

Two Dimensional Local Adaptive Discrete Velocity Grids For Rarefied Flow Simulations

S. Brull¹, L. Forestier-Coste¹ and L. Mieussens^{1,2,a)}

¹*Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France*

²*INRIA, F-33400 Talence, France*

^{a)}Corresponding author: Luc.Mieussens@math.u-bordeaux.fr

Abstract. We propose a deterministic method designed for unsteady flows, based on a discretization of the Boltzmann (BGK) equation with local adaptive velocity grids. These grids dynamically adapt in time and space to the variations of the width of the distribution functions. This allows a significant reduction of the memory storage and CPU time, as compared to standard discrete velocity methods, and avoid the delicate problem to construct a priori a sufficient global velocity grid.

INTRODUCTION

Most deterministic numerical methods for rarefied gas dynamics are based on a common idea: the kinetic equation is discretized with a finite set of discrete velocities (see [1]). This set is generally given by a global Cartesian velocity grid, which is the same grid for every point in the physical domain, and for every time. The advantages of this approach are its simplicity and the strong mathematical properties that inherits the discrete model from the continuous one (stability, positivity, etc.). This is due to the fact that all the distribution functions are discretized on the same grid. However, for some practical problems with strong variations of macroscopic temperature and velocity fields, like in atmospheric re-entry flows, this approach is very expensive: the discrete velocity grid must be very large and very thin in order to capture all the different distribution functions.

Recently, several people proposed to use instead local velocity grids: for each point in space, and for each time, the grid is refined or coarsened for an optimal representation of the distribution functions: see Kolobov et al. [2], Chen et al. [3], Bernard et al. [4]. However, in all these papers, the local grids have all the same bounds, that has to be determined a priori. In some sense, all the local grids are obtained by a local refinement or coarsening of an initial global grid. This means that these methods still require to find, a priori, a grid which is sufficiently large to contain all the distributions. Moreover, since all the local grids have the same bounds, some of them might be unnecessary large, and hence contain more points than necessary. This happens in particular for hypersonic reentry flows, for which the flow shows both very thin and very large distributions, centered on small and very large velocities.

In [5], we have proposed an approach in which the bounds of the local velocity grids may vary in time and space, and hence adapt to the width of the local distribution functions, by using the evolution of the macroscopic velocity and of the temperature. This solves the two aforementioned problems, and allows to simulate test cases that can hardly be solved with standard discrete velocity methods. However, this work was made in the monodimensional case (1D in space and 1D in velocity) only. In this new paper, this method is extended to multidimensional problems. This extension requires a major modification, since we observed that the method of [5] induces systematic interpolations, which is much too expensive in 2D. In the following sections, this new method is described, and it is illustrated with several 1D problems in space, but 2D in velocity (shock wave, Couette flows), and for a 2D re-entry problem.

Presentation of the method

To simplify the presentation of the method, everything will be presented in a 1D like setting, even if the method will be used in 2D for the simulations given at the end of the paper.

The Boltzmann equation is

$$\partial_t f + v \partial_x f = Q(f), \quad (1)$$

where $Q(f)$ is the collision operator, and $f(t, x, v)$ is the distribution function at time t , position x , and velocity v . We shall need the macroscopic values like mass density $\rho(t, x)$, velocity $u(t, x)$, and temperature $T(t, x)$, defined through the moment vector of f , defined by

$$U(t, x) = \langle mf \rangle := \int_{\mathbb{R}} m(v) f(t, x, v) dv, \quad (2)$$

where $m(v) = (1, v, \frac{1}{2}|v|^2)^T$ is the vector of collisional invariants. Indeed, we have $U = (\rho, \rho u, \frac{1}{2}\rho|u|^2 + \frac{3}{2}\rho RT)$.

The time variable will be discretized by $t_n = n\Delta t$, and the space variable by $x_i = i\Delta x$. At each t_n and x_i , we will define a local velocity grid $\mathcal{V}_i^n = (v_{i,k}^n)$ of K_i^n points, and the distribution function $f(t_n, x_i, v)$ will be discretized by $f_i^n = (f_{i,1}^n, f_{i,2}^n, \dots, f_{i,K_i^n}^n)$. The moment vector U_i^n of this discrete distribution will be defined by using a standard quadrature formula applied to (2):

$$U_i^n = \langle m f_i^n \rangle_{\mathcal{V}_i^n} := \sum_{k=1}^{K_i^n} m(v_{i,k}^n) f_{i,k}^n \omega_{i,k}^n, \quad (3)$$

where $\omega_{i,k}^n$ are the weights of the quadrature.

