A survey of deterministic solvers for rarefied flows

Luc Mieussens

Institut de Mathématiques de Bordeaux,
Université de Bordeaux,
351, cours de la Libération, 33405 Talence cedex,
France

Abstract. Numerical simulations of rarefied gas flows are generally made with DSMC methods. Up to a recent period, deterministic numerical methods based on a discretization of the Boltzmann equation were restricted to simple problems (1D, linearized flows, or simple geometries, for instance).

In the last decade, several deterministic solvers have been developed in different teams to tackle more complex problems like 2D and 3D flows. Some of them are based on the full Boltzmann equation. Solving this equation numerically is still very challenging, and 3D solvers are still restricted to monoatomic gases, even if recent works have proved it was possible to simulate simple flows for polyatomic gases. Other solvers are based on simpler BGK like models: they allow for much more intensive simulations on 3D flows for realistic geometries, but treating complex gases requires extended BGK models that are still under development.

In this paper, we discuss the main features of these existing solvers, and we focus on their strengths and inefficiencies. We will also review some recent results that show how these solvers can be improved: - higher accuracy (higher order finite volume methods, discontinuous Galerkin approaches) - lower memory and CPU costs with special velocity discretization (adaptive grids, spectral methods) - multi-scale simulations by using hybrid and asymptotic preserving schemes - efficient implementation on high performance computers (parallel computing, hybrid parallelization)

Finally, we propose some perspectives to make these solvers more efficient and more popular.

Keywords: discrete velocity models, deterministic solvers, Boltzmann equation, BGK model
PACS: 47.45.-n,47.11.-j

INTRODUCTION

Deterministic solvers for Rarefied Gas Dynamics (RGD) are solvers in which the Boltzmann equation (or some simplified models) are numerically solved. Most of them are based on a method called "discrete velocity models" or "discrete ordinates" that goes back to the work of Chandrasekhar in 1950 in radiative transfer theory [1]. This method was applied in RGD in the 60's, like for instance by Giddens et al. [2] in 1967, and then applied by several other authors. Soon after, the DSMC method was proposed by Bird [3], and this method rapidly became very popular, while the deterministic solvers, much too computationally expensive for realistic applications, were less and less used. It is interesting to note that this was not the case in linear transport (neutrons, radiative transfer), where deterministic solvers have always been used intensively.

In the past 20 years, there has been a renewed interest in deterministic solvers in RGD, probably due to the strong power increase of the computational resources.

This is why we shall try to make a brief review of these solvers in this paper. Of course, we do not intend to be exhaustive in such a short paper. Our idea is rather to give an idea of the main basic tools used to solve kinetic equations of RGD, and its main recent developments. In particular, we do not discuss the moment based methods [4].

Moreover, we point out that there were several new developments of deterministic solvers presented in the 29th RGD Symposium, but it was not possible to take most of them into account in this review. The reader is invited to look at them in the other papers of this volume.

In the first section, we present the main deterministic methods for solving Boltzmann and BGK equations. Theses methods have recently been greatly improved, and recent developments are reviewed in the second section. Finally, we give in the last section our personal point of view on the perspectives and main challenges for the future deterministic solvers.

DETERMINISTIC SOLVERS

Different applications

First of all, it is important to note that there are different deterministic solvers because there are different applications. Let us now describe these applications.

Many methods were designed to compute accurate solutions of fundamental problems, like Couette or Poiseuille flows, flows around a sphere, discontinuity of the distribution function w.r.t the velocity variable, etc. For these problems, it was generally sufficient to write a 1D code, which made possible to have a huge number of degrees of freedom and hence gave very accurate results. It was also sufficient to write a specific code for every problem, since the corresponding programs were quite short.

This is completely different for aerodynamics. In this field, people are generally interested in 2D and even 3D solutions, for steady flows, and for small Knudsen numbers. The computational requirements are huge, and High Performance Computers are necessary. Fast convergent solvers are required to get the steady solution in a reasonable CPU time. The computational code needs to be versatile, since the corresponding programs are very long (more than several thousand lines): indeed, they cannot be rewritten for any specific problem, and they have to work for various geometries and various kind of flow.

Another kind of applications is the computation of slow flows or micro flows. For fundamental problems, 1D computations were sufficient, but many new problems require 2D and even 3D solutions, with moving obstacles, unsteady flows, and complex geometries. Slow flows are often very well described by linear kinetic equations, that are solved with specific solvers.

Finally, the most difficult problems are probably multiscale flows, like in a turbo-molecular pump for instance. In these flows, the gas passes continuously from the continuous regime to the free molecular regime. Moreover, there can be fast moving and curved boundaries in the flow. Another important constraint is that vacuum pump manufacturers who want to use kinetic solvers to improve the design of their pumps do not have high performance computers, as opposed to research centers of aerospace engineering companies. This requires the development of very specific methods with a reduced computational cost.

