# Direct solution of the Boltzmann equation for a binary mixture on GPUs

Aldo Frezzotti, Gian Pietro Ghiroldi and Livio Gibelli

*Dipartimento di Matematica del Politecnico di Milano - Piazza Leonardo da Vinci 32 - 20133 Milan - Italy*

**Abstract.** We show how to accelerate the numerical solution of the Boltzmann equation for a binary gas mixture by using Graphics Processing Units (GPUs). In order to fully exploit the computational power of the GPU, we adopt a semi-regular method of solution which combines a finite difference discretization of the free-streaming term with a Monte Carlo evaluation of the collision integral. The efficiency of the code is demonstrated by studying the propagation of plane harmonic waves of small amplitude in a binary gas mixture of hard spheres for a wide range of Knudsen numbers and wave frequencies. The GPU-based code is about two order of magnitudes faster than the CPU version thus proving that GPUs can substantially speedup the numerical solution of kinetic equations.

**Keywords:** kinetic theory, Monte Carlo methods, mixtures, parallel computing
**PACS:** 51.10.+y,51.40.+p

## INTRODUCTION

Graphics Processing Units (GPUs), originally developed to execute complex computer graphics applications, have been increasingly used in a wide range of scientific and engineering computational applications. The attracting feature of GPUs is their capability of delivering hundreds gigaflops peak performance at the price of conventional workstations. However, only certain classes of algorithms can achieve the maximum theoretical throughput and, even in the most favorable cases, algorithms have to be revised, if not redesigned. Numerical schemes for the solution of kinetic equations are not an exception. For instance, GPUs single instruction multiple data (SIMD) architecture does not allow an easy porting of DSMC schemes in their traditional form [1]. On the contrary, regular and semi-regular methods of solution of kinetic equations are very well adapted to GPUs structure [2, 3, 4].
The main aim of the present work is to show that the simulation of acoustic problems in a rarefied gas environment can greatly benefit from GPU acceleration. The sound wave propagation in a binary gas mixture is a classical subject of investigation in kinetic theory and its applications range from the design of acoustical gas sensors [5] to the analysis of planetary atmospheres [6]. This work differs from previous studies on the same topic in two respects. First, the binary mixture is described by a coupled system of fully nonlinear hard spheres Boltzmann equations. We do neither resort to their linearized form [5, 7] nor replace the collision integrals with simplified collision models [8, 9]. Second, the kinetic equations are solved by a semi-regular method. Unlike particle schemes [10], this method of solution does not require time averaging to provide smooth macroscopic fields and it is better adapted to the parallel architecture of GPUs, as shown in the next sections.

## MATHEMATICAL FORMULATION

We consider a binary mixture of dilute hard sphere gases confined between two flat, infinite and parallel plates located at $x = 0$ and $x = L$, respectively. The atomic species have masses $m_\alpha$ and diameters $d_\alpha$, $\alpha = 1,2$. The state of the mixture is described by two velocity distribution functions $f_\alpha(x,\mathbf{v},t)$ which, in absence of external forces, satisfy the following set of spatially one-dimensional Boltzmann equations:

$$\frac{\partial f_\alpha}{\partial t} + v_x \frac{\partial f_\alpha}{\partial x} = \sum_{\beta=1}^{2} Q_{\alpha\beta}, \qquad \alpha = 1,2 \tag{1}$$

where $Q_{\alpha\beta}(x,\mathbf{v},t)$ denote the nonlinear collision integrals

$$Q_{\alpha\beta}(x,\mathbf{v},t) = \frac{1}{2}\left(\frac{d_\alpha + d_\beta}{2}\right)^2 \int \left[f_\beta(x,\mathbf{v}_1^*,t) f_\alpha(x,\mathbf{v}^*,t) - f_\beta(x,\mathbf{v}_1,t) f_\alpha(x,\mathbf{v},t)\right] |\mathbf{v}_r \cdot \hat{\mathbf{k}}| d\mathbf{v}_1 d^2\hat{\mathbf{k}}. \tag{2}$$

