Deterministic Solution of the Boltzmann Equation Using a Discontinuous Galerkin Velocity Discretization

A. Alekseenko*,† and E. Josyula†

*Department of Mathematics, California State University Northridge, Northridge, California USA
†Air Force Research Laboratory, Wright-Patterson AFB, Ohio USA

Abstract. We propose an approach for high order discretization of the Boltzmann equation in the velocity space using discontinuous Galerkin methods. Our approach employs a reformulation of the collision integral in the form of a bilinear operator with a time-independent kernel. In the fully non-linear case the complexity of the method is $O(n^8)$ operations per spatial cell where n is the number of degrees of freedom in one velocity direction. The new method is suitable for parallelization to a large number of processors. Techniques of automatic perturbation decomposition and linearisation are developed to achieve additional performance improvement. The number of operations per spatial cell in the linearised regime is $O(n^6)$. The method is applied to the solution of the spatially homogeneous relaxation problem. Mass momentum and energy is conserved to a good precision in the computed solutions.

Keywords: Boltzmann collision operator, discontinuous Galerkin methods
PACS: 47.45.-n, 51.10.+y, 47.45.Ab

INTRODUCTION

Perhaps one of the biggest challenges in gas dynamics today is the development of methods for simulation of non-continuum flows in the regimes of slow velocities and time-dependent regimes. In these cases the traditional statistical methods, such as the direct simulation Monte-Carlo techniques [1] become expensive due to the large computational times necessary to decrease the statistical noise. The deterministic approaches are believed to provide a viable alternative to the statistical methods in such cases. While there has been a considerable advancement in the deterministic simulation of non-continuum gases using model kinetic equations (see, e.g., [2]) and using the extended hydrodynamics models (see, e.g. [3]), an accurate and efficient solution of the Boltzmann equation itself remains elusive. The difficulty is the evaluation of the multidimensional collision integral whose straightforward computation involves $O(n^8)$ operations at every spatial cell, where n is the number of degrees of freedom in one velocity dimension. Various formulations have been proposed to expedite the evaluation of the collision integral. A good review of computational techniques for the solution of the Boltzmann equation can be found in [4]. In [5] Fourier transforms were used to construct fast evaluation algorithms of the collision integral. Tcheremissine in [6] used quasi-stochastic multidimensional numerical integration to achieve a high efficiency of the computations. However a deterministic solution of the Boltzmann equation in three dimensions for a broad range of Mach numbers is still beyond our reach.

We proposed an approach based on discontinuous Galerkin (DG) discretizations in the velocity variable for high-order deterministic solution of non-continuum flows. A key feature of our approach is that the collision operator takes the form of a bilinear operator with a time-independent kernel. This form of the collision operator can be found in the works of Struchtrup, e.g. [3], in the context of the method of moments. It has been recently used by Green and Vedula [7] for developing a lattice-Boltzmann method that uses the full collision integral and by Fox and Vedula [8] for constructing a closure in the method of moments. In our work, however, this form of the collision operator arises as a part of the procedure for the derivation of the governing equations for coefficients of the Galerkin discretization. Because our derivation is aimed at a high order high resolution Galerkin discretizations in the velocity space, it is designed to process large numbers of degrees of freedom in the velocity variable. This distinguishes our approach from that of Green and Vedula [7], where a very limited number of moments was used. The method proposed in the present paper appears to be a generalization of the symmetric formulation of the collision operator used in [6] where the author uses Korobov schemes for the evaluation of the multidimensional collision integral and achieves good efficiency. In our paper, however, integrals are evaluated deterministically using Gauss quadratures. An advantage of the present formulation as compared to the formulation of [6] is that the numerical quadratures do not employ values of the distribution function in post-collision velocities. This eliminates the need for sophisticated interpolation employed in [6]. In [5] a method for the deterministic evaluation of the collision operator is developed using Fourier transforms. Their method also has the benefit of not using the values of post-collision velocities in the evaluation of collision integral. Although the method in [5] is different from the one presented here, the total amount of integration is, however, comparable in both approaches. Results presented in this paper are obtained for the hard spheres collision model. However, any spherically symmetric collision potential can be easily implemented.