Boltzmann solvers

Most of solvers based on the Boltzmann equations use the discretization of the distribution function with respect to the velocity variable, which leads to a system of equations falling in the class of the so-called "Discrete Velocity Model" (DVM). While these models have been studied on their own [5, 6], we only describe here the models used for computational purposes, that is to say to develop deterministic solvers.

The basic idea is to approximate the collision operator on a Cartesian velocity grid as follows: the distribution function is replaced par its values on the grid, and to approximate the collision operator by some quadrature formula. This is done by selecting the collisional velocities in a way such that the conservation properties of the operator are satisfied at the discrete level. It seems that the first of these solvers was proposed by Rogier and Schneider in 1994 [7] and at the same time by Tan and Varghese [8]. Another DVM was proposed more recently by Heintz and Panferov [9] based on the Carleman representation of the collision operator.

The main problem of this approach is first its large computational cost. Indeed, the number of operations required to evaluate the discrete collision operator is at least larger than $O(N^2)$, where N is the number of discrete velocities. This is mainly due to the fact that the collision operator takes into account binary collisions and involves 5 fold integrals. The other drawback is the low accuracy of these solvers, which requires to take N very large. Finally, this approach is difficult to extend to general models of interaction potentials, and is also restricted to uniform Cartesian grids.

This is why several modifications were proposed. The method of Tan and Varghese [8] reduces the cost by using a Monte-Carlo evaluation of the collision integral (this scheme is hence not fully deterministic).

Several deterministic approximations of the Boltzmann collision operator were proposed by the Moscow group (Aristov, Tcheremissine, etc.), see for instance the book by Aristov [10] and the projection method of Tcheremissine in 2005 [11] (also see the work of Popov and Tcheremissine in 1999 [12]). This allowed Kolobov et al. to develop a variant called the NtcN method in 2007 [13]: this method gives a solver more accurate than the DVM mentioned above that works for general models of interaction potentials and can be applied to non uniform velocity grids. However, its computational cost is still more than $O(N^2)$, but it is shown in [13] that this cost can be reduced (by using the so-called

Korobov sequences): this allows the authors to build a 3D solver that is used for aerodynamic simulations. Up to our knowledge, this was the first time that a 3D Boltzmann solver was shown in the literature. However, the computational cost of this method is not given in the paper. We also mention that another 3D simulation (for a Holweck pump) was made with the Tcheremissine's method by Anikin et al. in 2012 [14].

Another approach is based on a finite element method, as proposed by Ohwada in 1993 [15] for the Boltzmann equation with a 2D velocity variable in cylindrical geometry. The accuracy is very good, but the computational cost is still larger than $O(N^2)$ (where N here is the total number of basis functions).

In our opinion, these solvers can give very accurate solutions, and hence be very useful to compute reference solutions, however, their high computational cost makes them hardly usable for intensive real applications. This is why many people use instead the simpler BGK models, as described in the next section.

BGK based solvers

BGK like models

In this model, the collision operator is replaced by a relaxation of the distribution function to its own local Maxwellian. This model has been extended to take into account some important physical properties:

- correct Prandtl number: the ES-model of Holway [16], the S-model of Shakhov [17]
- polyatomic gases: a simple model with two relaxation term was proposed by Holway in the same paper where he derived the ES-model [16], the extension of the ES-model by Andries et al. [18], and the R-model, which is an extension of the S-model proposed by Rykov [19].

The main advantages of these relaxation models are the followings:

- the additional degrees of freedom, like the internal energy for polyatomic gases, or the third component of the velocity for 2D flows, can be eliminated by using the *reduced distribution technique*, as first proposed by Chu in 1965 in [20]. This greatly reduced the complexity of the kinetic equation.
- the computational cost of the velocity discretization of is *linear* with respect to N, which is the best that can be done. This is due to the fact that the relaxation operator only requires to compute the moments of f, and hence 3 fold integrals only, instead of the 5 fold integrals of the Boltzmann equation.

BGK solvers

For computing fundamental solutions, two groups of people made several important contributions: the Kyoto group (Aoki, Ohwada, Sone, Takata, etc.), see references in [21], and the Russian group of Shakhov, Larina and Rykov [22, 23, 24]. Their common tools were finite differences, discrete velocity approximation, Cartesian grids, finite point iteration for steady flows. The Kyoto group developed an important method to capture the discontinuities of the distribution function with respect to the velocity variable [25].

For linearized problems, like in slow flows or flows in vacuum devices, several people made use of some efficient tools of the linear transport community, like Valougeorgis [26] for an acceleration technique and [27] for a fast linear solver, and Varoutis et al. [28] for an integral method.