In Eq. (2), $\mathbf{v}_r = \mathbf{v}_1 - \mathbf{v}$ is the relative velocity between two colliding atoms and integration extends over the whole velocity space whereas the solid angle integration extends over the unit sphere, whose points are associated with the unit vector $\hat{\mathbf{k}}$. The pre-collisional velocities, $(\mathbf{v}^*, \mathbf{v}_1^*)$, are obtained from the post-collision velocities, $(\mathbf{v}, \mathbf{v}_1)$, and the unit vector on the sphere, $\hat{\mathbf{k}}$, by the relationships

$$\mathbf{v}^* = \mathbf{v} + \frac{2m_\beta}{m_\alpha + m_\beta}\left(\mathbf{v}_r \cdot \hat{\mathbf{k}}\right)\hat{\mathbf{k}}, \qquad \mathbf{v}_1^* = \mathbf{v}_1 - \frac{2m_\alpha}{m_\alpha + m_\beta}\left(\mathbf{v}_r \cdot \hat{\mathbf{k}}\right)\hat{\mathbf{k}}. \tag{3}$$

Initially, the mixture is supposed to be at rest and in uniform equilibrium with densities $n_\alpha = n_{\alpha 0}$ and temperature $T_0$. Maxwell's completely diffuse boundary condition holds at each plate surface. Accordingly, the distribution functions of atoms re-emitted from the plates are given by the following expressions:

$$f_\alpha(\mathbf{v},t) = \frac{n_{\alpha,w}(t)}{(2\pi R T_0)^{3/2}} \exp\left\{-\frac{[\mathbf{v} - \mathbf{V}_w(t)]^2}{2RT_0}\right\}, \quad \mathbf{v}\cdot\hat{\mathbf{n}}_w > 0, \tag{4}$$

where $\hat{\mathbf{n}}_w$ is the unit vector normal to the plate directed towards the gas and $\mathbf{V}_w$ is the velocity of the plate. In Eq. (4) $n_{\alpha,w}$ is determined by requiring that

$$n_{\alpha,w}(t) = -\left(\frac{2\pi}{RT_0}\right)^{1/2} \int_{\mathbf{v}\cdot\hat{\mathbf{n}}_w < 0} [\mathbf{v} - \mathbf{V}_w(t)] \cdot \hat{\mathbf{n}}_w f_\alpha(\mathbf{v},t) d\mathbf{v} \tag{5}$$

The plate at $x = L$ is assumed to be fixed whereas the plate at $x = 0$ is supposed to oscillate harmonically with velocity $\mathbf{V}_w(t) = V_0 \sin(\omega t)\hat{\mathbf{e}}_x$, being $\hat{\mathbf{e}}_x$ a unit vector normal to the plates. The velocity amplitude $V_0$ and plate oscillation frequency $\omega$ are two specified problem parameters. It should be noted that, following previous numerical studies [7], the emitter plate displacement is not taken into account. The gas feels the plate motion only through the boundary condition (4). If the velocity amplitude of the vibrating plate, $V_0$, is sufficiently small, nonlinear effects are negligible and the solution of the problem is expected to be a standing wave given by the superposition of two propagating harmonic plane waves generated by the oscillating plate and by reflection at the stationary plate [10]. Accordingly, the expression of the hydrodynamic velocity profile sufficiently far away from the plates is expected to have the following form

$$V(x,t) = A(x)\cos(\omega t) + B(x)\sin(\omega t), \tag{6}$$

where the functions $A(x)$ and $B(x)$ are given by

$$A(x) = V_0\left[e^{-m(2L-x)} - e^{-mx}\right]\sin(kx), \tag{7}$$

$$B(x) = V_0\left[e^{-mx} + e^{-m(2L-x)}\right]\cos(kx). \tag{8}$$

In Eqs. (7) and (8), $k$ is the wave number and $m$ is the attenuation coefficient. Both $k$ and $m$ depend on the wave frequency $\omega$ through the real and imaginary part of the dispersion relation, respectively [5].