GALERKIN DISCRETIZATION IN THE VELOCITY SPACE

We assume that a rectangular parallelepiped is selected in the velocity space so that the contributions of the molecular distribution function to the first few moments are negligible outside of this parallelepiped. Let $K_j = [u^j_L, u^j_R] \times [v^j_L, v^j_R] \times [w^j_L, w^j_R]$, $j = 1, \ldots, M$ be a partition of the selected region into smaller parallelepipeds. To construct the basis functions on K_j, we consider the intervals $[u^j_L, u^j_R]$, $[v^j_L, v^j_R]$, and $[w^j_L, w^j_R]$ and introduce the nodes of the Gauss-Legendre quadratures on each of the intervals. Specifically, let these nodes be denoted as $\kappa^{j,u}_i$, $i = 1, s_u$, $\kappa^{j,v}_i$, $i = 1, s_v$, and $\kappa^{j,w}_i$, $i = 1, s_w$. The one-dimensional Lagrange basis functions are defined by

$$\phi^{j;u}_i(u) = \prod_{\substack{l=1,s^u \\ l\neq i}} \frac{\kappa^{j,u}_l - u}{\kappa^{j,u}_l - \kappa^{j,u}_i}, \quad \phi^{j;v}_i(v) = \prod_{\substack{m=1,s^v \\ m\neq i}} \frac{\kappa^{j,v}_m - v}{\kappa^{j,v}_m - \kappa^{j,v}_i}, \quad \phi^{j;w}_i(w) = \prod_{\substack{n=1,s^w \\ m\neq i}} \frac{\kappa^{j,w}_n - u}{\kappa^{j,w}_n - \kappa^{j,w}_i}. \tag{1}$$

The three-dimensional basis functions are given by $\phi^j_i(\vec{v}) = \phi^{j;u}_l(u)\phi^{j;v}_m(v)\phi^{j;w}_n(w)$, where $i = 1, \ldots, s = s_u s_v s_w$ is the index running through all combinations of l, n, and m. (In the implementation discussed in this paper, i is computed using the formula $i = (l-1)*s_v*s_w + (m-1)*s_w + n$.) We notice that the method can have different numbers of Gauss nodes used on each element K_j. However our implementation uses both uniform elements K_j and the same basis functions on each element. Although non-essential for the sake of the DG formulation, this uniformity allows us to achieve significant memory savings in the method which will be discussed later.

By replacing the integrals with Gauss quadratures and using the fact that the Gauss quadratures on s nodes are precise on polynomials of degree $2s-1$ it is straightforward to show that the basis functions have the following orthogonality properties:

$$\int_{K_j} \phi^j_p(\vec{v})\phi^j_q(\vec{v})\, d\vec{v} = \frac{\Delta\vec{v}^j}{8}\omega_p\delta_{pq}, \qquad \int_{K_j} \vec{v}\phi^j_p(\vec{v})\phi^j_q(\vec{v})\, d\vec{v} = \frac{\Delta\vec{v}^j}{8}\vec{v}^j_p\omega_p\delta_{pq}. \tag{2}$$

Here $\Delta\vec{v}^j = (u^j_R - u^j_L)(v^j_R - v^j_L)(w^j_R - w^j_L)$, $\omega_p := \omega^{s_u}_l \omega^{s_v}_m \omega^{s_w}_n$, and $\omega^{s_u}_l$, $\omega^{s_v}_m$, and $\omega^{s_w}_n$ are the weights of the Gauss quadratures with s_u, s_v, and s_w nodes, respectively. Also l, n, and m are the indices of the one-dimensional basis functions that make up the three-dimensional basis function $\phi^j_p(\vec{v}) = \phi^{j;u}_l(u)\phi^{j;v}_m(v)\phi^{j;w}_n(w)$. Finally, the vector $\vec{v}^j_p = (\kappa^{j,u}_l, \kappa^{j,v}_m, \kappa^{j,w}_n)$.

We assume that on each K_j the solution to the Boltzmann equation is sought in the form

$$f(t,\vec{x},\vec{v})|_{K_j} = \sum_{i=1,s} f_{i;j}(t,\vec{x})\phi^j_i(\vec{v}). \tag{3}$$

