Journal of Computational Physics 408 (2020) 109302

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A velocity space hybridization-based Boltzmann equation solver

G. Oblapenko [a,*], D. Goldstein [b], P. Varghese [a,b], C. Moore [c]

[a] *Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2201 E 24th St, Stop C0200, Austin, TX 78712, USA*
[b] *ASE-EM Department, The University of Texas at Austin, 2617 Wichita St., Stop C0600, Austin, TX 78712, USA*
[c] *Sandia National Laboratories, Albuquerque, NM, USA*

ARTICLE INFO

Article history:
Received 18 June 2019
Received in revised form 27 January 2020
Accepted 30 January 2020
Available online 3 February 2020

Keywords:
Rarefied gas dynamics
Direct simulation Monte Carlo
Boltzmann equation
Discrete velocity method

ABSTRACT

In the present work, a new method for simulation of rarefied gas flows is proposed, a velocity-space hybrid of both a DSMC representation of particles and a discrete velocity quasi-particle representation of the distribution function. The hybridization scheme is discussed in detail, and is numerically verified for two test-cases: the BKW relaxation problem and a stationary Maxwellian distribution. It is demonstrated that such a velocity-space hybridization can provide computational benefits when compared to a pure discrete velocity method or pure DSMC approach, while retaining some of the more attractive properties of discrete velocity methods. Further possible improvements to the velocity-space hybrid approach are discussed.

1. Introduction

Perhaps the most widely adopted approach to simulation of rarefied gas flows is the Direct Simulation Monte Carlo (DSMC) method developed by G.A. Bird [1,2]; however, it suffers from statistical noise and requires significant computational and memory resources to accurately simulate trace populations, such as those of the tails of a distribution function, of high-lying internal energy states, or of trace species present in the mixture. As such, DSMC can have problems when the trace species are either the main species of interest (for example, heavy molecules in a light carrier gas in substrate deposition processes [3]); when low populations can have a large influence on the flow, for example, high-velocity tails in dissociation or ionization reactions; in a beam skimmer set-up [4]; or in the case of a high-velocity beam colliding with a cold gas [5].

Several modifications to DSMC have been proposed to improve the resolution of trace species [3,6] and reduce the statistical variance of the method [7–9]. However, the developed weighting approaches improve the accuracy of modeling of trace species only, and do not improve the resolution of the stochastic representation of the non-trace species' distribution function.

The discrete velocity family of methods has been developed as an alternative to DSMC [10–12], and various interpolation schemes are used in order to conserve mass, momentum and energy during the collision process [12–16]. An overview of discrete velocity methods can be found in [17] and [5]. Discrete velocity methods are capable of accurate representa-

[*] Corresponding author.
E-mail addresses: georgii.oblapenko@austin.utexas.edu (G. Oblapenko), david@ices.utexas.edu (D. Goldstein), varghese@mail.utexas.edu (P. Varghese), chmoore@sandia.gov (C. Moore).

https://doi.org/10.1016/j.jcp.2020.109302
0021-9991/© 2020 Elsevier Inc. All rights reserved.

tion of weakly-populated tails of the distribution function, and have been successfully applied to high-fidelity modeling of populations of high-lying internal energy states [18,19].

Lattice Boltzmann Equation methods [20–22], while also restricting the set of possible velocities, differ from the above discrete velocity methods in their use of the BGK linear collision operator [23], whereas discrete velocity methods referenced in the previous paragraph can model the full non-linear collision operator rather than a simplified BGK operator.

Discrete velocity methods can also benefit greatly from variance reduction techniques [24,25], which not only decrease the noise in the distribution function, but also provide a significant speed-up of the simulations. These techniques were originally developed in the context of removing stiffness in explicit time integration of discrete velocity models in the limit of a low Knudsen number [26] and later adapted to DSMC simulations [7]. Outside of DSMC and discrete velocity methods, variance reduction has been applied to Particle-in-Cell methods [27,28], as well as to Fokker-Planck-type equations [29–31]. Variance reduction may also be applied to flows of gases with internal degrees of freedom [5]. Other notable features of the discrete velocity method include the ability to adapt resolution in velocity space (and have independent velocity grids for different species) [18], adapt the noise level (without incurring additional memory cost) [15], accurately model chemical reactions even when reaction probabilities are extremely low [32], and determine the complete distribution function (including internal states) even over many orders of magnitude of variability.

However, the discrete velocity method has its own set of drawbacks: it requires a numerical scheme for the convection step (compared to the straightforward convection step in DSMC); treatment of boundary conditions may require ghost cells and calculation of number density fluxes; and velocity space discretization can be a source of “ripples” in the values of computed macroscopic quantities in case of discontinuous boundary conditions [33–37]. Also, discrete velocity methods have not yet incorporated the vast range of physical models and capabilities available to DSMC (such as state-specific inelastic cross-sections, including those of chemical reactions [38,39], ionization [40] and radiation processes [41,42]; coupling with Particle-in-Cell (PIC) codes [43,44]; the possibility of coupling with quasi-classical trajectory simulations [45]), and in general have seen less application to large-scale simulations.

Research in the area of hybrid methods involving the Boltzmann equation has been mostly focused on the hybridization of DSMC and CFD solvers, using continuum methods in flow regions with a low Knudsen number and utilizing DSMC to accurately model the rarefied parts of the flow; examples of such approaches can be found in [46–49]. While this hybrid approach allows one to save computational effort (compared to a pure DSMC computation of the same flow), the interface between the DSMC and CFD regions, and the noise associated with DSMC (which is a source of noise in the macroscopic boundary conditions for the continuum region) remains an outstanding issue. It is worth mentioning that the lower noise of the discrete velocity method might make it more amenable to such a physical space hybridization with continuum solver [50], especially when the trace species and/or tails of the velocity distribution function are important to the flow dynamics as can be the case in atmospheric pressure gas discharges [51].

However, to the authors' knowledge, few works have attempted a hybridization in velocity space. In [52], an approach based on solving Euler equations in a region of velocity space (assuming a Maxwellian distribution) and a BGK model equation outside of that “equilibrium” region was developed. In [53] several approaches based on discrete velocity models applied to the BGK model equation were considered, including a velocity-space hybrid approach, in which the tails of the distribution function were represented by particles with arbitrary velocities, while the bulk of the distribution function was represented by particles on a discrete velocity grid. However, such a hybridization loses some of the more attractive properties of the discrete velocity methods — namely, the ability to accurately simulate the tails of the distribution function. In [54], the DSMC method was used to represent the bulk of the distribution function, and a discrete Galerkin approach was used to model the tails of the distribution; however, no extensive studies of the errors and computational efficiency of such a scheme were undertaken. In [55], a similar approach was investigated, but the tails of the distribution function were modeled using the BGK collision operator, rather than the full Boltzmann equation.

Therefore, it is of interest to develop and study a computational scheme that is hybrid in its velocity-space representation and does not rely on the assumptions of a near-equilibrium distribution function and the BGK collision operator: the majority of the mass in a cell in physical space being represented by a few large DSMC-type particles (whose velocities are not restricted to a discrete grid), while the tails of the distribution function are simulated using the discrete velocity method. The expected benefits of such an approach are an improvement in computational efficiency compared to a pure discrete velocity method, while retaining the flexibility of DSMC, and ability to accurately resolve the tails of the distribution functions, as well as the populations of trace species.

The paper is structured as follows: In Sect. 2, a brief overview of our present discrete velocity method is given for a single-component gas with no internal degrees of freedom. Then, in Sect. 3, the approach to hybridization in velocity space is described in detail. After that, the hybrid method is applied to two spatially homogeneous problems, results of numerical simulations are presented, and various approaches to modeling of rarefied gas flows are compared.

2. The Boltzmann equation and the discrete velocity method

Let $\phi(\mathbf{r}, \boldsymbol{\eta}, t)$ be the density of the mathematical expectation of the number of particles in an element of phase space $(\mathbf{r}, \mathbf{r} + d\mathbf{r})$, $(\boldsymbol{\eta}, \boldsymbol{\eta} + d\boldsymbol{\eta})$ at time t.

The definitions of macroscopic quantities such as the bulk velocity $\mathbf{u}$, number density n and temperature T through the function ϕ are well-known and not given here, the reader is referred to works such as [2,56,57]).

Defining a reference temperature T_r, mass m_r, number density n_r and a cross-section σ_r, and a characteristic length L, one can define the reference velocity as

$$\eta_r = \sqrt{\frac{2kT_r}{m_r}}, \tag{1}$$

where k is Boltzmann's constant. One can also introduce a reference mean free path λ_r:

$$\lambda_r = \frac{1}{\sigma_r n_r}, \tag{2}$$

and define a Knudsen number as $Kn = \frac{\lambda_r}{L}$.

Now we can introduce dimensionless quantities:

$$\hat{\boldsymbol{\eta}} = \boldsymbol{\eta}/\eta_r;\ \hat{t} = t\eta_r/L;\ \hat{\sigma_t} = \sigma_t/\sigma_r; \hat{\mathbf{r}} = \frac{\mathbf{r}}{L};\ \overline{\phi} = \phi/n_r, \tag{3}$$

where σ_t is the total collision cross-section, integrated over all scattering angles.

One can then write the Boltzmann equation in a scaled form [24]:

$$\frac{\partial\overline{\phi}}{\partial\hat{t}} + \hat{\boldsymbol{\eta}} \cdot \nabla_{\hat{\mathbf{r}}}\overline{\phi} = \frac{1}{Kn}\int\limits_{\hat{\boldsymbol{\xi}}} \left[\overline{\phi}(\hat{\boldsymbol{\eta}}')\overline{\phi}(\hat{\boldsymbol{\xi}}') - \overline{\phi}(\hat{\boldsymbol{\eta}})\overline{\phi}(\hat{\boldsymbol{\xi}})\right]\hat{g}\hat{\sigma}_t d\hat{\boldsymbol{\xi}}, \tag{4}$$

where $\hat{\boldsymbol{\xi}}$ is the scaled velocity of the colliding particle, $\hat{g} = \left|\hat{\boldsymbol{\xi}} - \hat{\boldsymbol{\eta}}\right|$ is the magnitude of the relative collisional velocity, and primed variables denote the post-collisional quantities.