## NUMERICAL METHOD

The system (1) is solved numerically by a method based on the discretization of the distribution functions on a phase space grid. In the spatially one-dimensional geometry considered here, the region between the plates is divided into $N_x$ cells of the same size $\Delta x = L/N_x$. A grid in the velocity space is constructed for each species by replacing the whole infinite velocity space with finite regions $\mathscr{V}_\alpha$. Advantage is taken of the symmetry of the $f_\alpha$ which depend on the velocity through the arguments $v_\parallel = v_x$ and $v_\perp = (v_y^2 + v_z^2)^{1/2}$. Accordingly, $\mathscr{V}_\alpha$ are cylinders in which a regular net of nodes, $N_\parallel \times N_\perp$, is arranged. The cylinders size is large enough to contain the significant part of the distribution

functions at any spatial location. A separate grid for each species is used. The grid associated with the second species is obtained from the grid of the first species by multiplying the coordinates of each node by the factor $(m_1/m_2)^{1/2}$. The distribution functions are assumed to be constant within each cell of the phase space. Hence, $f_\alpha$ are represented by the arrays $f_{\alpha,i\mathbf{j}}(t) = f_\alpha(x(i), v_\parallel(j_\parallel), v_\perp(j_\perp), t)$, being $x(i), v_\parallel(j_\parallel), v_\perp(j_\perp)$ the values of the spatial coordinate and velocity components in the center of the phase space cell $(i,\mathbf{j})$ with $\mathbf{j} = (j_\parallel, j_\perp)$.
The numerical algorithm that advances $f_\alpha$ from time $t_n$ to the next time level $t_{n+1} = t_n + \Delta t$ is constructed by time-splitting the evolution operator into a free streaming step, in which the right hand side of Eq. (1) is neglected, and a purely collisional step, in which spatial motion is frozen and only the effect of the collision integrals are taken into account. More precisely, intermediate values of the distribution functions are computed from the free-streaming equations according to the first order explicit upwind scheme

$$f^{n+1/2}_{\alpha,i\mathbf{j}} = (1-C_{\mathrm{F}}) f^{n}_{\alpha,i-\mathrm{sign}(v_x)C_{\mathrm{I}}\mathbf{j}} + C_{\mathrm{F}} f^{n}_{\alpha,i-\mathrm{sign}(v_x)(C_{\mathrm{I}}+1)\mathbf{j}} \tag{9}$$

where $C_{\mathrm{I}}$ and $C_{\mathrm{F}}$ are the integer and fractional part of the absolute value of the Courant number, $C = C_{\mathrm{I}} + C_{\mathrm{F}} = |v_x \Delta t|/\Delta x$. After completing the free flight step, the homogeneous relaxation step takes place in each cell of the spatial grid according to the implicit scheme

$$\tilde{f}^{n+1}_{\alpha,i\mathbf{j}} = f^{n+1/2}_{\alpha,i\mathbf{j}} + \left[ Q^{n+1/2}_{\alpha,i\mathbf{j}} - \nu^{n+1/2}_{\alpha,i\mathbf{j}} \left( \tilde{f}^{n+1}_{\alpha,i\mathbf{j}} - f^{n+1/2}_{\alpha,i\mathbf{j}} \right) \right] \Delta t, \qquad \alpha = 1,2 \tag{10}$$

where

$$Q^{n+1/2}_{\alpha,i\mathbf{j}} = \frac{Q^{n+1/2}_{\alpha\alpha,i\mathbf{j}} + Q^{n+1/2}_{\alpha\beta,i\mathbf{j}}}{\Delta \mathscr{V}_{\alpha,\mathbf{j}}}, \qquad \beta \neq \alpha$$