The DG velocity discretization follows by substituting the representation (3) into the Boltzmann equation and multiplying the result by a test basis function and integrating over K_j. Repeating this for all K_j and using the identities (2) we arrive at

$$\partial_t f_{i;j}(t,\vec{x}) + \vec{v}^j_i \cdot \nabla_x f_{i;j}(t,\vec{x}) = I_{\phi^j_i}, \tag{4}$$

where $I_{\phi^j_i}$ is the projection of the collision operator on the basis function $\phi^j_i(\vec{v})$ given by

$$I_{\phi^j_i} = \frac{8}{\omega_i \Delta\vec{v}^j} \int_{K_j} \phi^j_i(\vec{v}) \int_{R^3} \int_0^{2\pi} \int_0^{b_*} (f(t,\vec{x},\vec{v}')f(t,\vec{x},\vec{v}'_1) - f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1))|\vec{g}| b\, db\, d\varepsilon\, d\vec{v}_1\, d\vec{v}. \tag{5}$$

Here $g = \vec{v} - \vec{v}_1$, $\vec{v}'$ and $\vec{v}'_1$ are the post-collision velocities of the colliding pair of particles and b and ε are the collision parameters.

Notice that our velocity discretization is still incomplete because we have to specify how to evaluate the projection of the collision term. This is described in the next section. We want, however, to emphasize the simplicity of the obtained discrete velocity formulation. Indeed, the transport part of (4) has the complexity of a discrete ordinate formulation. A formulation with similar properties has been presented in [9]. Their Galerkin formulation, however, uses global basis functions of high order Hermite polynomials. Formulation (3) therefore generalizes their approach.

VELOCITY DISCRETIZATION OF THE COLLISION INTEGRAL

We will now introduce the formalism for evaluating (5). We notice that $\phi_i^j(\vec{v})$ can be extended by zero to the entire $\mathbb{R}^3$. Then

$$I_{\phi_i^j} = \frac{8}{\omega_i \Delta\vec{v}^j} \int_{R^3} \phi_i^j(\vec{v}) \int_{R^3} \int_0^{2\pi} \int_0^{b_*} (f(t,\vec{x},\vec{v}')f(t,\vec{x},\vec{v}_1') - f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1))|\vec{g}|b\,db\,d\varepsilon\,d\vec{v}_1\,d\vec{v}. \tag{6}$$

Using symmetry properties of the collision operator (e.g., [10]) we can replace the last expression with

$$I_{\phi_i^j} = \frac{8}{\omega_i \Delta\vec{v}^j} \int_{R^3} \int_{R^3} \frac{1}{2} \int_0^{2\pi} \int_0^{b_*} (\phi_i^j(\vec{v}') + \phi_i^j(\vec{v}_1') - \phi_i^j(\vec{v}) - \phi_i^j(\vec{v}_1)) f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1)|\vec{g}|b\,db\,d\varepsilon\,d\vec{v}_1\,d\vec{v}. \tag{7}$$

It follows from the assumptions underpinning the Boltzmann equation that values of $\vec{x}$ in $f(t,\vec{x},\vec{v})$ and $f(t,\vec{x},\vec{v}_1)$ can be considered independent of b and ε. Following an idea suggested in [4] (see also [3]) we remove $f(t,\vec{x},\vec{v})$ and $f(t,\vec{x},\vec{v}_1)$ from under the integrals in b and ε in (7) so that

$$\begin{aligned} I_{\phi_i^j} &= \frac{8}{\omega_i \Delta\vec{v}^j} \int_{R^3} \int_{R^3} f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1) \frac{|\vec{g}|}{2} \int_0^{2\pi} \int_0^{b_*} (\phi_i^j(\vec{v}') + \phi_i^j(\vec{v}_1') - \phi_i^j(\vec{v}) - \phi_i^j(\vec{v}_1)) b\,db\,d\varepsilon\,d\vec{v}_1\,d\vec{v} \\ &= \frac{8}{\omega_i \Delta\vec{v}^j} \int_{R^3} \int_{R^3} f(t,\vec{x},\vec{v})f(t,\vec{x},\vec{v}_1) A(\vec{v},\vec{v}_1;\phi_i^j)\,d\vec{v}_1\,d\vec{v}, \end{aligned} \tag{8}$$

where

$$A(\vec{v},\vec{v}_1;\phi_i^j) = \frac{|\vec{g}|}{2} \int_0^{2\pi} \int_0^{b_*} (\phi_i^j(\vec{v}') + \phi_i^j(\vec{v}_1') - \phi_i^j(\vec{v}) - \phi_i^j(\vec{v}_1)) b\,db\,d\varepsilon. \tag{9}$$