Definition of the local velocity grids at the initial time

Usually, the initial distribution $f(t=0, x, v)$ is a Maxwellian defined by the macroscopic quantities $\rho^0(x)$, $u^0(x)$, $T^0(x)$. For a given position x , it is known that 99% of this distribution is contained in the interval $[u^0(x) - 4\sqrt{RT^0(x)}, u^0(x) + 4\sqrt{RT^0(x)}]$. Therefore, for each space cell i (of center x_i), the initial distribution can be accurately approximated with a Cartesian grid \mathcal{V}_i^0 , defined with bounds $v_{i,min}^0 = u^0(x_i) - 4\sqrt{RT^0(x_i)}$ and $v_{i,max}^0 = u^0(x_i) + 4\sqrt{RT^0(x_i)}$, and a step $\Delta v_i^0 = \alpha\sqrt{RT_i^0}$, with α such that the grid contains around 10 points. In 2D or 3D, this definition must be understood component-wise, that is to say in each direction of the velocity space.

Construction of the local grids at time t_{n+1}

Here, we assume that at time t_n , the distributions f_i^n and their corresponding local velocity grids \mathcal{V}_i^n are known in every cell i . To define the local grids at the next time t_{n+1} , the idea is to note that it is possible to compute the moments U_i^{n+1} of f_i^{n+1} before the distributions f_i^{n+1} are computed. Indeed, it is standard to derive the following conservation laws

$$\partial_t U + \partial_x \langle v m f \rangle = 0 \quad (4)$$

from the Boltzmann equation (1). These equations can be discretized by any finite volume upwind scheme. For instance, a first order scheme gives

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + \frac{\Phi_{i+\frac{1}{2}}^n - \Phi_{i-\frac{1}{2}}^n}{\Delta x} = 0, \quad (5)$$

where $\Phi_{i+\frac{1}{2}}^n$ is the numerical flux across the interface between cell i and cell $i+1$ defined by

$$\Phi_{i+\frac{1}{2}}^n = \langle v^+ m f_i^n \rangle_{\mathcal{V}_i^n} + \langle v^- m f_{i+1}^n \rangle_{\mathcal{V}_{i+1}^n}, \quad (6)$$

which is composed of two half fluxes in which each distribution is integrated on its own local velocity grid. Note that here, we use the standard notation $v = (v \pm |v|)/2$ for the positive (resp. negative) part of v .

Now, U_i^{n+1} can be used to define the velocity and temperature u_i^{n+1} and T_i^{n+1} at time t_{n+1} . Like at the initial time, these variables can be used to define the new local velocity grid \mathcal{V}_i^{n+1} . The bounds are

$$v_{min,i}^{n+1} = u_i^{n+1} - 4\sqrt{RT_i^{n+1}} \quad \text{and} \quad v_{max,i}^{n+1} = u_i^{n+1} + 4\sqrt{RT_i^{n+1}} \quad (7)$$

while the step is

$$\Delta v_i^{n+1} = \alpha\sqrt{RT_i^{n+1}}. \quad (8)$$

Computation of f_i^{n+1}

Now, we are ready to compute f_i^{n+1} by discretizing the Boltzmann equation (1). Again, we shall use a finite volume upwind scheme. However, note that here we need to use some interpolation technique, since all the distributions are not defined on the same velocity grid. The scheme reads

$$\frac{f_{i,k}^{n+1} - I(f_i^n, v_{i,k}^{n+1})}{\Delta t} + (v_{i,k}^{n+1})^+ \frac{I(f_i^n, v_{i,k}^{n+1}) - I(f_{i-1}^n, v_{i,k}^{n+1})}{\Delta x} + (v_{i,k}^{n+1})^- \frac{I(f_{i+1}^n, v_{i,k}^{n+1}) - I(f_i^n, v_{i,k}^{n+1})}{\Delta x} = I(Q(f_i^n), v_{i,k}^{n+1}), \quad (9)$$

for every $v_{i,k}^{n+1}$ of \mathcal{V}_i^{n+1} , where I denotes an interpolation operator on the local grid \mathcal{V}_i^{n+1} such that $I(f_i^n, v_{i,k}^{n+1})$ is the interpolation of f_i^n at point $v_{i,k}^{n+1}$.