$$Q^{n+1/2}_{\alpha\alpha,i\mathbf{j}} = \frac{d_\alpha^2}{8} \int \left[\chi(\mathbf{v}) + \chi(\mathbf{v}_1) - \chi(\mathbf{v}^*) - \chi(\mathbf{v}_1^*)\right] \left[ f^{n+1/2}_{\alpha,i}(\mathbf{v}_1^*) f^{n+1/2}_{\alpha,i}(\mathbf{v}^*) - f^{n+1/2}_{\alpha,i}(\mathbf{v}_1) f^{n+1/2}_{\alpha,i}(\mathbf{v}) \right] |\mathbf{v}_r \cdot \hat{\mathbf{k}}| d\mathbf{v} d\mathbf{v}_1 d\hat{\mathbf{k}}, \tag{11}$$

$$Q^{n+1/2}_{\alpha\beta,i\mathbf{j}} = \frac{1}{4} \left( \frac{d_\alpha + d_\beta}{2} \right)^2 \int \left[\chi(\mathbf{v}) - \chi(\mathbf{v}^*)\right] \left[ f^{n+1/2}_{\beta,i}(\mathbf{v}_1^*) f^{n+1/2}_{\alpha,i}(\mathbf{v}^*) - f^{n+1/2}_{\beta,i}(\mathbf{v}_1) f^{n+1/2}_{\alpha,i}(\mathbf{v}) \right] |\mathbf{v}_r \cdot \hat{\mathbf{k}}| d\mathbf{v} d\mathbf{v}_1 d\hat{\mathbf{k}}, \tag{12}$$

being $\chi$ the characteristic function of the velocity cell $\mathscr{C}_{\alpha,\mathbf{j}}$ and

$$\nu^{n+1/2}_{\alpha,i\mathbf{j}} = \frac{1}{2\Delta \mathscr{V}_{\alpha,\mathbf{j}}} \int_{\mathscr{C}_{\alpha,\mathbf{j}}} d\mathbf{v} \sum_{\beta=1}^{2} \left( \frac{d_\alpha + d_\beta}{2} \right)^2 \int f^{n+1/2}_{\beta}(\mathbf{v}_1) |\mathbf{v}_r \cdot \hat{\mathbf{k}}| d\mathbf{v}_1 d\hat{\mathbf{k}}. \tag{13}$$

In Eq. (10), the implicit treatment of the loss term prevents the distribution functions from becoming negative during the calculation. The eight-fold integrals given by Eqs. (11)-(13) are calculated by a Monte Carlo quadrature method, since a regular quadrature formula would be too demanding in term of computing time. A drawback of the technique is that, owing to the discretization in the velocity space, mass, momentum and energy are not exactly conserved. The numerical error is usually small but tends to accumulate during the time evolution of the distribution function. The correction procedure proposed in Ref. [11] has been adopted to overcome this difficulty. At each time step the distribution functions are corrected in the following way

$$f^{n+1}_{\alpha} = \tilde{f}^{n+1}_{\alpha} \left( 1 + A_\alpha + \mathbf{B}_\alpha \cdot \mathbf{v} + C_\alpha \mathbf{v}^2 \right), \tag{14}$$

where the constants $A_\alpha, \mathbf{B}_\alpha$ and $C_\alpha$ are determined from the conditions

$$\int \psi(\mathbf{v}) f^{n+1}_{\alpha}(\mathbf{v}) d\mathbf{v} = \int \psi(\mathbf{v}) f^{n+1/2}_{\alpha}(\mathbf{v}) d\mathbf{v} + \frac{1}{2} \Delta t \int \left[\psi(\mathbf{v}) - \psi(\mathbf{v}^*)\right] Q^{n+1/2}_{\alpha} d\mathbf{v}, \tag{15}$$

being $\psi(\mathbf{v}) = 1, \mathbf{v}, \mathbf{v}^2$. The eight-fold integrals appearing in the second term at the right hand side of Eq. (15) are evaluated while calculating the collision integrals themselves, therefore the increase in computing time is small.