Equation (4) can be discretized by defining the velocity distribution function $\overline{\phi}$ on a finite set of points in velocity space. For a uniform discrete velocity grid, it takes on the following form [24]:

$$\frac{\partial\overline{\phi}}{\partial\hat{t}} + \hat{\boldsymbol{\eta}} \cdot \nabla_{\hat{\mathbf{r}}}\overline{\phi} = \frac{\beta^3}{Kn}\sum_{\hat{\boldsymbol{\xi}}\neq\hat{\boldsymbol{\eta}}} \left[\overline{\phi}(\hat{\boldsymbol{\eta}}')\overline{\phi}(\hat{\boldsymbol{\xi}}') - \overline{\phi}(\hat{\boldsymbol{\eta}})\overline{\phi}(\hat{\boldsymbol{\xi}})\right]\hat{g}\hat{\sigma}_t. \tag{5}$$

Here $\beta = \Delta\eta/\eta_r$ is the scaled step size in velocity space. We can define $\hat{\phi} = \beta^3\overline{\phi}$, with $\hat{\phi}(\hat{\boldsymbol{\xi}})$ being the density of the mathematical expectation of the number of particles in a volume of size β^3 centered around the velocity $\hat{\boldsymbol{\xi}}$. Equation (5) then takes on the following form:

$$\frac{\partial\hat{\phi}}{\partial\hat{t}} + \hat{\boldsymbol{\eta}} \cdot \nabla_{\hat{\mathbf{r}}}\hat{\phi} = \frac{1}{Kn}\sum_{\hat{\boldsymbol{\xi}}\neq\hat{\boldsymbol{\eta}}} \left[\hat{\phi}(\hat{\boldsymbol{\eta}}')\hat{\phi}(\hat{\boldsymbol{\xi}}') - \hat{\phi}(\hat{\boldsymbol{\eta}})\hat{\phi}(\hat{\boldsymbol{\xi}})\right]\hat{g}\hat{\sigma}_t. \tag{6}$$

While the collision operator can be evaluated in a deterministic fashion, this requires $\mathcal{O}(N_v^2)$ operations, where N_v is the number of points on the discrete velocity grid, and thus is very expensive from a computational view. However, the collision operator can be computed using a stochastic Monte-Carlo scheme [5,24], which is the method used in the present work. The number of collisions to be performed is computed as

$$N_{coll} = nint\left(\frac{\Delta\hat{t}\hat{T}^{2/3}}{KnC_{RMS}^2\hat{m}_r^{2/3}\beta^3}\left(\frac{\hat{n} - 2\hat{n}_{neg}}{\hat{n}}\right)^2\right), \tag{7}$$

where *nint* denotes rounding to the nearest integer, $\hat{m}_r$ is the scaled collision-reduced mass, $\hat{n}_{neg}$ is the amount of negative mass in the numerical representation of the velocity distribution function at that particular location in physical space (explained below), and C_{RMS} is a parameter that is proportional to the user-defined level of acceptable noise in the distribution function at equilibrium [5]. Decreasing the value of C_{RMS} leads to lower noise in the distribution function, at the cost of having to compute more collisions.

A collision is computed by selecting a pair of locations on the discrete velocity grid for depletion. The probability of a location being selected is proportional to the absolute magnitude of the distribution function at that point:

$$P(\hat{\boldsymbol{\eta}}) = \frac{|\hat{\phi}(\hat{\boldsymbol{\eta}})|}{\sum_{\hat{\boldsymbol{\xi}}}|\hat{\phi}(\hat{\boldsymbol{\xi}})|}. \tag{8}$$

Once two velocity locations are chosen for a collision, the density at those locations is depleted by the quantity

$$\Delta\hat{\phi} = \Delta\hat{t}\frac{(\hat{n} - 2\hat{n}_{neg})^2}{2KnN_{coll}} sign\left(\hat{\phi}(\hat{\boldsymbol{\eta}})\hat{\phi}(\hat{\boldsymbol{\xi}})\right)\hat{g}\hat{\sigma}_t. \tag{9}$$

The removed mass is then immediately replenished — its velocity is computed using standard collision mechanics [2]; however, since the post-collision velocity does not necessarily lie on the discrete velocity grid, a remapping scheme is used [13,15], which conserves mass, momentum and energy. The remapping scheme utilizes a 7-point stencil; and the scheme delivers negative mass to 3 of the stencil points in order to yield a net conservation of energy. This can result in small amounts of net negative mass at a few velocity space locations, typically near the tails of the distribution function (see [5]). The presence of negative mass is accounted for in Eqns. (7) and (9).

3. Description of the velocity-space hybrid method

Let $\mathcal{D} \subset \mathbb{R}^3$ denote the set of discrete velocities used for the computation. Let us also define a region $\mathcal{F} \subset \mathbb{R}^3$ of velocity space in which particles can have any velocity (i.e. a particle whose velocity lies in $\mathcal{F}$ is not restricted to the discrete velocity grid), as well as define a region $\mathcal{Q} = \mathcal{D} \setminus \mathcal{F}$ in which the distribution function is defined only on the discrete velocity grid.

Fig. 1 shows a schematic of such a hybrid representation of the velocity distribution function (for some fixed value of v_z).

The basic velocity-space hybrid algorithm for simulation of rarefied gas flows consists of the following steps (performed during a single simulation timestep):

1. Collide particles in region $\mathcal{F}$ using DSMC collision mechanics, that is, a No-Time-Counter (NTC) [2] scheme or some other DSMC collision scheme, such as the Majorant Frequency method [58];
2. Collide mass in region $\mathcal{Q}$ with mass in $\mathcal{Q}$ and particles in $\mathcal{F}$ using discrete velocity collision mechanics;
3. Remap any particles that lie outside region $\mathcal{F}$ to the discrete velocity grid in $\mathcal{Q}$;
4. Perform merging of particles in region $\mathcal{F}$ (the necessity of this procedure will be discussed in the next subsections);
5. Perform convection, using a finite difference or finite volume scheme for the distribution function defined on the discrete velocity points in $\mathcal{Q}$ and using Lagrangian convection for the particles in $\mathcal{F}$.

During step 1, a post-collision velocity of a particle may lie in $\mathcal{Q}$, in which case it is remapped onto the discrete velocity grid (this is done after all the collision steps have been performed, so as to allow a DSMC particle to undergo multiple collisions during a single timestep). Conversely, during step 2, a post-collision velocity of the replenishment quantity may lie inside $\mathcal{F}$, in which case the mass is not remapped onto the discrete velocity grid, but rather immediately created as a new particle in $\mathcal{F}$. Finally, during the remapping procedure, one or more points of the remapping stencil may lie inside $\mathcal{F}$, in which case the mass replenished at these points is also created as new particles which are not restricted to the discrete velocity grid. It is important to note that all particles in $\mathcal{F}$ have a different weight associated with them, unlike standard DSMC, where particles usually have identical weights. Even if one started the simulation with equal-weight particles in $\mathcal{F}$, due to the remapping scheme, particles with different computational weights will be created in $\mathcal{F}$.

While variable weight DSMC algorithms have been developed that combine the splitting and merging steps [3,59], they are generally not conservative, and utilize various workarounds in order to ensure energy conservation; in order to avoid this issue, in the present work the collisions of variable weight particles and the merging step are performed separately in order to construct a conservative scheme.

3.1. Collision algorithm

As described previously, three types of collisions are possible:

1. Collisions between particles in region $\mathcal{F}$;
2. Collisions between particles in region $\mathcal{F}$ and mass in region $\mathcal{Q}$;
3. Collisions between mass in region $\mathcal{Q}$.

Since one of the aims of discrete velocity methods is to have a low amount of noise in the tails of the distribution function, collisions of types 2 and 3 are treated using the Monte Carlo collision scheme as detailed in [5]. Collisions between particles in region $\mathcal{F}$ are treated using DSMC collision mechanics, and are performed first — since collisions of types 2 and 3 might create a large number of DSMC particles. In the present work, the No-Time-Counter collision method will be utilized [2] (although the Majorant Frequency method also works well). However, a modified version of the method has to be used to account for the fact that, in general, particles in $\mathcal{F}$ have different weights.

The required number of collisions (for a single-species gas) is computed as follows [60]:

$$N_{coll,\mathcal{F}} = \frac{1}{2}\frac{N_p(N_p-1)(F_{num}\sigma g)_{max}\Delta t}{V}, \tag{10}$$

where N_p is the number of particles in $\mathcal{F}$, V is the volume of the cell in physical space in which the collisions are considered, F_{num} is the number of real gas particles represented by a single model DSMC particle in $\mathcal{F}$, and Δt is the timestep of the simulation. $(F_{num}\sigma g)_{max}$ denotes the maximum value of the product $F_{num} \cdot \sigma \cdot g$ (taken over the whole region $\mathcal{F}$).

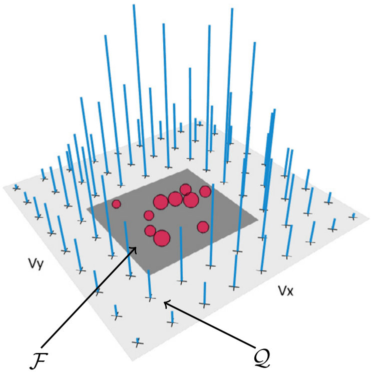

Fig. 1. Schematic showing the velocity-space hybrid representation of the velocity distribution function (for a fixed value of the z-velocity v_z). Blue bars denote the distribution function defined at the discrete grid points in $\mathcal{Q}$ (with their height corresponding to the value of the distribution function at that point), the red circles denote the DSMC particles (with their size corresponding to their computational weight, see below for details) in region $\mathcal{F}$ (shown here as the dark gray rectangle). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Re-writing (10) in terms of the scaled quantities, we obtain

$$N_{coll,\mathcal{F}} = \frac{1}{2Kn} N_p(N_p - 1)(\hat{n}\hat{\sigma}\hat{g})_{max}\Delta\hat{t}, \tag{11}$$

where $(\hat{n}\hat{\sigma}\hat{g})_{max}$ is the maximum value of $\hat{n} \cdot \hat{\sigma} \cdot \hat{g}$ taken over the region $\mathcal{F}$.

When a particle pair is tentatively chosen for a collision, it is accepted with a probability of

$$P = \frac{\hat{n}_1\hat{\sigma}\hat{g}}{(\hat{n}\hat{\sigma}\hat{g})_{max}}. \tag{12}$$

Here $\hat{n}_1$ denotes the weight (dimensionless number density) of the heavier of the two colliding particles.

When a pair is accepted for a collision, the heavier particle (whose weight is denoted $\hat{n}_1$) is split into two particles: one with weight $\hat{n}_2$ (the weight of the lighter collision partner) and one with weight $\hat{n}_1 - \hat{n}_2$. The particles with weights $\hat{n}_2$ collide and their post-collision velocities are computed based on the cross-section model used in the simulation, while the particle with weight $\hat{n}_1 - \hat{n}_2$ does not have its velocity affected.

Due to the fact that the remapping scheme causes negative mass to be created at the “exterior” stencil points, negative mass particles may be present in region $\mathcal{F}$. When colliding a negative mass particle from $\mathcal{F}$ with another particle from $\mathcal{F}$, if one uses the particle splitting scheme as described above, an additional negative mass particle is created as a result of the splitting. Another option is to treat such collisions using the discrete velocity method Monte Carlo collision scheme [24] (which, due to the small depletion quantity involved, might not create as much negative mass in the system), and use the DSMC collision scheme only for collisions between positive mass particles from $\mathcal{F}$. This is the approach used in the present work.

It should be noted that the developed algorithm places no constraints on the size of $\mathcal{F}$ or where it lies in velocity space (e.g., in Fig. 1 it is offset from the center); moreover, for a multi-species flow, different-sized regions $\mathcal{F}$ may be defined for each species. As a limiting case, one can also do species-wise velocity hybridization, treating some chemical species in a flow using the discrete velocity method, and modeling others with DSMC.

In the present work, we consider the case where $\mathcal{F}$ contains the bulk of the distribution function. As such, the values of the lower-order moments (number density, velocity, temperature) are governed by the particles in $\mathcal{F}$, and therefore, it is expected that the performance of the hybrid scheme in terms of dependence on Knudsen and Mach numbers will be similar to that of standard DSMC. Possible improvements at low Knudsen numbers might be obtained by using a Fokker-Planck-based DSMC scheme [61,62].

3.2. Merging algorithm

It can be seen that the number of particles in $\mathcal{F}$ will constantly grow, both due to particle splitting and due to creation of new particles as a result of both the remapping scheme and some post-collision replenishment quantities appearing in $\mathcal{F}$. Additionally, the total amount of negative mass will also keep increasing, which leads to an increase in the computational cost of the collision loop (due to a larger number of collisions as defined by (7)), as well as to an increase in the level of noise in the representation of the distribution function. Therefore, a strategy for merging particles in $\mathcal{F}$ is also needed.