Merging and reduction of embedded grids

The advantage of the previous approach are: (1) the local velocity grids adapt in time and space to the local temperature T and velocity u , and (2) only initial values for u and T are required.

However, this method has two drawbacks. First, in non equilibrium regions, the distribution might be so different from its corresponding local Maxwellian that the grid defined by (7) and (8) is not sufficient to capture the distribution. Second, we have observed that this method requires much too many interpolations. Indeed, even if u_i^{n+1} and T_i^{n+1} are very close to u_i^n and T_i^n , \mathcal{V}_i^{n+1} and \mathcal{V}_i^n will be slightly different, and interpolations will be necessary for every discrete velocities. This induces a high sensitivity to the interpolation errors, and requires a high order interpolation procedure. Moreover, in 2D, this also induces a very high computational cost, that makes the method not competitive with respect to a standard approach.

Instead, we propose the two following improvements. To avoid systematic interpolations, we define the step of \mathcal{V}_i^{n+1} by using the following threshold: if Δv_i^{n+1} (defined by (8)) is too close to Δv_i^n , then it is modified to $\Delta v_i^{n+1} := \Delta v_i^n$, otherwise Δv_i^{n+1} is replaced by its closest value of the form $2^k \Delta v_i^n$ (where the integer k can be positive or negative). The bounds of \mathcal{V}_i^{n+1} are modified accordingly (see Fig. 1). This procedure results in the following properties: most

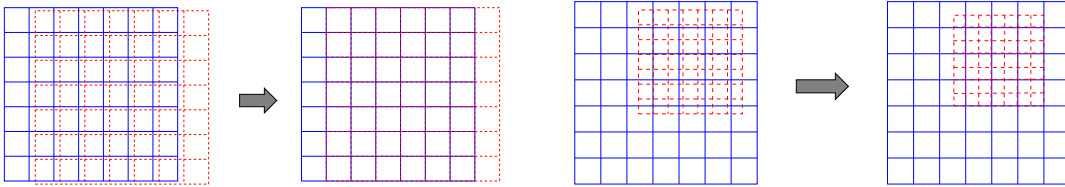


FIGURE 1. Modification of \mathcal{V}_i^{n+1} to avoid too many interpolations

of the points of \mathcal{V}_i^{n+1} are either points of \mathcal{V}_i^n or are outside it (and no interpolation is required), or they are centers of cells of \mathcal{V}_i^n (and the interpolation is easy). It dramatically reduces the number of interpolations required by the scheme.

To take into account non Maxwellian distributions, we also propose the following merge and reduce procedure: first, \mathcal{V}_i^{n+1} is increased by merging it to the neighboring grids of the previous time step:

$$\mathcal{V}_i^{n+1} := \text{Merge}(\mathcal{V}_i^{n+1}, \mathcal{V}_i^n, \mathcal{V}_{i-1}^n, \mathcal{V}_{i+1}^n). \quad (10)$$

Then, to avoid inflation of the grids, \mathcal{V}_i^{n+1} is reduced to its essential part by coarsening and truncation based on the values of f_i^{n+1} (see Fig. 2).

Numerical tests

In all the following tests, we apply the previous method to the BGK equation. This equation is solved with a code which is 2D in space and 2D in velocity. The three-dimensional structure of the real velocity space is taken into account with a standard reduced distribution technique. Our method is compared to a standard discrete velocity method for the BGK equation that uses a global Cartesian velocity grid, and the same numerical scheme for the time and space discretization.