We notice that because $A(\vec{v},\vec{v}_1;\phi_i^j)$ is independent of time, it can be pre-computed and stored to be used in many individual simulations as long as the velocity discretization is the same. Integrals in the operator $A(\vec{v},\vec{v}_1;\phi_i^j)$ can be computed with good accuracy. Thus we can assume that the error of approximation of the collision integral is dominated by the truncation errors of the discretizations of the integrals in $\vec{v}_1$ and $\vec{v}$ in (8). Also, we notice that $A(\vec{v},\vec{v}_1;\phi_i^j)$ is symmetric in $\vec{v}$ and $\vec{v}_1$. This can be used to reduce the amount of summation by half in the evaluation of (8). The idea of changing the order of summation to extract a pre-computed collision matrix was used in [4] for discrete ordinate discretization of the Boltzmann equation. A form of the collision operator identical to (8) was used in the lattice Boltzmann solver in [8] in connection with the moment closure of the lattice method. In [3] (see also [7]) analytic expressions for the integrals (9) are presented in the case of polynomial kernels ϕ_i^j and the hard spheres collision model. In our DG formulation, however, functions ϕ_i^j are locally supported; therefore no exact formulas are expected. A symmetric form of the collision operator was used in [6]. Their formulation employs the formalism of Dirac delta-functions in a discrete ordinate approximation of the collision integral. We argue that the DG velocity discretization presents a unifying framework for both approaches in the sense that test and trial DG functional spaces can be constructed in such a way that the formulation of [6] will follow from the DG approximation by taking an appropriate limit.

IMPLEMENTATION AND EFFICIENCY

The numerical approximation of (8) follows by replacing the velocity distribution function with the DG approximation (3) and the integrals in (8) with Gauss quadratures using the nodes $\vec{v}_p^j$ introduced earlier. The resulting approximation of the collision operator takes the form

$$I_{\phi_i^j} = \frac{8}{\omega_i \Delta\vec{v}^j} \sum_{j^*=1}^{M} \sum_{i^*=1}^{s} \sum_{j'=1}^{M} \sum_{i'=1}^{s} f_{i^*;j^*}(t,\vec{x}) f_{i';j'}(t,\vec{x}) \frac{\Delta\vec{v}^{j^*}\omega_{i^*}}{8} \frac{\Delta\vec{v}^{j'}\omega_{i'}}{8} A(\vec{v}_{i^*}^{j^*}, \vec{v}_{i'}^{j'}; \phi_i^j). \tag{10}$$

The quantities $A_{i^* i' i}^{j^* j' j} = \frac{\Delta\vec{v}^{j^*}\omega_{i^*}}{8} \frac{\Delta\vec{v}^{j'}\omega_{i'}}{8} A(\vec{v}_{i^*}^{j^*}, \vec{v}_{i'}^{j'}; \phi_i^j)$ are independent of time and are pre-computed using adaptive quadratures based on Simpson's rule. Because the basis functions are discontinuous, the convergence rate for the adaptive quadrature rules is fairly slow. This is because the adaptive error control in the standard algorithm amounts to tracking the boundaries of support of the basis function using bisection. The authors are looking for more effective algorithms that can achieve accuracy less expensively. However, the evaluation of $A_{i^* i' i}^{j^* j' j}$ needs only be done once.

Table 1 illustrates the growth rate of the total number of non-zero entries in $A_{i^* i' i}^{j^* j' j}$ in the case of a finite domain in the velocity space. The top row gives the numbers of velocity cells M used in each velocity dimension. The same number of basis functions

TABLE 1. Non-zero components of $A^{j^* j' j}_{i^* i' i}$ for a single basis function ϕ^j_i and the estimated order of growth.

Number of cells, M	3	5	7	9	11
Number of components	39022	459455	2355130	8142006	21915065
Order	4.83	4.86	4.94	4.93	

($s=3$) were used in each cell in each velocity dimension. The middle row gives the number of non-zero records in $A^{j^* j' j}_{i^* i' i}$ for a single basis function. The last row gives the estimated order of the growth rate. One can see that the number of records grows approximately as $O(M^5)$. This is not at all surprising since each evaluation of the Boltzmann collision equation in its original form requires the knowledge of $f(t,\vec{x},\vec{v})$ in about n^5 velocity points, where n is the total number of velocity points in one dimension. Indeed, in this case the n^3 different selections of the second pre-collision velocity should be multiplied by about n^2 combinations of collision parameters to produce on the order of n^5 various combinations of $\vec{v}$, $\vec{v}_1$, $\vec{v}'$ and $\vec{v}'_1$ that need to averaged. We therefore believe that (8) maintains the same information as the direct discretization of the full collision operator.