For 2D and 3D steady flows in aerodynamics, it was necessary to develop implicit solvers: indeed, these solvers use a pseudo-transient method that allows to take very large time step, and hence to converge to the stationary solution much faster than standard explicit methods or fixed point techniques. This was first done by Yang and Huang in 1996 [29] with a linear semi-implicit solver. This was extended to unstructured grids by Titarev in 2009 [30]. A linearized implicit scheme was proposed by Mieussens in 2000 for curvilinear structured grids [31, 32], and this was extended to unstructured grids by Brook [33]. The common tools of these solvers are standard tools from Computational Fluid Dynamics (CFD), like block structured grids, unstructured grids, finite volume schemes, linear solvers for large and sparse linear systems, etc. Up to our knowledge, [29] was the first to introduce these efficient tools in RGD. These different solvers were applied to 2D and 3D aerodynamical flows with various geometries, and were shown to be also efficient for slow flows (see the references above and [34, 35], for instance).

Finally, we point out that the same tools were used to design unsteady solvers by Li and Zhang [36] and Chigullapalli and Alexeenko [37].

RECENT IMPROVEMENTS

More accurate and faster Boltzmann solvers

We pointed out in section that most of DVM based Boltzmann solvers were too computationally expensive and not accurate enough. Several solutions were recently proposed: for the first point, we mention the Monte-Carlo random sampling, Korobov sequences, and importance sampling methods. For the second point, it seems that the projection methods of Popov and Cheremisin [12] and Morris et al. [38] work very well for uniform Cartesian grids. The extension of [38] by Arslanbekov et al. [39] to non uniform grids gives good results. However, note that all these approaches (except the Korobov sequences) use a kind of stochastic process, and hence are not fully deterministic. For details, we refer the reader to [39].

A completely different approach is the spectral method. There are several variants that use the Fourier transform of the collision operator, Carleman representation, fast Fourier transform, among other techniques. This was first proposed by Pareschi and Perthame in 1996 [40], and then developed in many other papers. The first version allowed to obtain a computational cost in $O(N^2)$, hence better than standard DVM. Fast spectral methods were proposed by Mouhot and Pareschi in 2006 [41] and extended to other collision kernels by Wu et al. in 2013 [42] to get a close to optimal cost in $O(N \log N)$. These methods are spectrally accurate, that is to say smaller than any inverse power of N, which is excellent. Contrary to DVM, these methods lack conservation properties (but it can be enforced by a constrained optimization technique [43, 44]), positivity, and preservation of Maxwellian states. However, the spectral accuracy limits the importance of these drawbacks. We refer the reader to the review of Dimarco and Pareschi [45] for an exhaustive bibliography and many technical details. Consequently, these methods seem to be very promising tools, since they combine computational efficiency and high accuracy. We mention they have very recently been successfuly applied to 2D space - 3D velocity problems, like in the works of Wu et al. [46, 47] published in 2014.

We also mention the idea to use finite element or Galerkin approximation of the Boltzmann collision operator which has been recently renewed by Majorana [48], Ghiroldi and Gibelli [49], and Alekseenko and Josyula [50]. They obtain very accurate solvers (with spectral accuracy), but with a huge computational cost and memory storage requirements, since both are larger than $O(N^3)$, which is more expensive than standard DVM.

Adaptive velocity grids

In aerodynamics, for high speed flow like in re-entry problems, the Mach number is very large and induces a prohibitively dense velocity grid.

For steady flows, it was first proposed by Baranger et al. in 2012 [51, 52] to locally refine the velocity grid wherever it is necessary, that is to say where there are narrow distributions. This was made possible by an estimation of the spreading of the distributions in the velocity space by a Navier-Stokes pre-simulation. This technique allows an important reduction of the number of discrete velocities (by a factor 30 in 3D).

For unsteady flows, it is attractive to use adaptive grids, that is to say a velocity grid which is refined differently for different time and space positions. This was proposed by Chen et al. in 2012 [53] for BGK models and Arslanbekov et al. in 2013 [39] for the Boltzmann equation. One of the difficulty here is to exchange information between the velocity grids of two neighboring space cells, but it was solved in these two papers and gives very efficient methods.

However, these approaches are still based on the same background velocity grid, or in other words, the bounds of the adaptive grids are fixed at the beginning of the computation and are constant in time and space. It is therefore attractive to allow for a dynamic adaptation of these bounds in time and space. The first attempt in this direction was proposed by Aristov in 1977 [54]. A difficulty is to properly estimate the bounds, and it was shown by Brull and Mieussens in 2014 [55] that it could be done by using the conservation laws. This method was only proposed for one dimensional velocity variable so far.

High order space accuracy

For flows in complex geometries, it is important to have solvers that can be used with unstructured grids with good accuracy. This was proposed by Titarev for the BGK equation in 2009 [30] with a finite volume method (FV). In this method, the distribution f is constant in each cell, and high order accuracy is obtained by a high order

polynomial reconstruction of f to compute the numerical fluxes at cell interfaces. This was done for various types of cells (hexahedral, tetrahedral, prisms, etc.).