While merging algorithms for performing a $2:1$ merge have been extensively developed [63–66], they are non-conservative by their nature, and employ various techniques to reduce errors and enforce conservation. In order to preserve mass, momentum and energy, a $N:2$ merge needs to be performed (with $N \geq 3$) [67–69]. Moreover, in order to avoid excessive thermalization of the distribution function, particles should be merged together based on their proximity in velocity

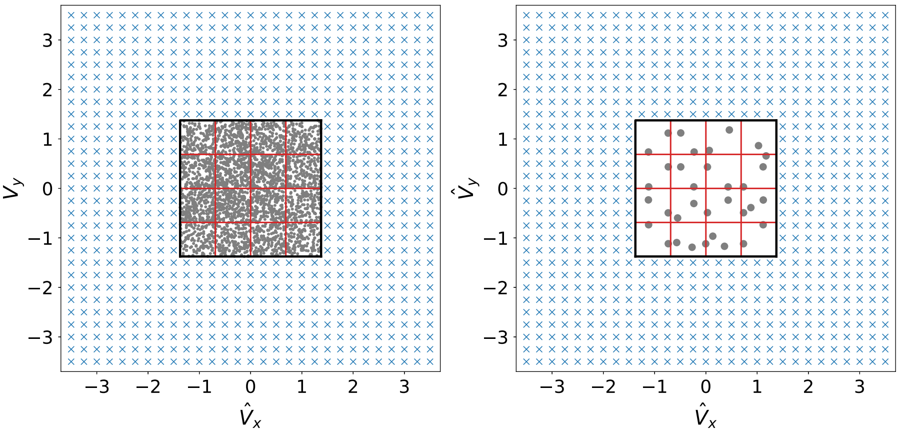

Fig. 2. Example of a merging grid for a fixed value of $\hat{V}_z$; blue symbols denote the points on the discrete velocity grid, gray dots denote the DSMC particles, black lines denote the boundaries of $\mathcal{F}$, and the red lines show the merging grid cells — $M = 16$ in this $\hat{V}_z$ plane, $M = 64$ in total. The left subplot shows the pre-merge particle, the right — the post-merge particles (a merge in a velocity-space cell does not necessarily produce particles that lie in the same merging cell, therefore, the number of post-merge particles in a velocity-space merging cell is not necessarily equal to 2).

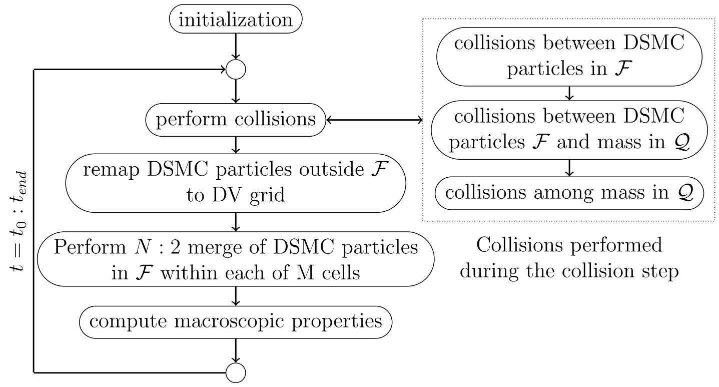
initialization
perform collisions
remap DSMC particles outside $\mathcal{F}$ to DV grid
Perform $N : 2$ merge of DSMC particles in $\mathcal{F}$ within each of M cells
compute macroscopic properties
$t = t_0 : t_{end}$
collisions between DSMC particles in $\mathcal{F}$
collisions between DSMC particles $\mathcal{F}$ and mass in $\mathcal{Q}$
collisions among mass in $\mathcal{Q}$
Collisions performed during the collision step

Fig. 3. Flow chart of the hybrid velocity-space algorithm, t_0 is the start time of the simulation and t_{end} is the end time.

space [69]. It should be noted that performing a $N : 3$ or $N : 4$ merge, while providing the possibility of conserving of all second-order moments (and some third-order moments in the case of the $N : 4$ merge), is complicated by having to solve a system of non-linear equations numerically [70], and thus, such approaches are not investigated in the present paper.

In the present work a uniform grid (unrelated to the discrete velocity grid) is overlaid over $\mathcal{F}$, and all particles within a single grid cell are merged into two particles ($N : 2$ merge). An example of such a grid is shown in Fig. 2. Due to the possibility of a particle having negative mass, the merge within a single such cell might fail, if either one of the following conditions is found to be true: 1) the total mass in the cell is negative or 2) the total energy in the cell is negative. However, the frequency of such events is relatively low, and if a total negative mass and/or energy is accumulated in a merging cell at a time-point, in most cases, during the next timestep, enough positive mass and/or energy will be input into the cell for the merge to be performed.

The merging algorithm is linear with respect to the number of particles N_p, and requires $\mathcal{O}(M)$ additional memory, where M is the number of cells in the velocity-space merging grid. We note, however, that such extra memory is used for only one spatial cell at a time and so is trivial for realistic multidimensional problems. Therefore, the memory required by the hybrid scheme scales linearly with the number of physical dimensions and spatial grid cells, as in standard DVM and DSMC methods. The amount of memory required by the merging procedure is constant and independent of the number of physical grid cells, since collisions and merging are done on a cell-by-cell basis.

The algorithm iterates over the particles three times, first computing and storing the number density in each merging grid cell, as well as computing and storing the bulk velocity of the particles in each cell. On the second pass, the kinetic energy (relative to the bulk velocity) in each cell is computed and stored, and after that, the mass and velocity components of the two-post merge particles in each cell are computed.

While it would be reasonable to assume that this merging procedure limits the number of particles in $\mathcal{F}$ to $2M$, the number M of cells in the merging grid does not set a strict limit on the number of particles being simulated – as discussed above, due to the presence of negative mass, it is possible that no merging will be performed at all in some merging cells. However, on average, the number of particles is indeed limited to approximately $2M$ (see results of numerical simulations below). Use of adaptive merging algorithms in order to better conserve higher-order moments and avoid issues related to the presence of negative mass will be explored in future work.

Fig. 3 presents the developed hybrid approach in a flow chart form for convenience.

4. Numerical results and discussion

In order to verify the velocity-space hybrid method, two spatially homogeneous test cases were considered: the BKW relaxation problem [71,72] and the case of an initial Maxwellian distribution. For both test cases, the reference temperature was set to 273 K, the reference number density was equal to 10^{23} m^{-3}.

The extent of the discrete velocity grid was ±3.5 for both test cases (that is, the velocity grid extends to ±3.5 times the mean thermal speed η_r in each Cartesian direction), and two grids were considered: a "fine" grid with $29 \times 29 \times 29 = 24,389$ points (with a grid spacing of $\beta = 0.25$) and a "coarse" grid with $15 \times 15 \times 15 = 3,375$ points (with a grid spacing of $\beta = 0.5$). The region $\mathcal{F}$ was considered to be a cube centered around the origin in velocity space. Two different extents of the region $\mathcal{F}$ were considered: ±1.375 and ±1.875 (that is, ±1.375 and ±1.875 times the mean thermal speed η_r in each Cartesian direction). The two different-sized regions $\mathcal{F}$ will be denoted as $\mathcal{F}_{small}$ and $\mathcal{F}_{large}$, correspondingly. On a fine grid, when the gas is initialized with a Maxwellian distribution, approximately 86% of the mass is contained in $\mathcal{F}_{small}$, and 98% – in $\mathcal{F}_{large}$; these regions occupy 1331 and 3375 points of the discrete velocity grid, respectively. On a coarse grid, the corresponding values are similar at 80% and 96%; $\mathcal{F}_{small}$ occupies 125 points of the discrete velocity grid and $\mathcal{F}_{large}$ occupies 343 points.

Three different merging grids were used in the hybrid simulations, all with an equal number of merging cells in each velocity direction: with $M = 4 \times 4 \times 4 = 64$ cells (corresponding to $\sim$128 post-merge particles), $M = 6 \times 6 \times 6 = 216$ cells (corresponding to $\sim$432 post-merge particles), and $M = 8 \times 8 \times 8 = 512$ cells (corresponding to $\sim$1024 post-merge particles).

Initialization was performed in the following manner: the initial distribution function (either the BKW distribution or the Maxwellian distribution) was evaluated at each grid point in $\mathcal{D}$, and the value of the distribution function at each discrete velocity was then normalized by the same factor so that the total number density was equal to 1. In case of a hybrid code, all the mass in $\mathcal{F}$ was treated as DSMC-type mass, and a merge was performed at $t = 0$ (otherwise the number of DSMC particles at $t = 0$ will be equal to the number of velocity grid points inside $\mathcal{F}$, and can be much higher than the number of DSMC particles during the following timesteps in case a fine grid is used). It should be noted that the initialization scheme used in the simulations only preserves the number density (compared to the true initial distribution), and leads to some small error in the higher-order moments; this can be mitigated by using a finer discrete velocity grid. Using a coarser grid leads to underestimation of the moments, as discussed in [5]. Another approach for the Maxwellian test-case would be to compute the exact equilibrium discrete distribution, as described in [52,73], and use it for initialization, however, for the purposes of this work this is not required. The problem of how to best represent a smooth analytic probability density function on a cubic grid for the purpose of simply establishing the initial conditions for a simulation will not be further examined here.

The simulations were run with the following values of C_{RMS}: 7.5×10^{-4}, 1.5×10^{-3}, 2.5×10^{-3}, 5×10^{-3}, 7.5×10^{-3}, 1×10^{-2}. At higher values of C_{RMS}, the discrete velocity method can become unstable to due large amounts of negative mass in the tails of the distribution function caused by the large depletion quantity being used during the crudely represented collisions.

The test gas considered was argon, and the Variable Hard Sphere (VHS) cross-section was used for the Maxwellian test case, with a collision diameter $d_{coll} = 4.17$ Å and $\omega = 0.81$ (for the BKW relaxation test case, a pseudo-Maxwell model was used for the collision cross-section). For all the test cases, a scaled time step equal to 0.1 of the scaled collision time was used: $\Delta\hat{t} = 0.1\hat{t}$.

4.1. BKW relaxation

The BKW relaxation is an analytic time-dependent solution of the Boltzmann equation for the case of a pseudo-Maxwell collision cross-section [71,72]. The analytic expression for the distribution function is given by

$$f(\mathbf{v}, t) = \frac{1}{(\pi C)^{3/2}} \frac{1}{2C} \left(5C - 3 + \frac{2(1-C)v^2}{C}\right) exp(-v^2/C), \tag{13}$$

where $C = 1 - 2/5exp(-t/6)$ is defined for all $t \geq 0$.

Analytic expressions for the moments of the distribution can also be obtained, and take on the following form (moments are scaled so that their limit as $t \to \infty$ is equal to 1):

$$\hat{M}^{an}_{2l}(t) = C^l \left(l - (l-1)C\right), \tag{14}$$

where $\hat{M}^{an}_{2l}$ is the analytic moment of order $2l$.