In general one expects the total amount of storage for $A^{j^* j' j}_{i^* i' i}$ to be on the order of $O(n^8)$. However, this number can be dramatically reduced if one uses uniform partitions of the velocity domain, which includes both using the cells of the same size and the same basis functions on each cell. We recall that the operator $A(\vec{v}^{j^*}_{i^*}, \vec{v}^{j'}_{i'}; \phi^j_i)$ only depends on the discretization of the velocity domain and the collision model. One can show that in the case of spherically symmetric potentials and a single gas specie operator $A(\vec{v}^{j^*}_{i^*}, \vec{v}^{j'}_{i'}; \phi^j_i)$ is invariant under the following translation in the velocity space:

$$A(\vec{v}+\vec{\xi}, \vec{v}_1+\vec{\xi}; \phi^j_i(\vec{v}-\vec{\xi})) = A(\vec{v}, \vec{v}_1; \phi^j_i). \tag{11}$$

As a consequence the discrete operator $A^{j^* j' j}_{i^* i' i}$ is periodic on uniform partitions of the velocity domain. Indeed, if we know the values of $A^{j^* j' j}_{i^* i' i}$ for all basis functions on a single velocity cell, the values for the remaining basis functions can be recovered using (11). This reduces the storage to $s^3 O(n^5)$. Because the components of $A^{j^* j' j}_{i^* i' i}$ are independent of each other, their evaluation can be effectively parallelized using thousands of processors. In the simulations presented in this paper up to 320 processors were used with MPI parallelization algorithms. The cases of $s=3$, $M=3,5,7,11$ were considered. The total numbers of degrees of freedom in one velocity dimension were $n=9,15,21,33$, respectively. While the current implementation allows, in principal, to evaluate up to $n=100$ degrees of freedom in one dimension, this has not been done due to the limited available computer resources. It is estimated that for $M=33$, $n=99$ the evaluation of $A^{j^* j' j}_{i^* i' i}$ will take about a month on ten thousands average capacity processors. The estimated storage for the components of $A^{j^* j' j}_{i^* i' i}$ for $s=3$, $M=33$ is about two terabytes. Although it is expected to be on the order of tens of terabytes for higher order local polynomial approximation for the same total number of degrees of freedom. We would like to stress here that all this is well within the capacity of the modern high performance computers.

The number of multiplications and therefore the computational time to evaluate (10) is estimated to grow as $O(n^8)$. Times to compute (10) for a single time step on a single 2.3 MHz processor are presented in Table 2.

TABLE 2. Time in seconds to advance one temporal step.

Number of cells, M	3	5	7
Single time step, s	0.65	35.17	486.82
Order	7.81	7.81	

One can see that the evaluation times grow very fast with resolution. The authors are identifying ways to improve the calculation times including utilising different summation algorithms to minimize memory copying. However, even with the current quite inefficient summation procedures the method can give reasonable results on just a single processor for $n=21$. For $n=100$ about of 10^4 processors will be required. Because the method involves very little data exchange, we expect that parallelization of this algorithm will be very efficient.

THE MAXWELLIAN PERTURBATION DECOMPOSITION AND AUTOMATIC LINEARISATION

Efficiency of (10) can be improved using the fact that $I_\phi = 0$ when $f(t,\vec{x},\vec{v})$ is a Maxwellian distribution. In [6] it was proposed to replace $f(t,\vec{x},\vec{v})$ with

$$f(t,\vec{x},\vec{v}) = f_M(t,\vec{x},\vec{v}) + \Delta f(t,\vec{x},\vec{v}), \tag{12}$$

where $f_M(t,\vec{x},\vec{v})$ is selected from some suitable considerations. In the results presented below, the equilibrium distribution has the mass, bulk velocity and the temperature of local solution. Using the symmetry of $A(\vec{v},\vec{v}_1;\phi_i^j)$ we write

$$I_{\phi_i^j} = \frac{8}{\omega_i \Delta\vec{v}^j}\left[2\int_{R^3}\int_{R^3} f_M(t,\vec{x},\vec{v})\Delta f(t,\vec{x},\vec{v}_1)A(\vec{v},\vec{v}_1;\phi_i^j)d\vec{v}_1\,d\vec{v} + \int_{R^3}\int_{R^3}\Delta f(t,\vec{x},\vec{v})\Delta f(t,\vec{x},\vec{v}_1)A(\vec{v},\vec{v}_1;\phi_i^j)d\vec{v}_1\,d\vec{v}\right]. \qquad (13)$$