Another method is the Discontinuous Galerkin approach (DG), another well known method of CFD (originally invented for kinetic models of neutron transport [56]). Here, the distribution is directly defined by a high order polynomial in each cell. The main advantage is that it naturally gives high order accuracy at the boundaries, since there is no need to use extra ghost cells to define the numerical boundary conditions, as opposed to FV methods. This has been applied in RGD for the first time by Su et al. in 2012 [57] for a 2D problem up to third order accuracy, and was shown to be more efficient than a standard second order FV method on this problem. This has also been demonstrated for a 1D Couette flow by looking at the heat flux at the boundary by Hérouard in 2014 [58]: in a comparison of second order FV and DG schemes: while they have the same order of accuracy, DG gives much more accurate results, and for a given accuracy, DG is more economic (it requires many fewer cells).

Multiscale problems

In such problems, the local Knudsen number can be very small in some zones, while it is not small elsewhere. For stability reasons, standard explicit schemes require very small time steps $\Delta t = O(\mathrm{Kn})$. To get correct results, standard space discretization schemes also require very small grid size in case of boundary layers: $\Delta x = O(\mathrm{Kn})$. Multiscale problems are hence very expensive to be solved and require a specific numerical method.

A first idea is to use a coupled solver that solves the Boltzmann equation in large Kn zones, and the Navier-Stokes equations in small Kn zones. This was first proposed by Bourgat et al. [59] in 1996. Many other coupling methods were proposed later, and it was probably most intensively done by Kolobov et al. in 2007 for 3D computations [13] with the so-called Unified Flow Solver (UFS). We refer to this paper for other references.

For unsteady problems, the time coupling can be made very efficient with the "seamless method" proposed by Lockerby et al. [60]. In [61], Patronis and Lockerby proposed a different kind of coupling : they suggest to correct a macroscopic simulation by putting small kinetic patches at various places (or time) in the computational domain. Even if the kinetic simulation is done with a stochastic solver, this very original approach could probably be used with any deterministic solver.

Another idea is to use Asymptotic Preserving schemes (AP). An AP scheme has the following property: it is uniformly stable with respect to the Knudsen number (by using some implicit techniques), and when Kn is very small, it is consistent with a numerical scheme for the corresponding fluid regime (Euler equation for infinitely small Kn, or Compressible Navier-Stokes equations for very small but finite Kn). It is rather standard to get an AP scheme for the Euler regime: a simple splitting scheme can be used for that, with an implicit solver for the collision step (see [45]). It is more difficult to get an AP scheme for the Compressible Navier-Stokes regime: indeed, the uniform stability itself is not sufficient: the scheme must be such that the dissipative terms (shear stress and heat transfer in viscous boundary layers) are correctly captured. For instance, there are simple splitting schemes as the one mentioned above that cannot resolve these viscous effects. This also requires a suitable space discretization.

This is a very active research field and we will only quote the main different approaches. Many schemes are based on some micro-macro decomposition of f (between a part which is close to the local equilibrium and a deviation), see for instance a review in [45]. A recent approach was proposed by Filbet and Jin in 2010 based on a penalization technique [62]. In our opinion, one of the most promising AP scheme is the Unified Gas Kinetic Scheme (UGKS) proposed by Xu and Huang in 2010 [63]: it is a finite volume scheme in which the collisions are taken into account to compute the numerical fluxes. It gives a very accurate scheme which is AP in the compressible Navier-Stokes regime (see an analysis and a comparison to another AP scheme in [64]). It could be interesting to have an implicit version of this scheme to compute steady flows.

Parallel computing aspects

Despite the improvements mentioned above, multidimensional RGD problems still require very large CPU time and memory storage. This is why using parallel computers is absolutely necessary. This requires development of specific parallel implementation of the numerical methods.

A standard way to parallelize a code is to use a domain decomposition technique: the computational domain is decomposed into several sub-domains, and all the sub-domains are treated by one processors in parallel. Since kinetic

equations contain two different variables (space and velocity), two different domain decomposition can be used. A velocity domain decomposition was first proposed by Li and Zhang in 2008 [36], then by Titarev in 2012 [65]. Titarev et al. then compared this approach to a more standard space domain decomposition in 2014 [66].

These techniques allow to treat large problems that require a lot of memory storage, and give of course an important speed up. However, they do not use the multicore architecture of modern computers. In this architecture, the computer is made of several processors, and each processor contains itself several computational units (the "cores"). There is hence a kind of two-scale parallelism that can be used in the code by using a "hybrid parallelization" which couples the MPI instruction to use several processors and Open-MP instructions to use several cores. This was proposed in 2014 by Baranger et al. [52]: roughly speaking, a space domain decomposition sends a subdomain to each processor (with MPI), and the velocity operations are made in parallel by all the cores of the processors (with Open-MP).