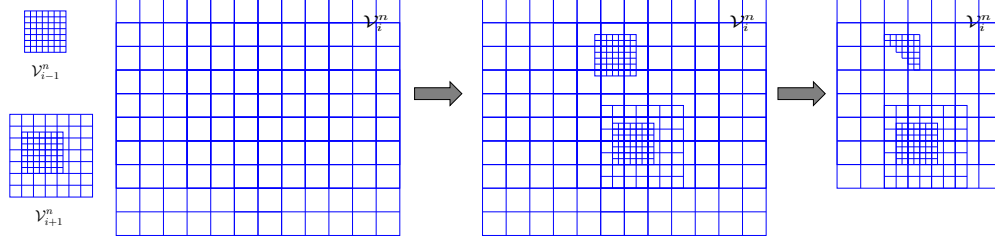


FIGURE 2. Merging, coarsening and truncation of \mathcal{V}_i^{n+1}

1D shock

In this test, a flow of argon is initialized with density $8.876 \cdot 10^{-8} \text{ kg.m}^{-3}$, velocity $-4164.86 \text{ m.s}^{-1}$ directed to a solid boundary located at $x = 0$, and temperature 124.95 K . The solid wall interaction generates a strong shock wave that propagates rightward, until it leaves the computational domain $[0, 1]$. At the solid wall, the molecules are reflected according to the diffuse reflection.

In Fig. 3, we show the contours of the distribution function in the velocity plane (v_x, v_y) and corresponding velocity grids in the rightmost cell space, at four different times $2 \cdot 10^{-5}$, $2.4 \cdot 10^{-4}$, $1.16 \cdot 10^{-3}$, and $3.96 \cdot 10^{-3} \text{ s}$. For this problem, the global grid requires 200 points in each direction to show converged results (with bounds $v_{x,max} = 6.000$, $v_{x,min} = -8.000$, $v_{y,max} = -v_{y,min} = 6.000 \text{ m.s}^{-1}$)! The figure clearly shows the dramatic reduction of the number of discrete velocities with the local grid approach: at initial time, the distribution is the upstream Maxwellian, which requires a small grid localize around the upstream velocity. As time increases, the grid is enlarged to contain the zero velocity, which is induced by the propagation into the domain of the half Maxwellian generated by the solid-wall reflection. In the third plot, the temperature has increased a lot, and the local grid is enlarged to correctly capture the distribution of very energetic molecules. This distribution is not visible yet (the distribution of molecules of the upstream flow is still there), but the local refinement shows that it is already taken into account. The last plot shows the large distribution and the corresponding larger grid.

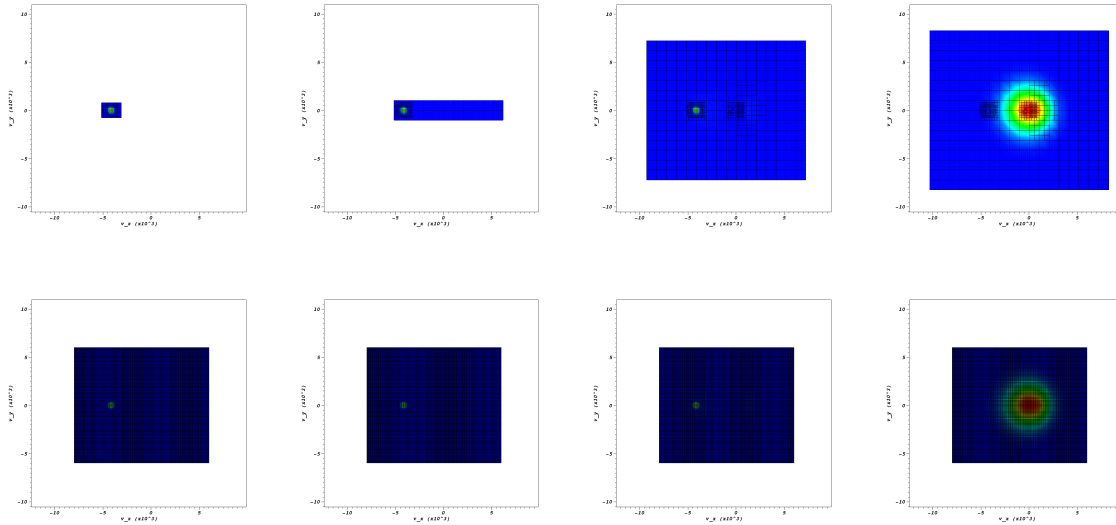


FIGURE 3. One dimensional reflecting shock wave: contours of the distribution function in the velocity plane (v_x, v_y) and corresponding velocity grids in the rightmost cell space, at four different times: (Top) with the local grids, (Bottom) with the global grid.

The accuracy of our results have been checked by a comparison of the macroscopic quantities obtained with the

global grid and local grids methods We have found a relative difference of less than 1% for the mass density, but around 5 to 7% for the velocity, temperature, and pressure. The maximum error appears at the right boundary of the domain, which suggests a problem in the treatment of the outflow boundary condition.