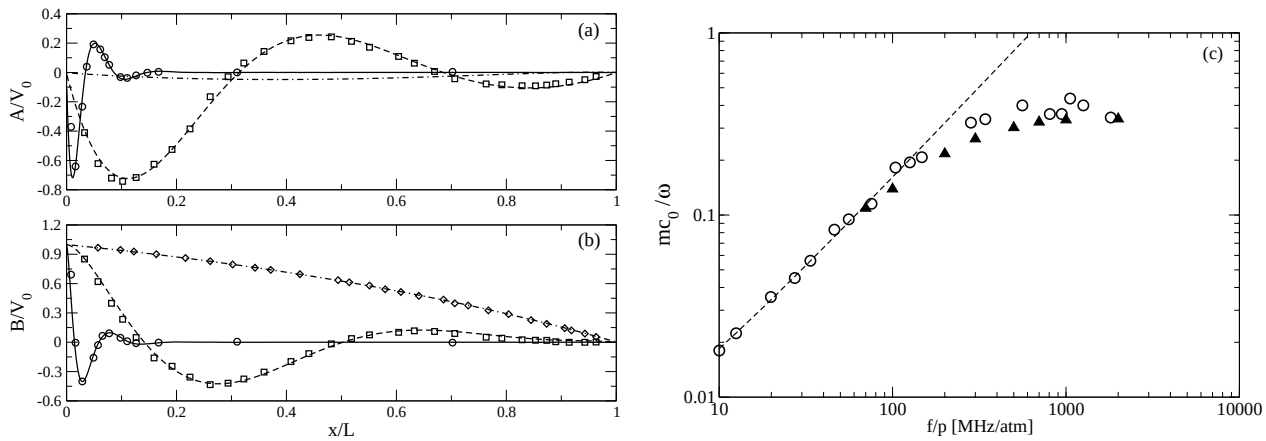


**FIGURE 1.** **(a)-(b)** Cosine component $A(x)$, (a), and sine component $B(x)$, (b), of the hydrodynamic velocity, Eq. (6), for $\omega t_{\rm ref} = 0.12$ (dashed-dotted line), $\omega t_{\rm ref} = 1.25$ (dashed line) and $\omega t_{\rm ref} = 12.56$ (solid line). Symbols are the numerical results reported in Ref. [9]. $\mathrm{Kn} = l_{\rm ref}/L = 0.09$. **(c)** Dimensionless attenuation coefficient, $mc_0/\omega$, versus the scaled frequency, $f/p$, in a Argon-Helium mixture. Triangles: numerical solution of Eq. (1). Circles: experimental data reported in Ref. [12]. Dashed line: theoretical prediction of the Navier-Stokes equations. $n_{20}/n_0 = 0.25$.

## CUDA™ IMPLEMENTATION

The computations described below have been performed on an NVIDIA®GPU consisting of a set of multiprocessor with a SIMD-like architecture. During each clock cycle, each core of the multiprocessor executes the same instruction but operates on different data. CUDA™ is the high level programming language specifically created for developing applications on this platform. A CUDA™ program is organized into a serial program which runs on the host CPU and one or more kernels which define the computation to be concurrently performed. Kernels are executed by threads, which are organized into a two-level grid and block hierarchy. One may think of a grid as the GPU itself, a block as multi-processor of the GPU and a thread as a processor core in the multi-processor. GPU memory has a hierarchical structure whose understanding is of fundamental importance for code optimization. Actually, data transfer is the rate limiting step in most applications. Each thread may access private, low latency memory registers. Threads belonging to the same block are allowed to synchronize with each other and are allowed to share data through a shared memory which is as fast as registers. However, threads from different blocks may coordinate only via operations in the slower global memory. The code for the numerical solution of Eqs. (1) is a straightforward extension of the one presented in Refs. [2, 4] to solve the single component hard sphere Boltzmann equation. The code is organized into a host program, which deals with all memory management and other setup tasks, and three kernels running on the GPU. The first kernel performs the streaming step whereas the second and third kernels perform the collision step. For each cell of the velocity space, the streaming step, Eq. (9), involves the distribution function at different space locations. The key performance enhancing strategy is to allow threads to cooperate in the shared memory. Blocks are composed by a one dimensional grid of threads, where each thread is associated with one cell of the physical space. As mentioned above, the collision step is organized into two kernels. The first kernel computes the sequence of samples needed in the Monte Carlo evaluation of the collision integrals in Eqs. (11)-(13). In this case, a different thread is associated to each sample. The relaxation step, described by Eq. (10), does not involve any information from nearby cells. Hence the most time consuming phase can be concurrently executed on the GPU by associating a thread to each cell of the phase space. In order to reduce data transfers from and to the slow global memory, the computation of macroscopic quantities and the collision step are performed in the same kernel, by having a thread associated to each cell of the physical space. The collision kernel updates the distribution function according to Eq. (10), applies the polynomial correction given by Eq. (14) to ensure conservation of mass, momentum and energy, and finally computes the macroscopic quantities of interest. The results shown below have been obtained on a commercially available GPU GeForce GTX 260 produced by NVIDIA® using CUDA™ version 2.0. Its theoretical peak performance is 715.4 GFLOPs. The graphic processing unit is hosted by a personal computer equipped with 4 GB of main memory and an Intel® Core Duo Quad Q9300 CPU, running at 2.5 GHz. The host machine has been used to run the sequential version of the program to obtain the speed-up data.