It is argued in [6], that when the solution is close to a Maxwellian, this decomposition allows to double the solution accuracy. Indeed, when the gas approaches equilibrium, we expect (10) to vanish. However, the truncation errors may be considerably large to prevent (10) from cancelling numerically. In Table 3 residuals in the evaluation of the collision integral are shown for a single velocity node near the origin. The distribution function corresponds to argon gas at equilibrium with density 6.634E−6 kg/m^3, temperature 210 K and zero bulk velocity. In all simulations the DG discretizations use $s = 3$ nodal points per dimension per cell. One can see that truncation errors in approximating (8) are relatively high even for $M = 11$ cells in each direction. At the same time, for $M = 11$, the macroparameters can be evaluated within six digits of accuracy. The corresponding perturbation $\Delta f(t,\vec{x},\vec{v})$, caused in this case by the errors in macroparameters, is very small. By using (13) truncation errors associated with the double integral in velocity space are effectively diminished by a factor proportional to the norm of $\Delta f(t,\vec{x},\vec{v})$. The term quadratic in $\Delta f(t,\vec{x},\vec{v})$ in this case is insignificant.

TABLE 3. Residuals in evaluating the collision integral for a Maxwellian distribution at a single velocity node.

Number of cells, M	3	5	7	11
Direct evaluation	0.247	0.075	0.017	0.002
Maxwellian perturbation	-0.0418	-9.4E-4	-4.7E-6	-7.9E-08

In the case when the solution is strongly deviating from the local Maxwellian, values of the collision integral obtained by the direct evaluation and by the perturbation approaches are in a good agreement. Table 4 summarizes these values for a single velocity node near the origin in the case of a constant distribution function with spherical support, shown in Figure 1a in the case of $M = 11$. The density, temperature and bulk velocity of this distribution function are the same as in the previous example. One can see that both methods produce comparable results. In particular the answers agree in three digits for $M = 11$. Notice, however, that the value of $\Delta f(t,\vec{x},\vec{v})$ is large in this case. The agreement is somewhat better if the non-equilibrium distributions are smooth and the truncation errors in the numerical integration are smaller. For smooth solutions, the use of the perturbation approach allows to improve mass, momentum and energy conservation.

TABLE 4. Values of the collision integral for a spherically supported constant distribution at a single velocity node.

Number of cells, M	3	5	7	11
Direct evaluation	4.2020	4.4735	4.9566	4.3567
Maxwellian perturbation	3.9628	4.4019	4.9393	4.3545

Finally, when the solution is approaching the local Maxwellian and $\Delta f(t,\vec{x},\vec{v})$ becomes small, simulations can be accelerated dramatically by dropping the quadratic term in (13). For as long as the local Maxwellian is not updated, the kernel of the Frechet derivative of the collision operator, $2\int_{R^3} f_M(t,\vec{x},\vec{v})A(\vec{v},\vec{v}_1;\phi_i^j)\,d\vec{v}$, can be stored and re-used. In this case the simulations are accelerated by a factor of n^2 per velocity node which results in a method of the total complexity of $O(n^6)$ operations per spatial cell. Implementations of both decomposition and linearization are simple and require only a minor code adjustment. In its present implementation, the solver switches automatically between the full non-linear, the decomposition and the linearised regimes based on the value of the L_1 norm of $\Delta f(t,\vec{x},\vec{v})$.

SPATIALLY HOMOGENEOUS RELAXATION

To illustrate the work of the algorithm above we present results of simulations of the relaxation of argon gas from perturbed states. Two cases are considered: relaxation of two equilibrium streams and relaxation of a discontinuous initial stage. The velocity domain is discretized using uniform partitions in both cases. The number of Gauss nodes in each cell in each dimension is the constant $s = 3$ in all simulations. Thus only the number of velocity cells M varies. The time discretization is by fifth order Adams-Bashforth method. The data for the Adams-Bashforth method is obtained using the fifth order Runge-Kutta method. In the first simulation, the initial data is given by a sum of two Maxwellian distributions with mass densities, bulk velocities and temperatures of $\rho_1 = 6.634\text{E}{-6}$ kg/m^3, $\vec{v}_1 = (967.78,0,0)$ m/s, and $T_1 = 300$ K and $\rho_2 = 1.99\text{E}{-5}$ kg/m^3, $\vec{v}_2 = (322.59,0,0)$ m/s, and $T_2 = 1100$ K correspondingly. In Figure 1b the cross section of the initial data by the plane $w = 0$ is shown. The spatially homogeneous relaxation is simulated for about 260 μs. The mean time between collisions for the steady state solution is estimated