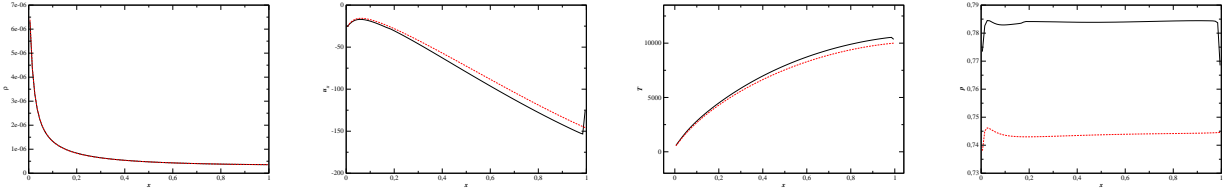


FIGURE 4. One dimensional reflecting shock wave: profiles of density (top left), velocity (top right), temperature (bottom left), and pressure (bottom right) obtain with the local grids (solid) and the global grid (dashed).

High speed Couette flow

Now, we present the results obtained for a 1D Couette flow. In this test, a gas is enclosed between two flat walls. The right wall moves upward with a given velocity, while the left one is fixed. A diffuse reflection is used on both walls. Since the local velocity grid approach is interesting for strong velocity and temperature gradients, we impose a very large velocity of 3.000 m.s^{-1} for the right wall. The initial density is $9.28 \cdot 10^{-6} \text{ kg.m}^{-3}$, and the initial temperature is 273 K, like the temperature of the walls.

In Fig. 5, we show the distribution functions and the corresponding grids (local and global) in the cell adjacent to the left solid (non moving) wall, for four different times. Like in the previous test case, a small grid is used at the initial time, which is sufficient to capture the initial distribution (a Maxwellian of zero velocity). As the time increases, the grid is enlarged in the $v_y > 0$ domain, to take into account the influence of the distribution of the molecules coming from the right part that were reflected by the moving wall. The grid is also refined around the vertical velocity of the right wall, even if the corresponding half wall Maxwellian is not visible here. The distributions obtained by the two methods look very similar, while the number of points required by the local velocity grid approach is much smaller.

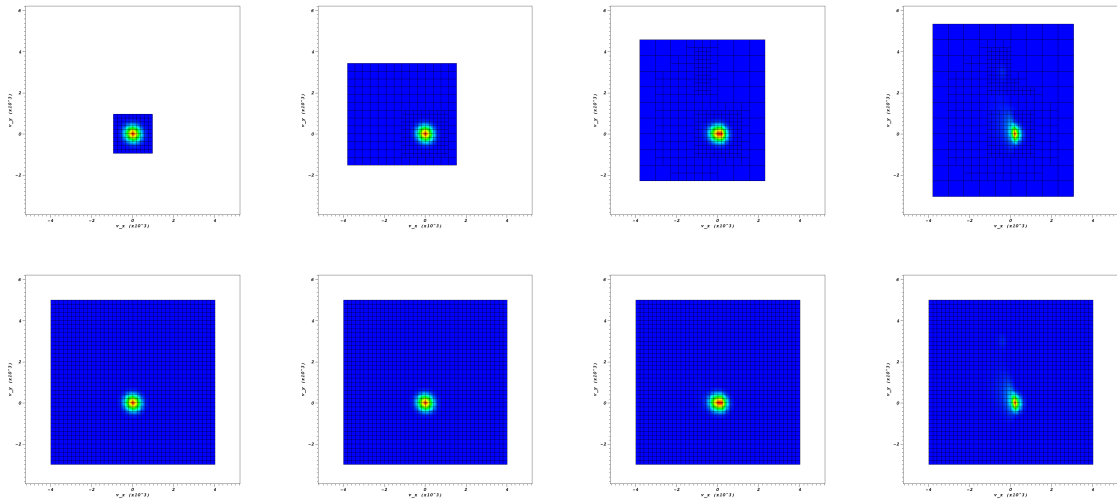


FIGURE 5. High speed Couette flow: contours of the distribution function in the velocity plane (v_x, v_y) and corresponding velocity grids at the left solid wall, at four different times $0, 10^{-4}, 3 \cdot 10^{-4}$ and 10^{-3} s: (Top) with the local grids, (Bottom) with the global grid.