## RESULTS AND DISCUSSION

The non-dimensional form of Eq. (1) has been adopted in actual computations by normalizing velocities $\mathbf{v}$ to $v_{\text{ref}} = (k_{\text{B}}T_0/m_1)^{1/2}$, being $k_{\text{B}}$ the Boltzmann constant, and the spatial coordinate $x$ to the mean free path $l_{\text{ref}} = 1/(\sqrt{2}\pi d_1^2 n_0)$, where $n_0 = n_{10} + n_{20}$ and $n_{10}, n_{20}$ are the uniform densities of the binary gas mixture in the physical domain in the initial state. The reference time is then given by $t_{\text{ref}} = l_{\text{ref}}/v_{\text{ref}}$. The dimensionless length of the physical cells has been varied in the range $\Delta x/l_{\text{ref}} \in [0.02, 0.1]$ depending on the oscillation frequency of the plate. In fact, higher spatial resolution is required for large values of $\omega$ because of the shorter wavelength of spatial oscillations. The velocity space grids have been constructed by distributing a variable number of nodes ($N_{\parallel} = 40 - 80$, $N_{\perp} = 20 - 40$) in the domain $(v_{\parallel}, v_{\perp}) \in \mathscr{V}_{\alpha} = [-V_{\alpha}, V_{\alpha}] \times [0, V_{\alpha}]$, with $V_{\alpha} = 4(m_1/m_{\alpha})^{1/2}$. The dimensionless time step has been varied in the range $\Delta t/t_{\text{ref}} \in [0.01, 0.25]$, in order to have a few hundreds time samples per wave period. Finally, the number of samples used in the Monte Carlo evaluation of each collision integral, $N_c$, has been varied between 2000 and 9000 per time step to obtain results almost free from statistical fluctuations.