Another kind of multicore architecture is that of the Graphical Processor Unit (GPU). An important fact is that a GPU is much cheaper than a multiprocessor computer. However, it requires very specific implementations, like vectorized operations. This was done in 2010 by Kloss et al. in [67] for the Boltzmann equation in 2D problems , and by Frezzoti et al. in 2011 for the BGK equation [68]. They have shown that this CPU implementation could lead to a very important reduction of the computational cost of a kinetic simulation for RGD. However, these authors mention that the relatively small size of the physical memory of the GPUs can be a important obstacle to its use for kinetic problems. However, in 2013, Zabelok et al. [69] were able to obtain large speed-up for 3D simulations made with a Boltzmann solver on a GPU.

We point out that a very innovative method that shares some similarities with the Lattice-Boltzmann schemes has been recently proposed by Dimarco and Loubère in [70]. This method is particularly fast and has been successfully implemented in a GPU [71].

PERSPECTIVES

Three important problems

We believe that there are still difficult problems in RGD that require further developments. We have chosen to point out three different problems.

The first one is in for re-entry problems. At a numerical level, according to our experience, the convergence of implicit solvers towards the steady solution is slower and slower as the Knudsen number decreases. At a physical level, it is essential to be able to take real gas effects into account in the deterministic solvers: this requires to use multispecies and reactive models.

The second problem is the full simulation of a turbo-molecular pump: it shows very fast flows, multiscale regime, and even chemical reactions with the boundaries when aggressive gases are pumped.

The last problem is that of rarefied flows with moving obstacles. They are particularly difficult when the obstacles have sharp boundaries in 3D geometries, like in the Crookes radiometer [72].

Some attractive solutions

In this section, we discuss some directions of research that could be used to solve the problems described above.

For re-entry and slowly convergent solvers, it seems that using a "fluid preconditioner" could be efficient. This idea comes from linear transport again (where it is called diffusion synthetic acceleration): the idea is to correct the approximate solution given at one step of the deterministic solver by computing an approximation of the error. Under some conditions, this can be done by solving a fluid equation that models the macroscopic evolution of this error. This idea was first applied in RGD by Valougeorgis and Naris in 2003 [26] for the linearized BGK equation (the fluid solver is here a diffusion equation). Then it has recently been proposed for the non linear BGK equation by Mieussens and Pruvost in 2014 [73] (the fluid solver is here a linearized Euler like system). For a 1D Couette flow, this induces a very important speed-up (around 20), with almost no additional cost.

For real gas effects in re-entry, one can first use the extended Boltzmann equation, modified to take into account different species and internal degrees of freedom of the molecules. This has been done by Tcherimissine in 2005 and 2012 [11, 74], Kolobov et al. in 2008 and 2012 [75, 76], and more recently by Munafo et al. in 2014 [77] with a spectral method, for polyatomic and multispecies (but not for reactive gases). Also see the recent work of Anikin et

al. [78]. However, this is very expensive, since there are as many distribution functions as species, and all of them have several additional variables of internal energy. This approach is still restricted to simple 1D problems so far, and is probably much too complex for realistic 2D or 3D simulations, at least for the next few years. Nevertheless, the tools mentioned above are very useful to obtain accurate reference solutions.

We think that using BGK models could be more suitable for realistic simulations, since they allow for a reduction of the internal degrees of freedom. For mixture models, this was already proposed in 1956 by Gross and Krook [79]. Many different models were proposed later, until the one of Kosuge [80], and the model of Brull et al. [81] (see the review in this last paper for many other references). Such models exist also for reactive gases, like the models proposed in 1999 by Groppi and Spiga [82], or the more recent models of Groppi et al. [83] introduced in 2009 and of Brull and Schneider [84] proposed in 2014. Up to our knowledge, there are no 3D or even 2D simulations made with these models so far, which is probably due to the large computational resources required by these models, even if they are much less expensive than Boltzmann models. However, we think that such computations could be made in the very near future.

For the turbo-molecular pump, we believe that the UGKS could be an efficient tool: as a multiscale solver, it gives a natural coupling with Navier-Stokes equations and it has been used with moving obstacles by using the moving mesh method [53, 85]. However, it is important to note that since it is based on a discrete velocity approximation, it probably suffers from the "ray effect" in large Knudsen number zones, which will require very fine velocity grids in the part of the pump in which there is a free molecular regime. This drawback is shared by most of deterministic solvers that use a Cartesian velocity grid. A solution might be the use of adaptive grids, as described in a previous section.