In Fig. 6, we show the distributions and the corresponding grids at the final time at three different positions in

space (at the left solid wall, in the middle of the domain, at the right solid wall). Here, we see that all the local grids have approximately the same bounds, while they are refined differently.

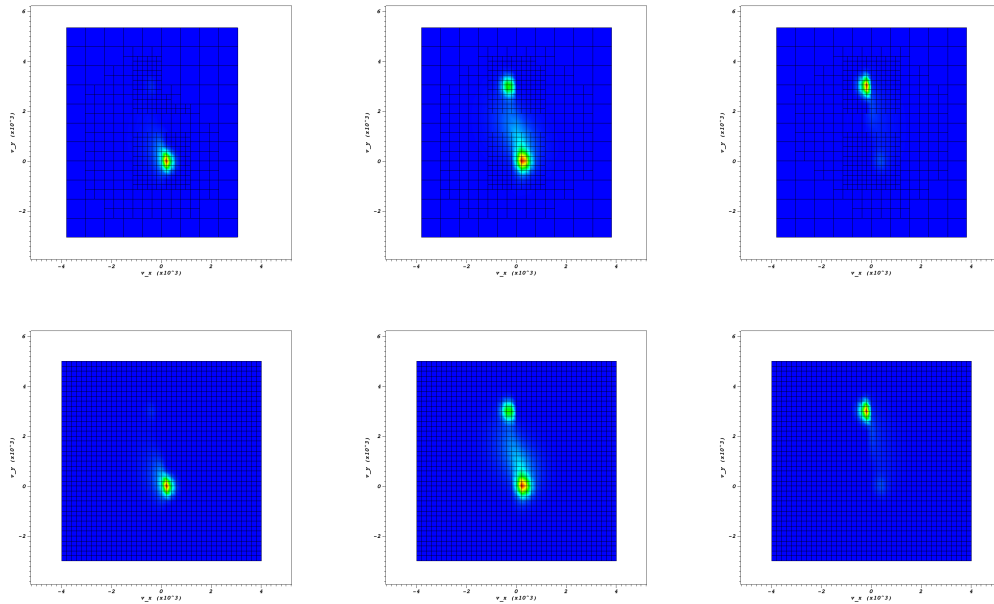


FIGURE 6. High speed Couette flow: contours of the distribution function in the velocity plane (v_x, v_y) and corresponding velocity grids at the final time $t = 10^{-3}$ s, at three different position in space (at the left solid wall, in the middle of the domain, at the right solid wall): (Top) with the local grids, (Bottom) with the global grid.

Again, the accuracy of our results have been checked by a comparison of the macroscopic quantities obtained with the global grid and local grids methods: the differences between the two methods are less that 2%.

Hypersonic flow around a cylinder

This test is taken from [6]: at $t = 0$, a hypersonic flow of argon (Mach number 20, density $3.17 \cdot 10^{-6}$ kg.m⁻³, temperature 242.4 K) impacts a cylinder. The strong interaction between the upstream flow and the solid wall induces a very large increase of the temperature, and later, a reflected bow shock.

This problem requires a global velocity grid which is very large (71 and 55 velocities in x and y directions, respectively) to capture the narrow distributions of the upstream flow and of the wall, as well as the high temperature distribution of the shock wave. At the contrary, the local velocity grid method allows to use much smaller grids, as shown in the following figures.

In Fig. 7, we show the distributions and their corresponding grids obtained in a cell adjacent to the solid wall, for four different times. At $t = 0$, the distribution is the upstream Maxwellian, centered around the horizontal upstream velocity. The corresponding grid is very small. After a very short time (actually, immediately after the first time iteration), the grid is enlarged around the zero velocity, so as to take into account the molecules reflected by the solid wall, normally distributed according to the half Maxwellian of the wall. However, this distribution is not visible. At the third plotting time, the increase of temperature leads to a larger grid, and one can observe that the grid has been refined so as to capture the wall Maxwellian. At the final time of the simulation, the grid is even larger, and the wall half Maxwellian is now visible, while the upstream distribution can still be seen. The second row shows the same results obtained with the global grid: it is clear that the grid is very large and dense, while the contours of the distributions look very similar to the ones obtained with the local grid algorithm.