The GPU-based code has been validated against the numerical solutions presented in Ref. [9] where the sound propagation in a simple monatomic gas has been studied by solving the linearized Shakhov model equation. The dispersion relationships for a simple hard sphere gas have been obtained from the numerical solutions of the Boltzmann equations by following the methodology described in Ref. [10]. After the decay of the initial transient, the velocity field is evaluated at each time step and the functions $A(x)$ and $B(x)$ are obtained via a chi-square fit of the data. As shown in Fig. 1a and 1b, the comparison with the numerical results reported in Ref. [9] is quite good. It is worth noticing that, for the chosen simulation parameters, the linearized Shakhov model equation provides predictions that agree very well with those of the full Boltzmann equation, as expected. In the application to mixtures, the attenuation coefficient for a sound wave propagating in a binary mixture of Argon and Helium has been computed. Argon is associated with species 1, the mass and hard sphere diameter ratios have been set equal to 10 and 1.8, respectively. The overall Helium molar fraction of the mixture has been set equal to 0.25. It has been verified that the length of the domain, $L$, resulted to be at least two times larger than the wave length, $2\pi/k$, for all the investigated frequencies. The approximation given by Eqs. (7) and (8) would not be accurate, for smaller values of $L$. The Nelder-Mead simplex method has been adopted to extract the wave number $k$ and the attenuation coefficient $m$ from $A(x)$ and $B(x)$. In order to take into account boundary effects, a phase shift is also included in the parameter fits at the higher frequency, as suggested in Ref. [10]. In Fig. 1c the numerical results are compared with the experimental data reported in Ref. [12]. The attenuation coefficient $m$ has been made non-dimensional by multiplication by the factor $c_0/\omega$, being $c_0$ the adiabatic sound speed of the mixture. The quantity $mc_0/\omega$ is plotted versus the scaled frequency $f/p$, being $f = \omega/(2\pi)$ the wave frequency and $p$ the pressure of the mixture at the reference state. Good agreement is found for all the investigated frequencies. As expected, the numerical results agree with the theoretical predictions of the Navier-Stokes equations at low frequencies. The performance of the GPU implementation is compared with the single-threaded version running on the CPU by computing the speed-up factor $S = T_{\text{CPU}}/T_{\text{GPU}}$, where $T_{\text{CPU}}$ and $T_{\text{GPU}}$ are the CPU and GPU run times, respectively. Run times are measured after initial setup, and do not include the time required to transfer data between the disjoint CPU and GPU memory spaces. Figure 2a shows the relative time which is spent on the streaming step, $T_s$, (dark bar) and on the collision step, $T_c$, (light bar) as well as the total execution time in seconds, indicated by an integer number over each bar. As expected, the collision step is more time consuming than the streaming step which takes at most 1% of the overall computing time. Figure 2b reports the obtained speed-up data as a function of the number of spatial grid points $N_x$. The speed-up grows rapidly and then it levels up at about 400 if $N_x$ approximately exceeds 3000. This behavior is the result of the parallel set up of the collision step in $N_x$ independent threads. As discussed above, in fact, the collision step absorbs most of the computational resources and its execution strongly affects the overall performances. As shown by the speed-up curve, the GPU power is not fully exploited till the number of concurrent threads reaches a threshold. Beyond, the speed-up saturates and the computing time approximately behaves as a linear function of $N_x$. This behavior closely patterns the one reported in Refs. [2, 3, 4]. It is worth noticing that neither the GPU-based code nor its CPU version have been fully optimized. Therefore only the order of magnitude of the reported speed-up should be considered correct.

## CONCLUSION

In this paper we have exploited the computational power of modern GPUs to simulate the propagation of small amplitude waves in a rarefied binary mixture of monatomic gases composed by hard spheres. A system of coupled