Finally, for moving obstacles, we mention the very accurate solution obtained in 2014 by Tsuji and Aoki [86] for computing the motion decay of a linear pendulum (see this reference for former works on this subject). This method captures all the discontinuities of the distribution function, which is essential in this case, but can hardly been extended to more complex problems. A more interesting approach is the moving mesh method, as used by Chen et al. in [53] and Dechristé and Mieussens in [87], but it cannot be used for arbitrary moving obstacles. It seems that the most promising methods are based on immersed boundary techniques on Cartesian grids. The computational domain is meshed with a Cartesian grid, and the cells are marked as gas cells or solid cells if they are in the gas or in the solid. This allows for a simple treatment of complex geometries (without meshing softwares). Various techniques differs in the numerical treatment of the interface solid/gas, and they can be classified into two kinds. The first one uses the penalization technique and was studied in 2011 by Arslanbekov et al. [88], in 2012 by Pekardan et al. [89], and in 2013 by Filbet and Tang [90]. The drawback of these approaches is that it is difficult to preserve mass conservation. The second kind of immersed boundary technique is the cut-cell method, as proposed by Dechristé and Mieussens in 2014 for a 2D Crookes radiometer [91].

Conclusion

As it has been recalled above, we now have very reliable tools for basic problems of RGD. Recently, many new numerical methods have been developed, but there is still a need for innovative deterministic numerical methods for some challenging problems. Moreover, we believe that further researches on efficient parallel implementation of deterministic solvers for many-core architectures is of great importance.

We also think it would be interesting to compare some of the recent deterministic solvers with some benchmarks, in order to test the multiscale solvers, the different space discretizations, the implicit methods, and their parallel implementation.

REFERENCES

1. S. Chandrasekhar, *Radiative transfer*, Dover Publications Inc., New York, 1960.
2. A. Huang, D. Giddens, and C. Bagnal, *Physics of Fluids* **10**, 498–502 (1967).
3. G. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Science Publications, 1994.
4. H. Struchtrup, *Macroscopic Transport Equations for Rarefied Gas Flows Approximation Methods in Kinetic Theory*, Interaction of Mechanics and Mathematics, Springer, 2005.
5. T. Płatkowski, and R. Illner, *SIAM Rev.* **30**, 213–255 (1988).
6. R. Gatignol, *Théorie Cinétique des Gaz à Répartition Discrète de Vitesses*, vol. 36 of *Lecture Notes in Physics*, Springer-Verlag, 1975.
7. F. Rogier, and J. Schneider, *Transp. Th. Stat. Phys.* **23**, 313–338 (1994).