Fig. 4 shows the temporal evolution of the 4th dimensionless moment as a function of the dimensionless time. The black line denotes the analytic solution as defined by expression (14). We see that in general, increasing the size of region $\mathcal{F}$, as well as decreasing the number of merging cells M leads to an increased amount of noise in the simulation (even with

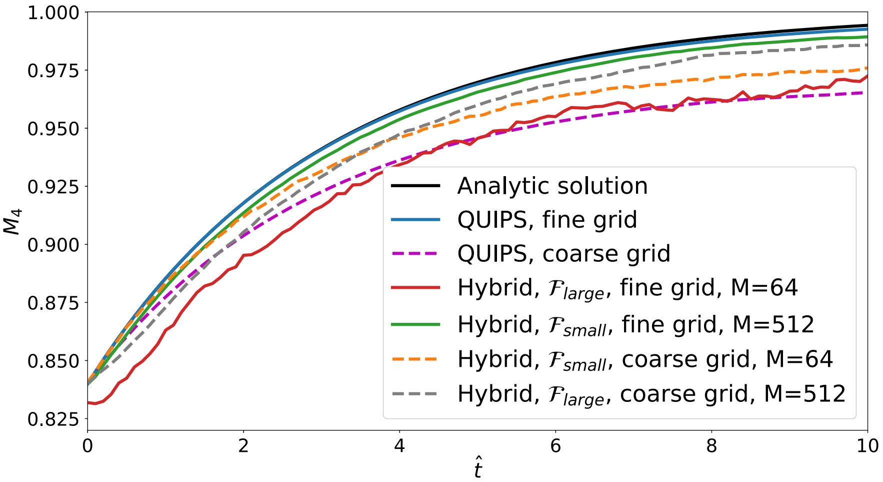

Fig. 4. 4th dimensionless moment (averaged over 200 ensembles) as a function of the dimensionless time for different combinations of grids, sizes of hybrid regions and number of merging cells. "QUIPS" denotes the non-hybrid discrete velocity simulations. For all the simulations, $C_{RMS} = 5 \times 10^{-3}$ was used. The 6th and 8th moments show a similar trend and are not presented.

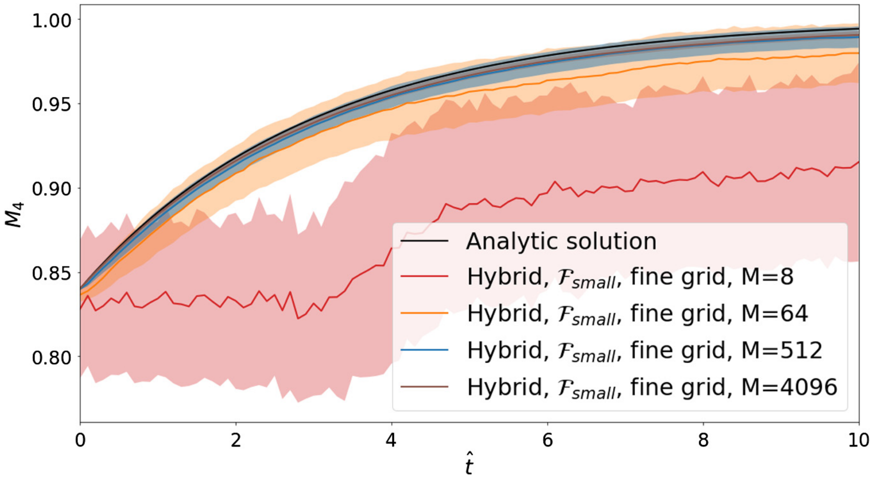

Fig. 5. The 4th dimensionless moment computed on a fine grid with $\mathcal{F}_{small}$ for different numbers of merging cells M; a value of $C_{RMS} = 0.005$ was used. The moments are plotted with the semi-transparent regions showing a range of plus/minus one standard deviation. 200 ensembles were used to compute the means and the standard deviations. The 6th and 8th moments show a similar trend and are not presented.

the ensemble averaging used). It can be seen that moments computed using the hybrid code relax towards the equilibrium value at approximately the expected rate for some of the simulation parameters. Using a low number of merging cells leads to a systematic underestimation of the moments due to 1) a low number of DSMC particles used, which leads to a negative bias [74,75] and 2) a bias introduced by the merging procedure (discussed in detail in Appendix A).

Fig. 5 shows convergence of the 4th moment in the number of DSMC particles used in the simulation (as governed by the number of merging regions M). Using a larger number of DSMC particles leads to a satisfactory convergence to the expected result. Because of the initialization scheme used (via a discrete velocity grid), further increase in the number of merging cells will provide diminishing returns due to the error in the initial distribution.

We now pose the question: how can we quantify the relative benefit of using a hybrid DSMC/discrete velocity approach targeted at improving the representation of the tails of the distribution function in a non-equilibrium flow? Let us consider the following metric: computational efficiency vs deviation of the moments when compared to the analytic solution. The Root Mean Square Error (RMSE) of the N^{th} moment, M_N, compared to the analytic solution is defined as follows:

$$RMSE(\hat{M}_N) = \sqrt{\frac{\sum_{\hat{t}=0}^{10}\left(\hat{M}_N - \hat{M}_N^{an}\right)^2}{100}}, \tag{15}$$

that is, it is computed over the first 100 timesteps (10 collision times). In equation (15) we consider instantaneous values of the moments, and not ensemble-averaged quantities. Ensemble averages of the RMSE themselves yield satisfactorily smooth

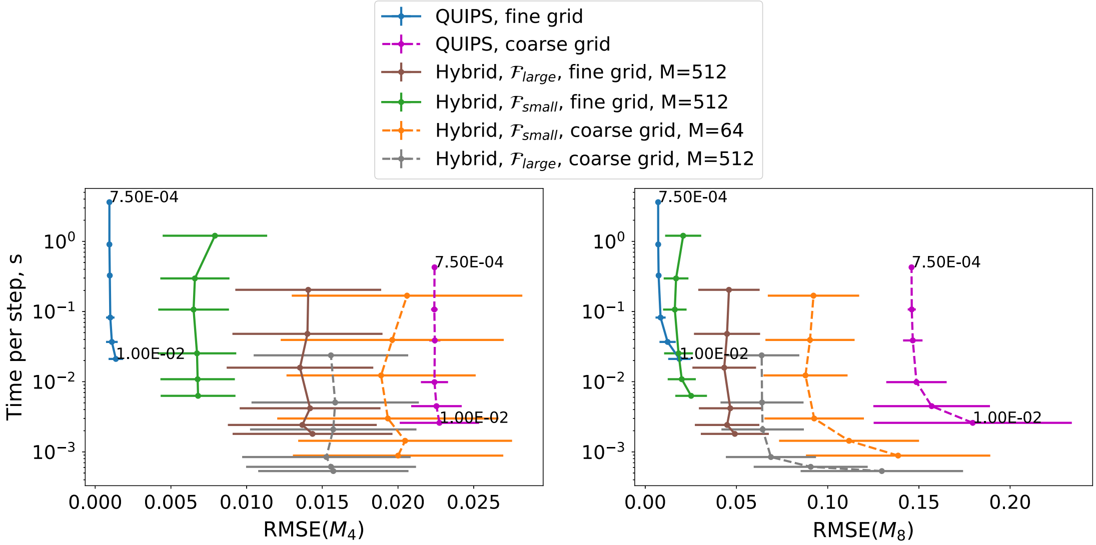

Fig. 6. Average time spent on collisions during a single timestep as a function of the RMSE in the 4th and 8th dimensionless moments (when compared to the analytic solution). Different points on the same curve correspond to different values of the C_{RMS} parameter (the pure discrete velocity curves have the lowest and highest values of C_{RMS} annotated).

results. The amount of computational time spent on calculating the collisions during a single timestep was used as a metric for the computational efficiency of the considered algorithms.

Fig. 6 shows the time spent on calculating the collisions plotted against the error in the moments (compared to the analytic solution). Different points on each curve correspond to different values of C_{RMS} – lowering the value of C_{RMS} leads to increased computational cost and lower error in the moments. It is evident that the C_{RMS} parameter does not have a significant impact on the error in the lower-order moment (the various curves are nearly vertical and the amount of statistical scatter does not change with varying C_{RMS} values) for the hybrid code. This is due to the lower influence of the tails of the distribution function on the lower-order moments. For the higher-order moment (8th moment), the tails of the distribution function have a more significant influence, and thus, the error in the moments in the hybrid simulations is more strongly influenced by the value of C_{RMS}, which mostly affects the $\mathcal{Q}$ region.

To better compare the speed-vs-error performance between the different codes, we also consider the following ratio:

$$\frac{\left(t_{coll} \cdot RMSE(\hat{M}_8)\right)_{QUIPS}}{t_{coll} \cdot RMSE(\hat{M}_8)}, \tag{16}$$

that is, the ratio of the computational time per collision step multiplied by the RMSE of the 8th moment for the discrete velocity code divided by the computational time per collision step multiplied by the RMSE of the 8th moment for a hybrid code (on the same velocity grid). This allows us to simultaneously compare both the noise level and computational efficiency of the various approaches. Fig. 7 shows these ratios as functions of C_{RMS}. In general it can be seen that the hybrid approach can provide an improvement over the purely discrete velocity method, either speed- or error-wise, and this improvement is especially large on a coarse velocity grid. If one only considers computational time, the hybrid code can provide up to a 5-fold speed-up and improved RMSE compared to a pure discrete velocity simulation (see gray and pink curves on Fig. 6). The benefit of the approach of hybridization in velocity-space is especially true for higher-order moments; however, with the considered simulation parameters, the hybrid method cannot achieve as low an error as the discrete velocity method on a fine grid; lowering the error further would require more DSMC particles (more merging cells M).

4.2. Maxwellian distribution

The second test case considered was the behavior of a gas initialized with a Maxwellian distribution. Due to the stochastic nature of the collision algorithm, a certain amount of noise will be present in the numerical simulation; the objective of this test case was to study the influence of various parameters on the noise in the tails of the distribution function and to confirm that the hybrid scheme does not force the equilibrium distribution away from Maxwellian. The tails of the distribution function are defined as all the discrete velocity points lying outside a cube with extent ±1.375, that is, all the points in $\mathcal{Q}_{tail} = \mathcal{D} \setminus \mathcal{F}_{small}$. Since for the larger hybrid region $\mathcal{F}_{large}$ the value of $\hat{\phi}$ is 0 at some of the grid points in $\mathcal{Q}_{tail}$, in order to perform a consistent comparison, during the output step, the particles in $\mathcal{F}_{large}$ are remapped to the discrete velocity grid and the resultant distribution function summed with $\hat{\phi}$ (this is strictly a visualization procedure and the remapping does not affect the actual particles and values of $\hat{\phi}$ used further in the simulation).

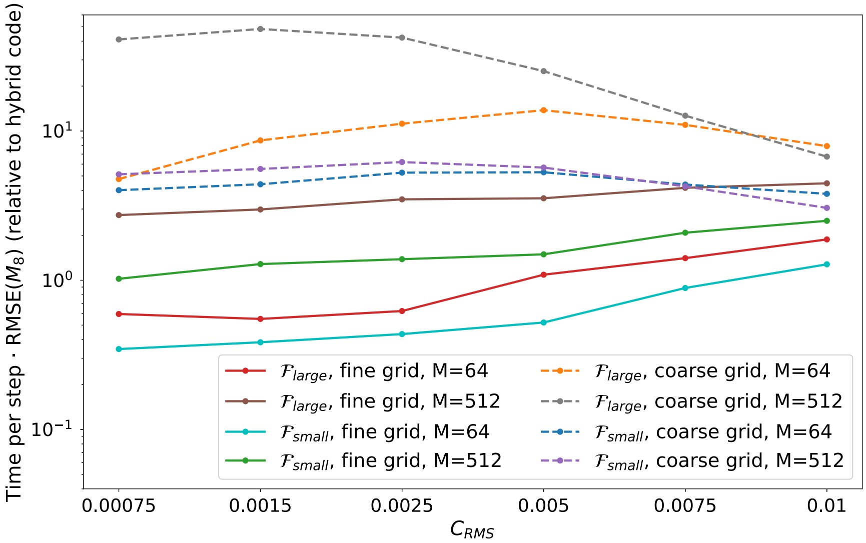

Fig. 7. Ratio of average time spent on collisions during a single timestep multiplied RMSE of the 8th moment for the discrete velocity code divided by the same quantity for a hybrid code as a function of C_{RMS}. A ratio of larger than 1 means that the hybrid code is outperforming a pure discrete velocity code in terms of either computational speed, RMSE, or both.