In Fig. 8, we show the same comparison for a cell adjacent to the upstream boundary. Here, the distribution remains very close to the upstream Maxwellian along the whole simulation. Consequently, the local grid remains a very small one, while the global grid is clearly oversized here.

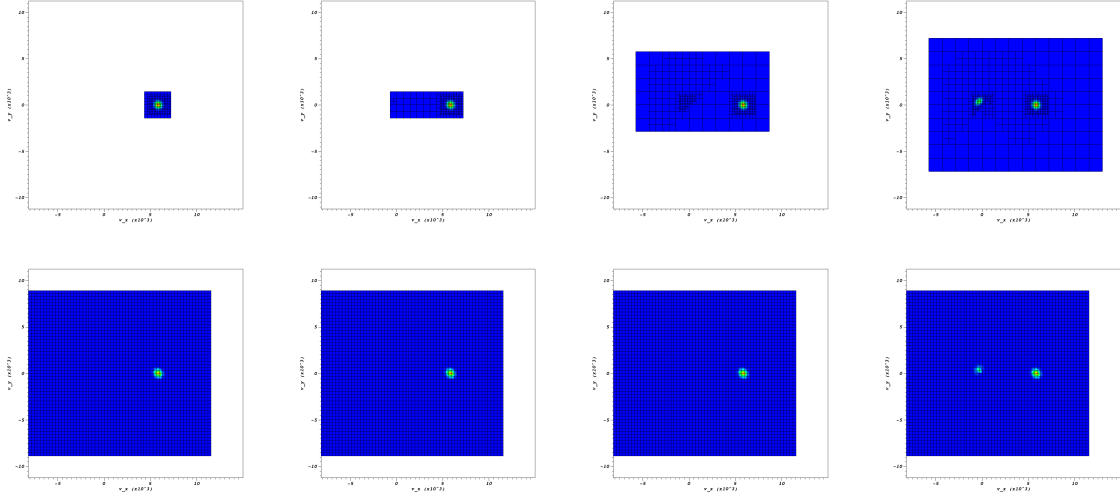


FIGURE 7. Hypersonic plane flow around a cylinder: contours of the distribution function in the velocity plane (v_x, v_y) and corresponding velocity grids in a cell close to the solid wall at four different times ($3.6 \cdot 10^{-10}$, $7.2 \cdot 10^{-10}$, $7.2 \cdot 10^{-9}$, and $1.88 \cdot 10^{-8}$ s): (Top) with the local grids, (Bottom) with the global grid.

The accuracy of our computation with the local grid algorithm can be checked by looking at some macroscopic profiles. For instance, in Fig. 9, we show the temperature and pressure along a line orthogonal to the wall, at 45° : the results given by our method are clearly very close to those obtained with the standard approach.

With this test case, we have also tried to compare the CPU time and memory storage needed for our method as compared to the global grid method. We did this comparison with 8 cores of a Nehalem Intel Xeon X5570 processor, by using a parallel implementation with openMP. We have found that the total number of velocities used by the local grid method is 30 to 20 times smaller (this number depends on the time step), which gives a very large gain in memory storage. However, the CPU time is about the same for both methods (about 2 minutes). The relatively poor performance of the new method in terms of CPU time is discussed in the conclusion below.

Conclusion

In this paper, we have proposed a first attempt to discretize the Boltzmann equation with local velocity grids. In this approach, at each time step and each space cell, the distribution function is discretized on its own local velocity grid. This grid is defined by using the local conservation laws and the grid of the neighboring cells at the previous time step. These local velocity grids naturally adapt in time and space to the variation of the flow and might provide an optimal representation of each distribution (in term of degrees of freedom and memory storage). It is very promising for flows with very large temperature and velocity gradients, like hypersonic reentry flows.

Of course, this method has to be improved to be fully competitive. First, it has to be extended to higher order scheme in time and space, but this should be quite straightforward. More important is the implementation of the method. The computations made by one iteration of our method on a single local velocity grid are much more expensive than those required by a global velocity grid: indeed, our method requires quadrature, merging, interpolation, truncation, coarsening, on a non Cartesian velocity grid. This additional cost might be compensated by the dramatic reduction of the number of velocities, but for an optimal computational gain, we clearly need to optimize the implementation of our approach.