8. Z. Tan, and P. L. Varghese, *Journal of Computational Physics* **110**, 327 – 340 (1994).
9. V. A. Panferov, and A. G. Heintz, *Mathematical Methods in the Applied Sciences* **25**, 571–593 (2002).
10. V. V. Aristov, *Direct methods for solving the Boltzmann equation and study of nonequilibrium flows*, vol. 60 of *Fluid Mechanics and its Applications*, Kluwer Academic Publishers, Dordrecht, 2001, ISBN 1-4020-0388-9.
11. F. G. Tcheremissine, "Direct numerical solution of the Boltzmann equation," in *Rarefied Gas Dynamics*, edited by M. C. (Ed.), 2005, vol. 762 of *AIP Conference Proceedings*, pp. 677–685.
12. F. G. C. S. P. Popov, *Comput. Math. Math. Phys.* **39**, 156–169 (1999).
13. V. Kolobov, R. Arslanbekov, V. Aristov, A. Frolova, and S. Zabelok, *Journal of Computational Physics* **223**, 589 – 608 (2007), ISSN 0021-9991.
14. Y. Anikin, O. Dodulad, Y. Kloss, D. Martynov, P. Shuvalov, and F. Tcheremissine, *Vacuum* **86**, 1770 – 1777 (2012).
15. T. Ohwada, *Phys. Fluids A* **5**, 217–234 (1993), ISSN 0899-8213.
16. L. H. Holway, *Phys. Fluids* **9**, 1658 (1966).
17. E. M. Shakhov, *Izv. Akad. Nauk SSSR. Mekh. Zhidk. Gaza* pp. 142–145 (1968).
18. P. Andriès, P. L. Tallec, J.-F. Perlat, and B. Perthame, *Eur. J. Mech. B/Fluids* (2000).
19. V. Rykov, *Fluid Dynamics* **10**, 959–966 (1975).
20. C. K. Chu, *Phys. Fluids* **8**, 12 (1965).
21. Y. Sone, *Kinetic theory and fluid dynamics*, Modeling and Simulation in Science, Engineering and Technology, Birkhäuser Boston Inc., Boston, MA, 2002, ISBN 0-8176-4284-6.
22. Y. M. Shakhov, *Zh. Vychisl. Mat. Mat. Fiz.* **35**, 83–94 (1994).
23. Y. M. Shakhov, *Comp. Maths and Math. Phys* **36**, 1123–1131 (1996).
24. I. N. Larina, and V. A. Rykov, *Comp. Maths. math. Phys* **38**, 1391–1403 (1998).
25. Y. Sone, and S. Takata, *Transport Theory Statist. Phys.* **21**, 501–530 (1992).
26. D. Valougeorgis, and S. Naris, *SIAM J. Sci. Comput.* **25**, 534–552 (electronic) (2003).
27. S. Naris, and D. Valougeorgis, *Eur. J. Mech. B Fluids* **27**, 810–822 (2008).
28. S. Varoutis, D. Valougeorgis, and F. Sharipov, *J. Comput. Phys.* **227**, 6272–6287 (2008).
29. J. Yang, and J. Huang, *Journal of Computational Physics* **120**, 323–339 (1995).
30. V. A. Titarev, *Computational Mathematics and Mathematical Physics* **49**, 1197–1211 (2009).
31. L. Mieussens, *Math. Models and Meth. in Appl. Sci.* **8**, 1121–1149 (2000).
32. L. Mieussens, *J. Comput. Phys.* **162**, 429–466 (2000).
33. R. G. Brook, *A parallel, matrix-free newton method for solving approximate boltzmann equations on unstructured topologies*, Ph.D. thesis, The University of Tennessee at Chattanooga (2008).
34. K. Aoki, P. Degond, and L. Mieussens, *Commun. Comput. Phys.* **6**, 919–954 (2009).
35. V. Titarev, *Commun. Comput. Phys.* **12**, 162–192 (2012).
36. Z.-H. Li, and H.-X. Zhang, *Journal of Computational Physics* **228**, 1116 – 1138 (2009).
37. S. Chigullapalli, and A. Alexeenko, "Unsteady 3D Rarefied Flow Solver Based on Boltzmann-ESBGK Model Kinetic Equations," in *41st AIAA Fluid Dynamics Conference and Exhibit*, AIAA, 2011, aIAA Paper 2011-3993.
38. A. B. Morris, P. Varghese, and D. B. Goldstein, *J. Comput. Phys.* **230**, 1265–1280 (2011).
39. R. R. Arslanbekov, V. I. Kolobov, and A. A. Frolova, *PHYSICAL REVIEW E* (2013).
40. L. Pareschi, and B. Perthame, "A Fourier spectral method for homogeneous Boltzmann equations," in *Proceedings of the Second International Workshop on Nonlinear Kinetic Theories and Mathematical Aspects of Hyperbolic Systems (Sanremo, 1994)*, 1996, vol. 25, pp. 369–382.
41. C.Mouhot, and L.Pareschi, *Math. Comp.* **75**, 1833–1852. (2006).
42. L. Wu, C. White, T. J. Scanlon, J. M. Reese, and Y. Zhang, *Journal of Computational Physics* **250**, 27 – 52 (2013).
43. A. V. Bobylev, and S. Rjasanow, *Eur. J. Mech. B Fluids* **18**, 869–887 (1999), ISSN 0997-7546.
44. I. Gamba, and S. Tharkabhushanam, *J. Comput. Math.* **28**, 430–460 (2010).
45. G. Dimarco, and L. Pareschi, *Acta Numerica* **23**, 369–520 (2014).
46. L. Wu, J. Reese, and Y. Zhang, *Journal of Fluid Mechanics* **746**, 53–84 (2014).
47. L. Wu, J. Reese, and Y. Zhang, *Journal of Fluid Mechanics* **748**, 350–367 (2014).
48. A. Majorana, *Kinet. Relat. Models* **4**, 139–151 (2011).
49. G.P.Ghiroldi, and L.Gibelli, *J. of Comput. Phys.* **258**, 568–584 (2014).
50. A. Alekseenko, and E. Josyula, *Journal of Computational Physics* **272**, 170 – 188 (2014).
51. C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens, *AIP Conference Proceedings* **1501**, 389–396 (2012).
52. C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens, *Journal of Computational Physics* **257, Part A**, 572 – 593 (2014).
53. S. Chen, K. Xu, C. Lee, and Q. Cai, *Journal of Computational Physics* **231**, 6643 – 6664 (2012).
54. V. Aristov, *USSR J. Comput. Math. Math. Phys.* **17(4)**, 261–267 (1977).
55. S. Brull, and L. Mieussens, *Journal of Computational Physics* **266**, 22–46 (2014).
56. W. Reed, and T. Hill, Triangular mesh methods for the neutron transport equation, Tech. rep., Los Alamos ScientiĭňĄc Laboratory (1973), tech. Report LA-UR-73-479.
57. W. Su, A. Alexeenko, and G. Cai, "A Runge-Kutta discontinuous Galerkin solver for 2D Boltzmann model equations: Verification and analysis of computational performance," in *Proceedings of 28th International Symposium on Rarefied Gas Dynamics, 9-13 July 2012*, edited by A. C. Proc., 2012, vol. 1501, pp. 381–388.
58. N. Hérouard, *Méthodes numériques pour la dynamique des gaz raréfiés*, Ph.D. thesis, Université de Bordeaux (2014).
59. J.-F. Bourgat, P. L. Tallec, and M. D. Tidriri, *J. Comput. Phys* **127**, 227–245 (1996).