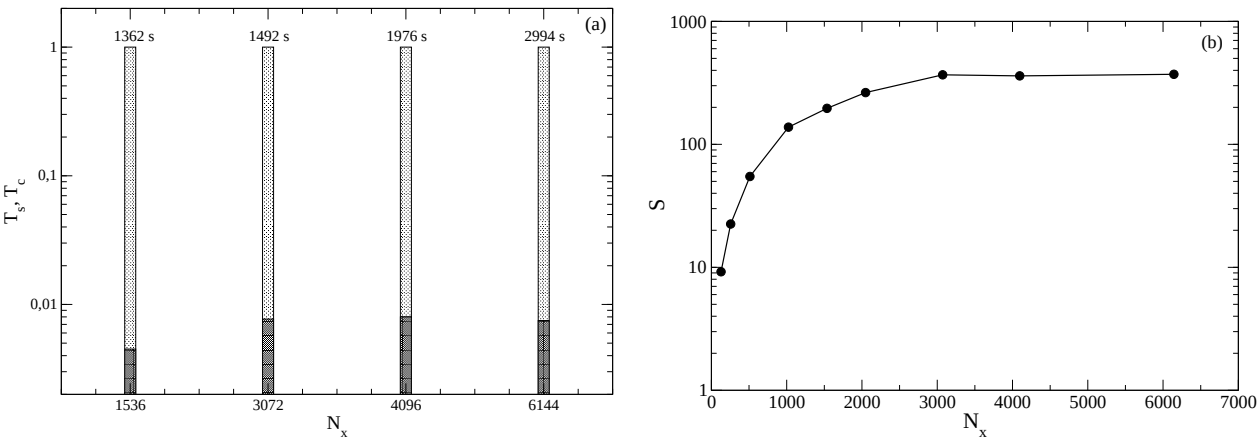


**FIGURE 2.** **(a)** Relative time spent on the streaming kernel (dark bar) and on collision kernel (light bar), versus the number of cells in the physical space, $N_x$. The numbers above the bars refer to the total execution time expressed in seconds. **(b)** Speed-up, $S$, versus number of cells in the physical space, $N_x$. $N_{\|} = 60$, $N_{\perp} = 30$, $N_c = 4096$.

nonlinear Boltzmann equations has been solved by means of a semi-regular method which combines a finite difference discretization of the free-streaming terms with a Monte Carlo evaluation of the collision integrals. This method of solution is ideally suited for SIMD-like parallel architectures provided by commercially available GPUs. For example, obtaining the attenuation coefficient for $f/p = 200 MHz/atm$ with $2048 \times 60 \times 30 \times 2$ cells in the phase space, 6144 collisions per time step and 6240 time steps takes only 25 minutes. Performing a similar calculation on a CPU would have taken a few days. Although we have here emphasized the computational aspects of the problem, this work is also the first step of a systematic investigation of non-equilibrium effects in gas mixture, such as anomalous sound dispersion [13] and ultrasonic-driven gas separation [14]. Further aspects of these phenomena will be studied at length in a future publication.

## ACKNOWLEDGMENTS

The authors gratefully acknowledges the support received from *Fondazione Cariplo* within the framework of the project *Surface Interactions in micro/nano devices*.

## REFERENCES

1. L. Dagum, AIAA J. Thermophys. Heat Transfer **6** (4), 637-642 (1992)
2. A. Frezzotti, G. P. Ghiroldi and L. Gibelli, *"Solving the Boltzmann equation on GPUs"*, Proceedings of Micro and Nano Flows Conference MNF2009, London, 1-7 (2009)
3. A. Frezzotti, G. P. Ghiroldi and L. Gibelli, arxiv:0903.4044v1
4. A. Frezzotti, G. P. Ghiroldi and L. Gibelli, arxiv:1005.5405v1
5. D. G. Napier and B. D. Shizgal, Physica A **387**, 4099-4118 (2008)
6. A. D. Hanford and L. N. Long, J. Acoust. Soc. Am. **125** (2), 640-650 (2009)
7. R. D. M. Garcia and C. E. Siewert, Z. angew. Math. Phys. **57**, 94-122 (2006)
8. A. S. Fernandes and W. Marques Jr., Physica A **332**, 29-46 (2004)
9. D. Kalempa and F. Sharipov, Phys. Fluids **21**, 103601 (2009)
10. N. G. Hadjiconstantinou and A. L. Garcia, Phys. Fluids **13** (4), 1040-1046 (2001)
11. A. Frezzotti, Fluid Dyn. Res. **8** 175-187 (1991)
12. G. J. Prangsma, R. M. Jonkman and J. J. M. Beenakker, Physica **48**, 323-330 (1970)
13. J. R. Bowler and E. A. Johnson, Phys. Rev. Lett. **54** (4), 329-332 (1985)
14. D.R. Rector, M. G. Greenwood, S. Ahmed, S. R. Doctor, G. J. Posakony and V. S. Stenkamp, J. Acoust. Soc. Am. **121** (6), 3446-3452 (2007)