular potentials on rarefied gas flows: Fast spectral solutions of the Boltzmann equation, *Physics of Fluids* (1994-present) 27 (8) (2015) 082002.
- [41] A. Alekssenko, E. Josyula, Deterministic solution of the Boltzmann equation using a discontinuous Galerkin velocity discretization, in: *A. C. P. A. I. of Physics* (Ed.), *Proceedings of the 28th International Symposium on Rarefied Gas Dynamics*, Vol. 1501, 2012, pp. 279–286.
- [42] L. Pareschi, G. Russo, On the stability of spectral methods for the homogeneous Boltzmann equation, *Trans. Theo. Stat. Phys.* 29 (2000) 431–447.
- [43] L. Wu, J. Zhang, J. M. Reese, Y. Zhang, A fast spectral method for the Boltzmann equation for monatomic gas mixtures, *Journal of Computational Physics* 298 (2015) 602–621.
- [44] C. Mouhot, L. Pareschi, Fast algorithms for computing the Boltzmann collision operator, *Math. Comp.* 75 (256) (2006) 1833–1852 (electronic).
- [45] L. Desvillettes, S. Mischler, About the splitting algorithm for Boltzmann and B.G.K. equations, *Math. Models Methods Appl. Sci.* 6 (8) (1996) 1079–1101.
- [46] F. Filbet, E. Sonnendrücker, P. Bertrand, Conservative numerical schemes for the Vlasov equation, *J. Comput. Phys.* 172 (1) (2001) 166–187.
- [47] T. Carleman, Sur la théorie de l'équation intégrodifférentielle de Boltzmann, *Acta Math.* 60 (1) (1933) 91–146.
- [48] A. V. Bobylev, Exact solutions of the Boltzmann equation, *Dokl. Akad. Nauk SSSR* 225 (6) (1975) 1296–1299.
- [49] M. Krook, T. T. Wu, Exact solutions of the Boltzmann equation, *Phys. Fluids* 20 (10) (1977) 1589.
- [50] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: *Quarteroni* (Ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Springer, 1998, pp. 325–432.
- [51] S. Prasad, G. Srinivas, Flow simulation over re-entry bodies at supersonic and hypersonic speeds, *International Journal of Engineering Research and Development* 2 (2012) 29–34.