At each timestep, given the values of $\hat{\phi}_{tail}$ (the set of values of $\hat{\phi}$ in the tails region of velocity space), two metrics are computed, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), defined as

$$RMSE(\hat{\phi}_{tail}) = \sqrt{\frac{\sum_{\hat{\boldsymbol{\eta}}\in\mathcal{Q}_{tail}}\left(\hat{\phi}_{tail}(\hat{\boldsymbol{\eta}}) - \hat{\phi}^{eq}_{tail}(\hat{\boldsymbol{\eta}})\right)^2}{M_{tail}}}, \tag{17}$$

$$MAE(\hat{\phi}_{tail}) = \frac{\sum_{\hat{\boldsymbol{\eta}}\in\mathcal{Q}_{tail}}\left|\hat{\phi}_{tail}(\hat{\boldsymbol{\eta}}) - \hat{\phi}^{eq}_{tail}(\hat{\boldsymbol{\eta}})\right|}{M_{tail}}, \tag{18}$$

where M_{tail} is the number of points on the velocity grid in region $\mathcal{Q}_{tail}$, and $\hat{\phi}^{eq}_{tail}$ is the equilibrium distribution function of the tails. These metrics are computed over 200 timesteps and their means calculated, to be used as measures of noise in the tails of the distribution function.

Due to the initialization procedure which does not produce an exact equilibrium distribution at the initial time $t = 0$ on the truncated discrete grid, one has to let the simulation code run for a few collision times before beginning to accumulate the errors in the tails, so that the initial coarse representation of the Maxwellian distribution can relax to its true equilibrium. The equilibrium distribution of the tails $\hat{\phi}^{eq}_{tail}$ can be computed as a long time-average of the tails distribution $\hat{\phi}_{tail}$:

$$\hat{\phi}^{eq}_{tail} = \sum_{\hat{t}=\hat{t}_1}^{\hat{t}_2} \frac{\hat{\phi}_{tail}(\hat{t})}{N_t}, \tag{19}$$

where $\hat{t}_1$ and $\hat{t}_2$ are the starting and ending times of the averaging process, and N_t is the number of timesteps over which the averaging is performed.

In the present work, the computation of the equilibrium distribution function was begun after 20 collision times (corresponding to 200 timesteps) had elapsed, and the averaging was performed over 300 timesteps (this number of timesteps was found to be sufficient).

Fig. 8 shows the number of particles in region $\mathcal{F}$ as a function of the dimensionless time for different numbers of merging cells M. We see that the number of particles does not grow in an unrestricted fashion, but occasional increases occur due to the presence of negative mass and/or energy in the merging grid cells.

As for the BKW test case, we also want to compare the pure discrete velocity method with the hybridized approach both in terms of noise and computational efficiency, and thus also consider the time spent on performing collisions during a single timestep.

To expand the scope of the present work, we also compare to a pure variable-weight DSMC code based on the hybrid code, as well as the open-source SPARTA DSMC code [76]. The hybrid code can be run as a pure DSMC code (with variable-weight particles) if one assumes that $\mathcal{F} = \mathbb{R}^3$; the initialization step is still based on evaluation of the distribution function

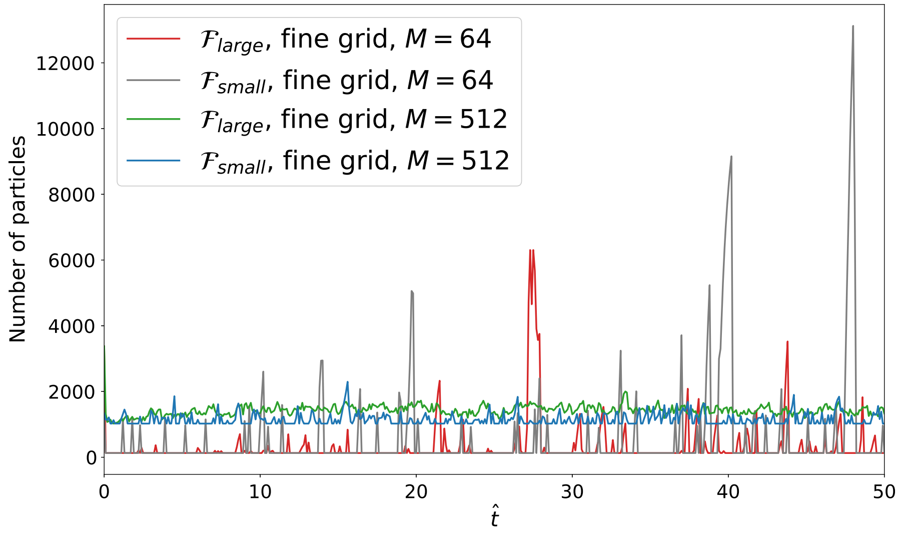

Fig. 8. Number of particles in region $\mathcal{F}$ as a function of time for different numbers of merging cells M; a value of $C_{RMS} = 0.005$ was used.

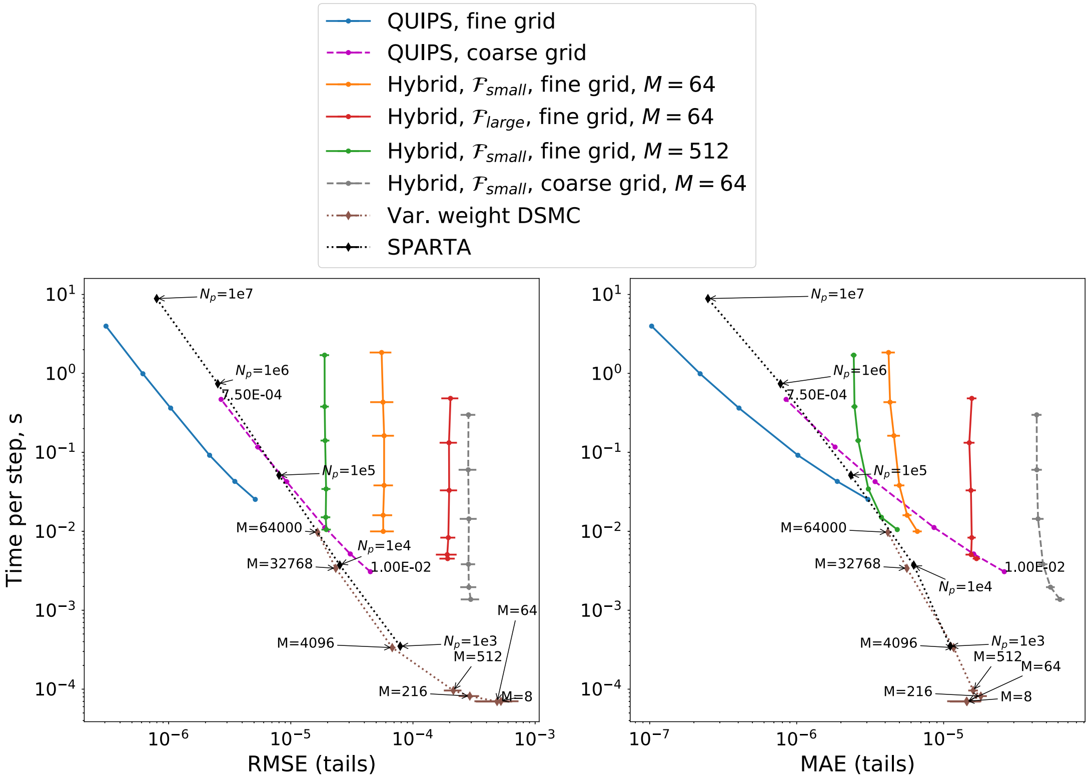

Fig. 9. Time spent on performing collisions during a single timestep plotted against RMSE (left) and MAE (right) in the tails. For the hybrid and QUIPS codes, different points on the same curve correspond to different values of the C_{RMS} parameter (the pure discrete velocity curve on a coarse grid has the lowest and highest values of C_{RMS} annotated); for the variable weight DSMC calculation, different points on the curve correspond to different numbers of merging cells M); for the SPARTA computation, different points correspond to different numbers of model particles N_p.

on a discrete velocity grid (which is only required at this initial step). The total number of DSMC particles is governed by the number of merging cells M, as before (although without any issues related to negative mass). The SPARTA code uses particles with equal computational weight, and thus does not require any merging procedures.

Fig. 9 shows the time spent on performing collisions during a single timestep plotted against the RMSE and MAE tail noise metrics. Different points on the curves correspond to different values of C_{RMS}, which directly affects the computational cost — the upper-left corner of the plot corresponding to low C_{RMS} (high computational cost, low error) and the lower-right

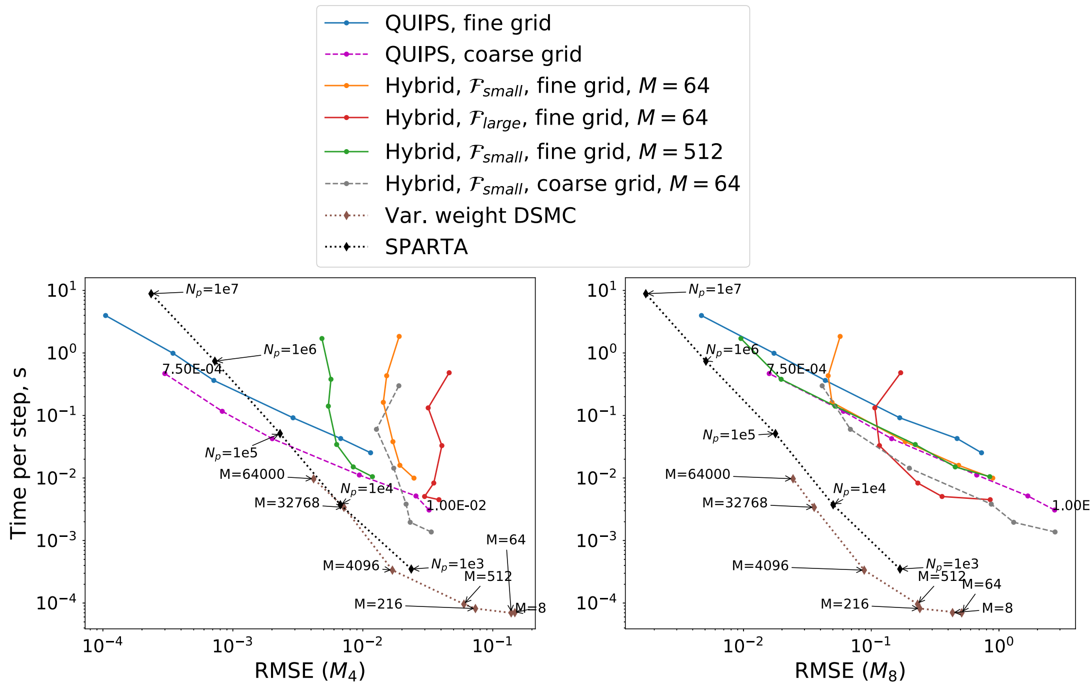

Fig. 10. Time spent on performing collisions during a single timestep plotted against RMSE in the 4th and 8th dimensionless moments. Different points on the same curve correspond to different values of the C_{RMS} parameter (the pure discrete velocity curve on a coarse grid has the lowest and highest values of C_{RMS} annotated); for the variable weight DSMC calculation, different points on the curve correspond to different numbers of merging cells M); for the SPARTA computation, different points correspond to different numbers of model particles N_p.