ACKNOWLEDGMENTS

This study has been carried out in the frame of “the Investments for the future” Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02). Experiments presented in this paper were carried out using the PLAFRIM ex-

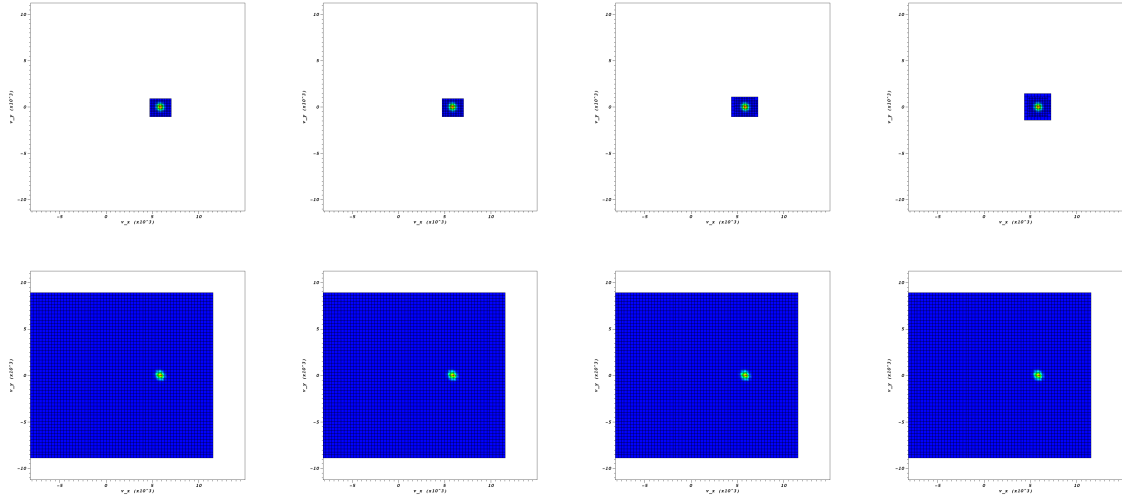


FIGURE 8. Hypersonic plane flow around a cylinder: contours of the distribution function in the velocity plane (v_x, v_y) and corresponding velocity grids in a cell close to the upstream flow at four different times: (Top) with the local grids, (Bottom) with the global grid.

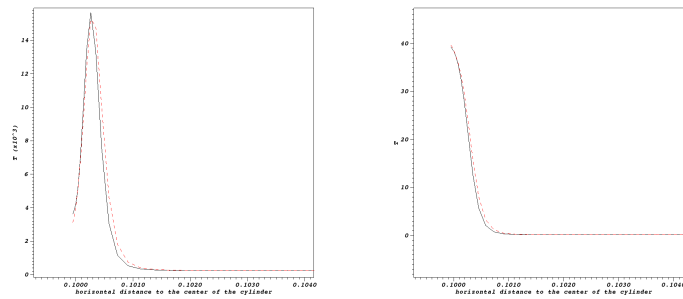


FIGURE 9. Hypersonic plane flow around a cylinder: profiles of temperature (left) and pressure (right) along a line at 45° orthogonal to the wall, obtained with the local grids (solid) and the global grid (dashed).

perimental testbed, being developed under the Inria PlaFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d’Aquitaine, FeDER, Université de Bordeaux and CNRS (see <https://plafrim.bordeaux.inria.fr/>). Computer time for this study was also provided by the computing facilities MCI (Mesocentre de Calcul Intensif Aquitain) of the Université de Bordeaux and of the Université de Pau et des Pays de l’Adour.

REFERENCES

- [1] L. Mieussens, AIP Conference Proceedings **1628**, 943–951 (2014).
- [2] V. I. Kolobov, R. R. Arslanbekov, and A. A. Frolova, AIP Conference Proceedings **1333**, 928–933 (2011).
- [3] S. Chen, K. Xu, C. Lee, and Q. Cai, Journal of Computational Physics **231**, 6643 – 6664 (2012).
- [4] F. Bernard, A. Iollo, and G. Puppo, Communications in Computational Physics **16**, 956–982 (2014).
- [5] S. Brull and L. Mieussens, Journal of Computational Physics **266**, 22 – 46 (2014).
- [6] C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens, Journal of Computational Physics **257**, Part A, 572 – 593 (2014).