60. D. A. Lockerby, C. A. Duque-Daza, M. K. Borg, and J. M. Reese, *Journal of Computational Physics* **237**, 344 – 365 (2013).
61. A. Patronis, and D. A. Lockerby, *Journal of Computational Physics* **270**, 532 – 543 (2014).
62. F. Filbet, and S. Jin, *Journal of Computational Physics* **229**, 7625 – 7648 (2010).
63. K. Xu, and J.-C. Huang, *J. Comput. Phys.* **229**, 7747–7764 (2010).
64. S. Chen, and K. Xu, A comparative study of an asymptotic preserving scheme and unified gas-kinetic scheme in continuum flow limit (????), preprint.
65. V. A. Titarev, *Communications in Computational Physics* **12**, 162–192 (2012).
66. V. Titarev, M. Dumbser, and S. Utyuzhnikov, *Journal of Computational Physics* **256**, 17 – 33 (2014).
67. Y. Kloss, P. Shuvalov, and F. Tcheremissine, *Procedia Computer Science* **1**, 1083 – 1091 (2010).
68. A. Frezzotti, G. Ghiroldi, and L. Gibelli, *Computers & Fluids* **50**, 136 – 146 (2011).
69. S. A. Zabelok, R. R. Arslanbekov, and V. I. Kolobov, Gpu accelerated kinetic solvers for rarefied gas dynamics (????), aIAA-2013-2863.
70. G. Dimarco, and R. Loubère, *Journal of Computational Physics* **255**, 680–698 (2013).
71. G. Dimarco, R. Loubere, and J. Narski, Towards an ultra efficient kinetic scheme. part iii: High performance computing (2014), submitted.
72. A. Ketsdever, N. Gimelshein, S. Gimelshein, and N. Selden, *Vacuum* **86**, 1644 – 1662 (2012), vacuum Gas Dynamics: Theory, experiments and practical applications.
73. L. Mieussens, and F. Pruvost, A fluid preconditioner for stationary rarefied flow simulations (2014), in preparation.
74. F. G. Tcheremissine, *Computational Mathematics and Mathematical Physics* **52**, 252–268 (2012).
75. V. Kolobov, R. Arslanbekov, S. Bayyuk, V. Aristov, A. Frolova, S. Zabelok, and F. Tcheremissine, Unified kinetic approach for simulations of gas flows in rarefied and continuum regimes, Tech. rep., AFRL-RB-WP-TR-2008-3031, Air Force Research Laboratory (2008), CFD Research Corporation.
76. V. Kolobov, and R. Arslanbekov, *Journal of Computational Physics* **231**, 839 – 869 (2012).
77. A. Munafò, J. R. Haack, I. M. Gamba, and T. E. Magin, *Journal of Computational Physics* **264**, 152 – 176 (2014).
78. Y. A. A. abd O.I Dodulad, Y. Y. Kloss, and F. G. Tcheremissine, *International Journal of Computer Mathematics* **0**, 1–15 (0).
79. P. Gross, and M. Krook, *Phys. Rev.* **102**, 593–604 (1956).
80. S. Kosuge, *European Journal of Mechanics - B/Fluids* **28**, 170–184 (2009).
81. S. Brull, V. Pavan, and J. Schneider, *European Journal of Mechanics - B/Fluids* **33**, 74–86 (2012).
82. M. Groppi, and G. Spiga, *Journal of Mathematical chemistry* **26**, 197–219 (1999).
83. M. Groppi, S. Rjasanow, and G. Spiga, *Journal of Statistical Mechanics: Theory and Experiment* **2009**, P10010 (2009).
84. S. Brull, and J. Schneider, *Communications in mathematical science* **12**, 1199âĂŞ1223 (2014).
85. S. Chen, K. Xu, and C. Lee, *Phys. Fluids* p. 111701 (2012).
86. T. Tsuji, and K. Aoki, *PHYSICAL REVIEW E* **89**, 052129 (2014).
87. G. Dechristé, and L. Mieussens, *AIP Conference Proceedings* **1501**, 366–372 (2012).
88. R. Arslanbekov, V. I. Kolobov, and A. A. Frolova, *AIP Conference Proceedings* **1333**, 873–877 (2011).
89. C. Pekardan, S. Chigullapalli, L. Sun, and A. Alexeenko, *AIP Conference Proceedings* **1501**, 358–365 (2012).
90. F. Filbet, and C. Yang, *Journal of Computational Physics* **245**, 43 – 61 (2013).
91. G. Dechristé, and L. Mieussens, An adaptively refined cartesan cut-cell method for boltzmann kinetic equation (2014), in this volume.