corner corresponding to high C_{RMS} (low computational cost, high error). Increasing the size of the region $\mathcal{F}$ leads to increased noise in the distribution function, however, the computational effort is reduced only slightly (see orange and red curves). Increasing the number of merging regions M reduces error without a significant increase in computational cost (since the NTC collision procedure is much faster than the Monte Carlo evaluation of the collision integral for the collisions involving mass in $\mathcal{Q}$). It can be seen that at high C_{RMS} values, the hybrid approach is marginally more effective than the pure discrete velocity method, if one looks at the Mean Absolute Error (MAE) metric. The fact that the MAE metric, which is not as sensitive to large errors as is RMSE, is seen to decrease with decreasing values of C_{RMS} for the hybrid method, supports the hypothesis that the noise introduced by the merging procedure and hybridization with a low number of DSMC-type particles is more significant than the noise introduced by the Monte-Carlo scheme for evaluation of collisions involving particles from region $\mathcal{Q}$ (which is influenced by the choice of C_{RMS}). It should be noted that our present hybrid code has not been specifically optimized and thus the metrics presented only serve as a coarse comparison between the non-hybrid and hybrid approaches. Moreover, the computational cost of maintaining accuracy of an already equilibrium distribution is likely different than maintaining accuracy in an evolving flow in which the tails and/or trace species play an important role in the dynamics.

For achieving very low-noise results, the pure discrete velocity method can be seen to be computationally more efficient than DSMC or hybrid computations.

Fig. 10 shows the computational time spent on a collision step plotted against the root-mean-square error in the 4th and 8th moments. While the discrete velocity method can achieve significantly lower noise in the 4th moment than the hybrid or DSMC methods, it does so at a noticeable increase in computational cost. For the higher order (8th) moment, the hybrid approach requires less computational time for a given noise level than the pure discrete velocity method. Both the SPARTA and pure DSMC computations are more cost-efficient than the hybrid and discrete velocity methods for the higher-order moments.

Another important aspect of computational efficiency is memory usage — modern supercomputers are often limited in their computational speed by memory bottlenecks, rather than processing power [77]; this is especially true for hybrid (GPGPU/CPU) architectures, since passing data between General Purpose Graphics Processing Units and CPUs is slow, and therefore, a code with a lower memory usage might be more amenable to large-scale computations. GPGPU units have been successfully used both for DSMC [78–80] and discrete velocity method [81] computations, and therefore, a comparison of memory usage of various computational methods might be indicative of potential scaling issues. It is important to note that for a given velocity grid, achieving lower noise with a discrete velocity method does not require additional memory (only more sampling operations are performed to better approximate the collision integral), whereas as with DSMC, the noise is inversely correlated with the number of simulation particles N_p. That is, to achieve less noise in DSMC, one has to incur not

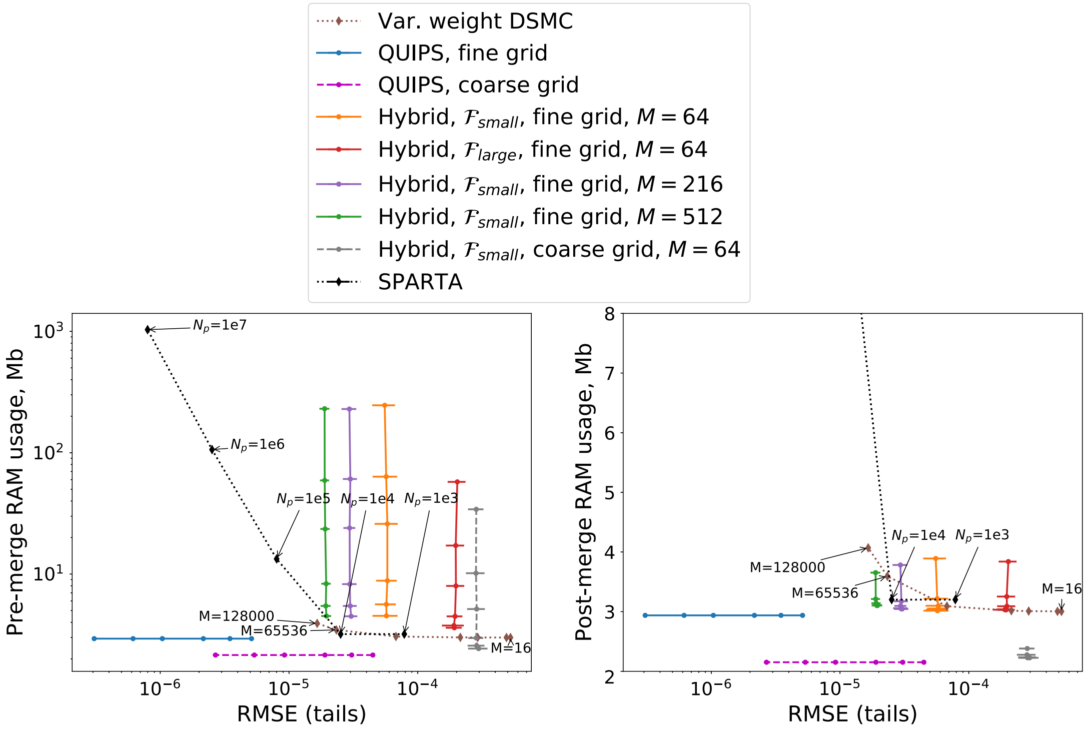

Fig. 11. Memory usage in Mb plotted against RMSE in the tails. Left subplot shows pre-merge memory usage, right subplot shows post-merge memory usage. Different points on the same curve correspond to different values of the C_{RMS} parameter (the pure discrete velocity curve on a coarse grid has the lowest and highest values of C_{RMS} annotated; for the pure DSMC calculation, different points on the curve correspond to different numbers of merging cells M); for the SPARTA computation, different points correspond to different numbers of model particles N_p.

only an increased computational cost, but also an increased memory cost. Moreover, for practical multidimensional problems low noise DSMC requires high memory usage for all cells (or at least many cells) in physical space. Even if a collision scheme that requires a relatively low number of particles per cell to be accurate is used (such as the simplified Bernoulli trials collision procedure [82]), and time or ensemble averaging is employed, it still is problematic to use fixed-weight DSMC to correctly model flows with large number density ratios between different species. For example, if modeling an Ar/e$^-$ mixture with a ratio of the number densities of electrons and argon atoms equal to 0.001, approximately 1000 argon simulator particles are required for each electron model particle; moreover, if the trace species (in this case electrons) are initialized with a non-equilibrium distribution, a larger number of particles is required to avoid initialization bias [75].

Fig. 11 shows the pre- and post-merge memory usage plotted against the RMSE metric for the distribution function. The amount of memory used by the discrete velocity code is dependent only the number of points on the velocity grid, and is not affected by the choice of C_{RMS}; for the hybrid code, the memory usage immediately before the merging procedure (after all the collisions at a given timestep have been performed and the number of particles in $\mathcal{F}$ is high) was used to produce the left subplot. It can be seen that for a given velocity grid and size of $\mathcal{F}$, the memory usage of the hybrid method is virtually independent of the number of merging regions M. This is due to the fact that the plot shows peak memory usage (number of particles before merging), and the number of particles in $\mathcal{F}$ before merging is mostly governed by the choice of C_{RMS}. While the pre-merge memory usage of the hybrid code can be rather high compared to a pure discrete velocity code, it should be emphasized that this is only needed to store the large amount of particles that are created in $\mathcal{F}$ due to the depletion-replenishment collision scheme; and this amount of additional memory would not increase even if one increases the number of cells in physical space, since the collision and merging steps are performed on a cell-by-cell basis. This is in contrast with the SPARTA code, in which memory usage does not change (due to an absence of a merging step), which can make it unfeasible to run complex simulations with large numbers of model particles. The pure (variable-weight) DSMC version of the developed hybrid code is marginally more effective with regards to memory usage than SPARTA, which might be due to its ability to better resolve the tails via particles with small computational weights.

For the hybrid code, the post-merge memory usage still has a dependence on C_{RMS} due to failed merges — if a merge in a merging grid sub-cell fails, a lower C_{RMS} means that more particles remain in that sub-cell, and memory usage is higher. This problem could be alleviated with a more sophisticated merging procedure.

Finally, Fig. 12 shows the pre- and post-merge memory usage plotted against the RMSE of the 8th moment of the distribution function. When one considers the post-merge memory usage, it can be seen that the hybrid method and variable-weight DSMC approaches, as well as the discrete velocity method, are noticeably more memory efficient than a fixed-weight DSMC code.

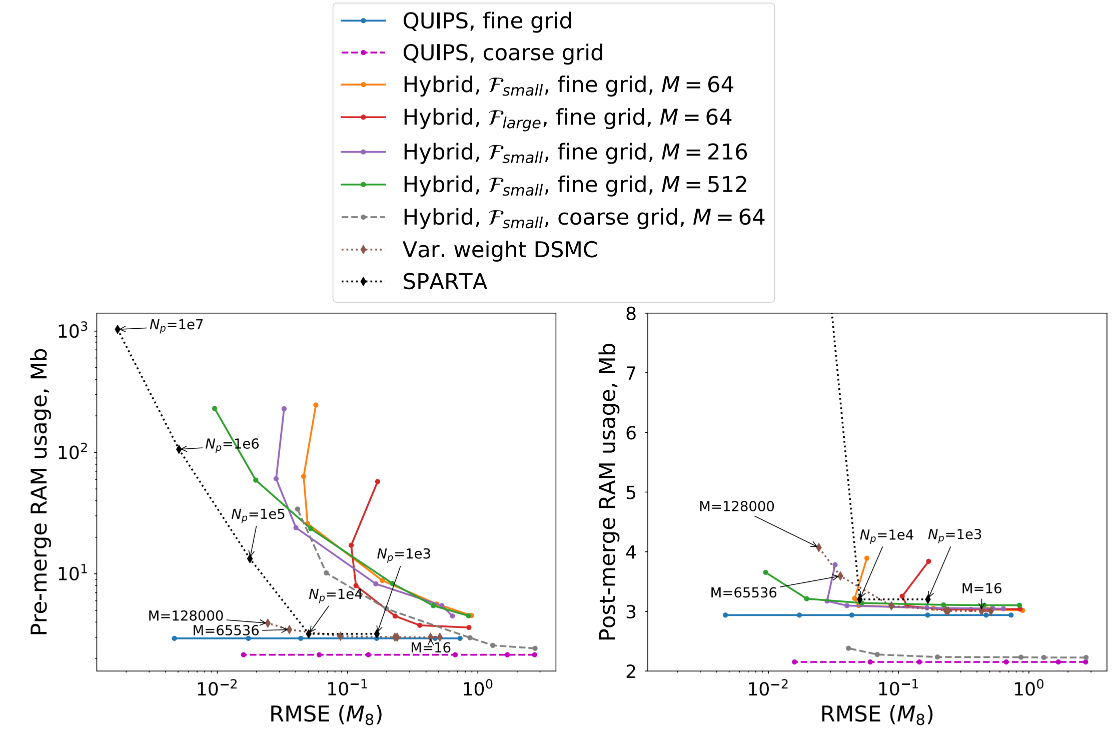

Fig. 12. Memory usage in Mb plotted against RMSE of the 8th moment. Left subplot shows pre-merge memory usage, right subplot shows post-merge memory usage. Different points on the same curve correspond to different values of the C_{RMS} parameter (the pure discrete velocity curve on a coarse grid has the lowest and highest values of C_{RMS} annotated; for the variable weight DSMC calculation, different points on the curve correspond to different numbers of merging cells M); for the SPARTA computation, different points correspond to different numbers of model particles N_p.

It can be concluded that with additional optimization of the computational procedures, the developed hybrid scheme in velocity space might be an attractive strategy for a given acceptable error in the tails, especially if variance reduction is implemented (as that can provide benefits both in terms of computational efficiency and noise level).

5. Conclusion

A new approach to modeling rarefied gas flows based on a hybridization in velocity space has been developed and verified for a single-species monatomic gas.

The hybrid approach has been tested on the BKW relaxation problem, and has been shown to provide improvement in the computational effort-to-noise ratio in moments of the distribution function compared to a purely discrete velocity simulation.

The influence of various simulation parameters on the noise in the tails of the distribution function as well as noise in the moments of the distribution function has been studied for a Maxwellian distribution.

It is shown that the main source of noise in the tails of the distribution function when using the hybrid approach is the low number of DSMC-type particles, rather than the value of C_{RMS} (the noise parameter in the discrete velocity method).

With the basic principle of hybridization having now been successfully demonstrated, further modifications of the hybrid method may include its generalization to flows containing particles with internal energies, multiple-species flows, flows with chemical reactions, and flows in which one implements a variance reduction algorithm [25]; these features have already been proven in DSMC and discrete velocity methods separately, as discussed earlier. The use of a more complex adaptive merging scheme to minimize the amount of noise in the distribution function introduced by the merging process and resolve issues with impossibility of merging due to presence of negative mass will be investigated in future work. The development of other metrics (perhaps more problem-specific) to better understand the trade-offs between various numerical methods and their parameters introduce also remains to be investigated.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Appendix A. Bias introduced by the merging procedure

To estimate the influence of the particle merging procedure on the higher-order moments, and to better understand the amount of noise and bias introduced by the merging, the BKW relaxation test case was run with a 29 × 29 × 29 velocity grid with $\mathcal{F}_{small}$ and different values of the number of merging cells M. The simulation was run with a scaled timestep of 0.1 for 100 timesteps (10 collision times); and at each time step, the difference between the post- and pre-merge value of the considered moments was computed:

$$\Delta\hat{M}_l(t) = \hat{M}_l^{post}(t) - \hat{M}_l^{pre}(t). \tag{A.1}$$

Here $\hat{M}_l^{pre}(t)$, $\hat{M}_l^{post}(t)$ are the values of the scaled moment of order l at time t before and after the merging procedure, correspondingly. The time-averages of these time-series were taken:

$$\Delta\hat{M}_l = \frac{\sum_{\hat{t}=0}^{10} \Delta\hat{M}_l(\hat{t})}{100}, \tag{A.2}$$

and ensemble-averaged over 200 ensembles.

Fig. A.13 shows the temporal evolution of the 4th moment for a single run with $M = 8$ and explicitly shows the changes in the values of the moment due to the merging procedure (red lines). We see that for such a low number of merging cells (8), the merging adds considerable noise.

Fig. A.14 shows the average bias introduced in the 4th moment by the merging procedure as a function of number of merging cells M. It can be seen that merging causes a consistent underestimation of the moments, and that increasing the number of merging cells leads to a reduction in the bias. The bias introduced in the 4th moment was also approximated as a function of M:

$$log_{10}(-\Delta M_4) \approx 0.44 - 0.917M^{1/5}, \tag{A.3}$$

and the analytic fit is also shown on Fig. A.14.

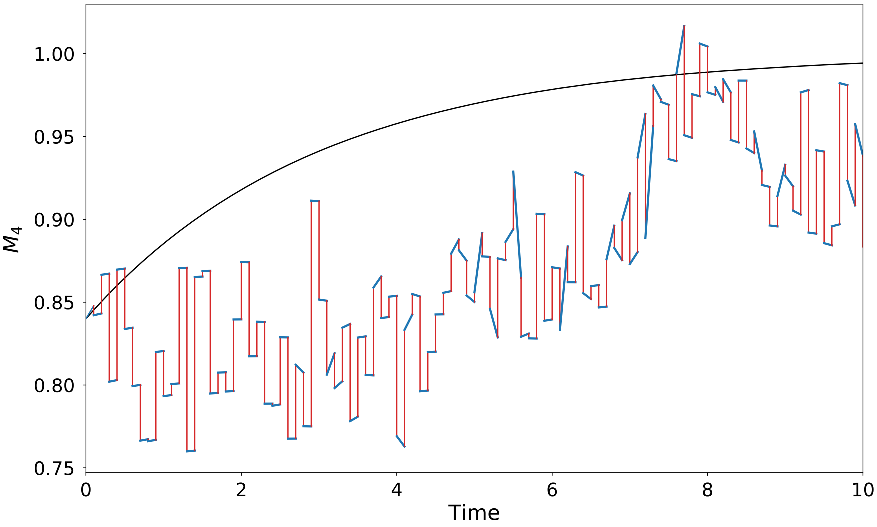

Fig. A.13. Time evolution of the 4th moment for the BKW relaxation. The black curve is the analytic solution. The red lines connect the pre- and post-merge moments at a fixed timestep, while the blue lines connect a post-merge moment at timestep i with a pre-merge moment at timestep $i+1$, that is, they show the temporal evolution of the moment. A C_{RMS} value of 0.005 was used, and the number of merging grid cells M was equal to 8.

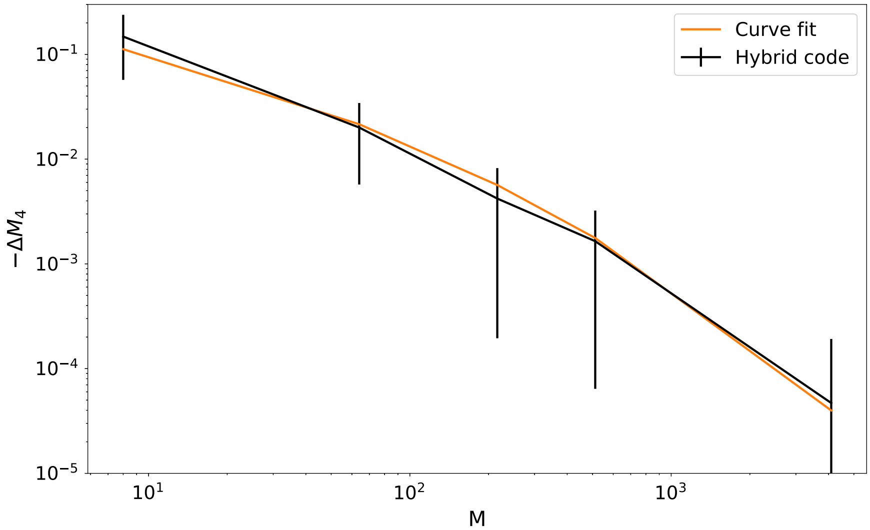

Fig. A.14. Average difference in the 4th moment introduced by the merging procedure for different numbers of merging cells M. Note that all cases lead to a negative change in M_4, but that with a moderate number of merging cells M the magnitude of the error can be quite small. The error bars show a range of plus/minus one standard deviation.

References

[1] G. Bird, Approach to translational equilibrium in a rigid sphere gas, Phys. Fluids 6 (1963) 1518–1519.
[2] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon, Oxford, England, UK, 1994.
[3] I.D. Boyd, Conservative species weighting scheme for the direct simulation Monte Carlo method, J. Thermophys. Heat Transf. 10 (1996) 579–585.
[4] G. Bird, Transition regime behavior of supersonic beam skimmers, Phys. Fluids 19 (1976) 1486–1491.
[5] P.B. Clarke, A discrete velocity method for the Boltzmann equation with internal energy and stochastic variance reduction, Ph.D. thesis, University of Texas at Austin, 2015.
[6] S. Rjasanow, W. Wagner, A stochastic weighted particle method for the Boltzmann equation, J. Comput. Phys. 124 (1996) 243–253.
[7] L.L. Baker, N.G. Hadjiconstantinou, Variance reduction for Monte Carlo solutions of the Boltzmann equation, Phys. Fluids 17 (2005) 051703.
[8] G.A. Radtke, N.G. Hadjiconstantinou, W. Wagner, Low-noise Monte Carlo simulation of the variable hard sphere gas, Phys. Fluids 23 (2011) 030606.
[9] C.R. Schrock, A.W. Wood, Distributional Monte Carlo solution technique for rarefied gasdynamics, J. Thermophys. Heat Transf. 26 (2012) 185–189.
[10] A. Nordsieck, B.L. Hicks, Monte Carlo evaluation of the Boltzmann collision integral, in: Proc. of the 5th Intern. Symposium on Rarefied Gas Dynamics, vol. 1, 1966, pp. 695–710.
[11] D. Goldstein, B. Sturtevant, J. Broadwell, Investigations of the motion of discrete-velocity gases, Prog. Astronaut. Aeronaut. 117 (1989) 100–117.
[12] Z. Tan, P.L. Varghese, The δ-ε method for the Boltzmann equation, J. Comput. Phys. 110 (1994) 327–340.
[13] P. Varghese, Arbitrary post-collision velocities in a discrete velocity scheme for the Boltzmann equation, in: Proc. of the 25th Intern. Symposium on Rarefied Gas Dynamics, 2007, pp. 225–232.
[14] F. Tcheremissine, Solution of the Boltzmann kinetic equation for low speed flows, Transp. Theory Stat. Phys. 37 (2008) 564–575.
[15] A. Morris, P. Varghese, D. Goldstein, Improvement of a discrete velocity Boltzmann equation solver with arbitrary post-collision velocities, in: AIP Conf. Proc., vol. 1084, AIP Publishing, 2008, pp. 458–463.
[16] O. Dodulad, F. Tcheremissine, Multipoint conservative projection method for computing the Boltzmann collision integral for gas mixtures, in: AIP Conf. Proc., vol. 1501, AIP Publishing, 2012, pp. 302–309.
[17] V.V. Aristov, Direct Methods for Solving the Boltzmann Equation and Study of Nonequilibrium Flows, vol. 60, Springer Science & Business Media, 2012.
[18] P. Clarke, P. Varghese, D. Goldstein, A. Morris, P. Bauman, D. Hegermiller, A novel discrete velocity method for solving the Boltzmann equation including internal energy and non-uniform grids in velocity space, in: AIP Conf. Proc., vol. 1501, AIP Publishing, 2012, pp. 373–380.
[19] P. Clarke, P. Varghese, D. Goldstein, A low noise discrete velocity method for the Boltzmann equation with quantized rotational and vibrational energy, J. Comput. Phys. 352 (2018) 326–340.
[20] G.R. McNamara, G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata, Phys. Rev. Lett. 61 (1988) 2332.
[21] F. Higuera, S. Succi, R. Benzi, Lattice gas dynamics with enhanced collisions, Europhys. Lett. 9 (1989) 345.
[22] S. Succi, The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond, Oxford University Press, 2001.
[23] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (1954) 511.
[24] A. Morris, P. Varghese, D. Goldstein, Monte Carlo solution of the Boltzmann equation via a discrete velocity model, J. Comput. Phys. 230 (2011) 1265–1280.
[25] P. Clarke, P. Varghese, D. Goldstein, Discrete velocity computations with stochastic variance reduction of the Boltzmann equation for gas mixtures, in: AIP Conf. Proc., vol. 1628, AIP Publishing, 2014, pp. 1032–1039.
[26] F. Cheremisin, Solving the Boltzmann equation in the case of passing to the hydrodynamic flow regime, in: Doklady Physics, vol. 45, Springer, 2000, pp. 401–404.
[27] A.Y. Aydemir, A unified Monte Carlo interpretation of particle simulations and applications to non-neutral plasmas, Phys. Plasmas 1 (1994) 822–831.
[28] G.J. Wilkie, W. Dorland, Fundamental form of the electrostatic δf- PIC algorithm and discovery of a converged numerical instability, Phys. Plasmas 23 (2016) 052111.
[29] S. Brunner, E. Valeo, J.A. Krommes, Collisional delta-f scheme with evolving background for transport time scale simulations, Phys. Plasmas 6 (1999) 4504–4521.
[30] M.H. Gorji, N. Andric, P. Jenny, Variance reduction for Fokker–Planck based particle Monte Carlo schemes, J. Comput. Phys. 295 (2015) 644–664.

[31] B.S. Collyer, C. Connaughton, D.A. Lockerby, Importance sampling variance reduction for the Fokker–Planck rarefied gas particle method, J. Comput. Phys. 325 (2016) 116–128.

[32] Y. Poondla, P. Varghese, D. Goldstein, K. Higdon, Modeling of chemical reactions using quasi-particle simulation (QuiPS), in: AIP Conf. Proc., vol. 2132, AIP Publishing, 2018, p. 140002.

[33] K.D. Lathrop, Ray effects in discrete ordinates equations, Nucl. Sci. Eng. 32 (1968) 357–369.

[34] W. Fiveland, Discrete-ordinates solutions of the radiative transport equation for rectangular enclosures, J. Heat Transf. 106 (1984) 699–706.

[35] J.C. Chai, H.S. Lee, S.V. Patankar, Ray effect and false scattering in the discrete ordinates method, Numer. Heat Transf., Part B 24 (1993) 373–389.

[36] A. Sekaran, P. Varghese, D. Goldstein, An analysis of numerical convergence in discrete velocity gas dynamics for internal flows, J. Comput. Phys. 365 (2018) 226–242.

[37] M.T. Ho, J. Li, L. Wu, J.M. Reese, Y. Zhang, A comparative study of the DSBGK and DVM methods for low-speed rarefied gas flows, Comput. Fluids 181 (2019) 143–159.

[38] D.C. Wadsworth, I.J. Wysong, Vibrational favoring effect in DSMC dissociation models, Phys. Fluids 9 (1997) 3873–3884.

[39] J. Kim, I. Boyd, State resolved thermochemical modeling of nitrogen using DSMC, in: 43rd AIAA Thermophysics Conference, 2012, p. 2991.

[40] D. Liechty, M. Lewis, Treatment of electronic energy level transition and ionization following the particle-based chemistry model, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, p. 449.

[41] A.B. Carlson, A. Hassan, Radiation modeling with direct simulation Monte Carlo, J. Thermophys. Heat Transf. 6 (1992) 631–636.

[42] A. Fierro, C. Moore, B. Scheiner, B.T. Yee, M.M. Hopkins, Radiation transport in kinetic simulations and the influence of photoemission on electron current in self-sustaining discharges, J. Phys. D, Appl. Phys. 50 (2017) 065202.

[43] V.V. Serikov, S. Kawamoto, K. Nanbu, Particle-in-cell plus direct simulation Monte Carlo (PIC-DSMC) approach for self-consistent plasma-gas simulations, IEEE Trans. Plasma Sci. 27 (1999) 1389–1398.

[44] A. Fierro, C. Moore, B. Yee, M. Hopkins, Three-dimensional kinetic modeling of streamer propagation in a nitrogen/helium gas mixture, Plasma Sources Sci. Technol. 27 (2018) 105008.

[45] T.E. Schwartzentruber, M.S. Grover, P. Valentini, Direct molecular simulation of nonequilibrium dilute gases, J. Thermophys. Heat Transf. 32 (2017) 892–903.

[46] R. Roveda, D.B. Goldstein, P.L. Varghese, Hybrid Euler/particle approach for continuum/rarefied flows, J. Spacecr. Rockets 35 (1998) 258–265.

[47] H. Carlson, R. Roveda, I. Boyd, G. Candler, A hybrid CFD-DSMC method of modeling continuum-rarefied flows, in: 42nd AIAA Aerospace Sciences Meeting and Exhibit, 2004, p. 1180.

[48] H.S. Wijesinghe, N.G. Hadjiconstantinou, Discussion of hybrid atomistic-continuum methods for multiscale hydrodynamics, Int. J. Multiscale Comput. Eng. 2 (2004).

[49] T.E. Schwartzentruber, L.C. Scalabrin, I.D. Boyd, Hybrid particle-continuum simulations of nonequilibrium hypersonic blunt-body flow fields, J. Thermophys. Heat Transf. 22 (2008) 29–37.

[50] V. Kolobov, R. Arslanbekov, Towards adaptive kinetic-fluid simulations of weakly ionized plasmas, J. Comput. Phys. 231 (2012) 839–869.

[51] Y.P. Raizer, Gas Discharge Physics, Springer-Verlag, Berlin Heidelberg, 1991.

[52] N. Crouseilles, P. Degond, M. Lemou, A hybrid kinetic/fluid model for solving the gas dynamics Boltzmann – BGK equation, J. Comput. Phys. 199 (2004) 776–808.

[53] G. Dimarco, L. Pareschi, Hybrid multiscale methods II. Kinetic equations, Multiscale Model. Simul. 6 (2008) 1169–1197.

[54] T.-J. Pan, K.A. Stephani, Investigation of velocity-space coupling approach in DSMC for tail-driven processes, in: AIP Conf. Proc., vol. 1786, AIP Publishing, 2016, p. 050017.

[55] T.-J. Pan, K.A. Stephani, Investigation of a coupling approach of DSMC and DG methods for tail-driven processes, in: 47th AIAA Thermophysics Conference, 2017, p. 4023.

[56] J.H. Ferziger, H.G. Kaper, H.G. Kaper, Mathematical Theory of Transport Processes in Gases, North-Holland, 1972.

[57] S. Chapman, T.G. Cowling, D. Burnett, The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases, Cambridge University Press, 1990.

[58] M. Ivanov, S. Rogasinsky, Analysis of numerical techniques of the direct simulation Monte Carlo method in the rarefied gas dynamics, Russ. J. Numer. Anal. Math. Model. 3 (1988) 453–466.

[59] D. Petkow, A. Mirza, G. Herdrich, S. Fasoulas, Treatment of differently weighted particles in reactive re-entry flows with DSMC, in: AIP Conf. Proc., vol. 1501, AIP Publishing, 2012, pp. 1507–1514.

[60] D.P. Schmidt, C. Rutland, A new droplet collision algorithm, J. Comput. Phys. 164 (2000) 62–80.

[61] M.H. Gorji, P. Jenny, Fokker–Planck – DSMC algorithm for simulations of rarefied gas flows, J. Comput. Phys. 287 (2015) 110–129.

[62] E. Jun, M.H. Gorji, M. Grabe, K. Hannemann, Assessment of the cubic Fokker–Planck – DSMC hybrid method for hypersonic rarefied flows past a cylinder, Comput. Fluids 168 (2018) 1–13.

[63] G. Lapenta, J.U. Brackbill, Dynamic and selective control of the number of particles in kinetic plasma simulations, J. Comput. Phys. 115 (1994) 213–227.

[64] D.W. Hewett, Fragmentation, merging, and internal dynamics for PIC simulation with finite size particles, J. Comput. Phys. 189 (2003) 390–426.

[65] D.R. Welch, T.C. Genoni, R.E. Clark, D.V. Rose, Adaptive particle management in a particle-in-cell code, J. Comput. Phys. 227 (2007) 143–155.

[66] J. Teunissen, U. Ebert, Controlling the weights of simulation particles: adaptive particle management using k-d trees, J. Comput. Phys. 259 (2014) 318–330.

[67] G. Lapenta, J.U. Brackbill, Dynamic and selective control of the number of particles in kinetic plasma simulations, J. Comput. Phys. 115 (1994) 213–227.

[68] S. Rjasanow, T. Schreiber, W. Wagner, Reduction of the number of particles in the stochastic weighted particle method for the Boltzmann equation, J. Comput. Phys. 145 (1998) 382–405.

[69] R.S. Martin, J.-L. Cambier, Octree particle management for DSMC and PIC simulations, J. Comput. Phys. 327 (2016) 943–966.

[70] R.S. Martin, J.-L. Cambier, Moment preserving adaptive particle weights using octree velocity distributions for PIC simulations, in: AIP Conf. Proc., vol. 1501, AIP Publishing, 2012, pp. 872–879.

[71] A.V. Bobylev, One class of invariant solutions of the Boltzmann equation, in: Akademiia Nauk SSSR Doklady, vol. 231, 1976, pp. 571–574.

[72] M. Krook, T.T. Wu, Exact solutions of the Boltzmann equation, Phys. Fluids 20 (1977) 1589–1595.

[73] L. Mieussens, Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics, Math. Models Methods Appl. Sci. 10 (2000) 1121–1149.

[74] D. Rader, M. Gallis, J. Torczynski, W. Wagner, Direct simulation Monte Carlo convergence behavior of the hard-sphere-gas thermal conductivity for Fourier heat flow, Phys. Fluids 18 (2006) 077102.

[75] W. McDoniel, C. Moore, G. Radtke, K. Cartwright, M. Bettencourt, On-average error from DSMC collisions, in: Book of Abstracts, International Conference on Phenomena in Ionized Gases, Sapporo, Japan, 2019.

[76] S. Plimpton, M. Gallis, SPARTA direct simulation Monte Carlo (DSMC) simulator, Sandia National Laboratories, USA, http://sparta.sandia.gov, 2015.

[77] N. Attig, P. Gibbon, T. Lippert, Trends in supercomputing: the European path to exascale, Comput. Phys. Commun. 182 (2011) 2041–2046.

[78] C.-C. Su, C.-W. Hsieh, M. Smith, M. Jermy, J.-S. Wu, Parallel direct simulation Monte Carlo computation using CUDA on GPUs, in: AIP Conf. Proc., vol. 1333, AIP Publishing, 2011, pp. 343–347.

[79] M. Goldsworthy, A GPU–CUDA based direct simulation Monte Carlo algorithm for real gas flows, Comput. Fluids 94 (2014) 58–68.

[80] A. Kashkovsky, 3D DSMC computations on a heterogeneous CPU-GPU cluster with a large number of GPUs, in: AIP Conf. Proc., vol. 1628, AIP Publishing, 2014, pp. 192–198.

[81] E. Malkov, S. Poleshkin, M. Ivanov, Discrete velocity scheme for solving the Boltzmann equation with the GPGPU, in: AIP Conf. Proc., vol. 1501, AIP Publishing, 2012, pp. 318–325.

[82] S.K. Stefanov, On DSMC calculations of rarefied gas flows with small number of particles in cells, SIAM J. Sci. Comput. 33 (2011) 677–702.