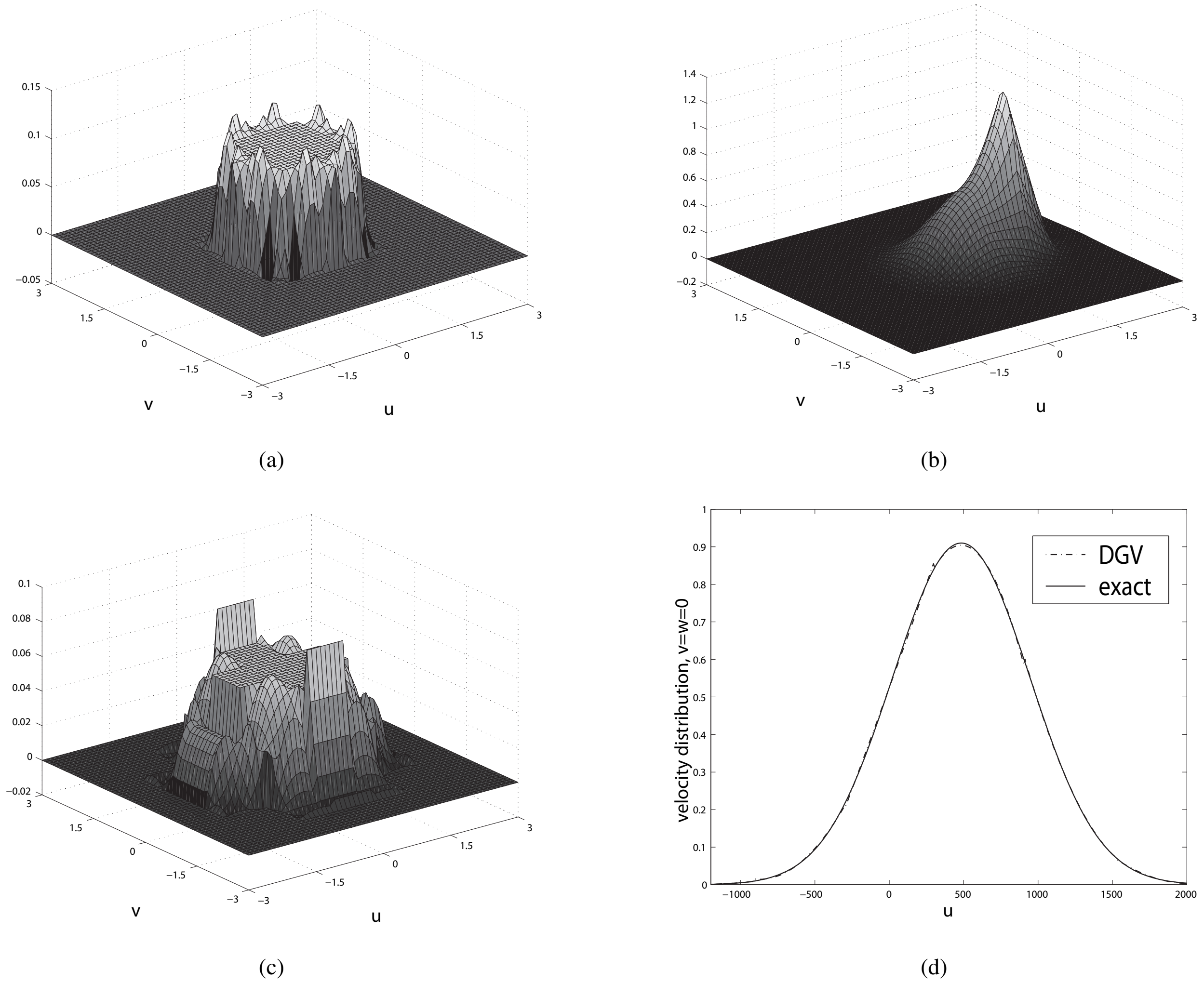
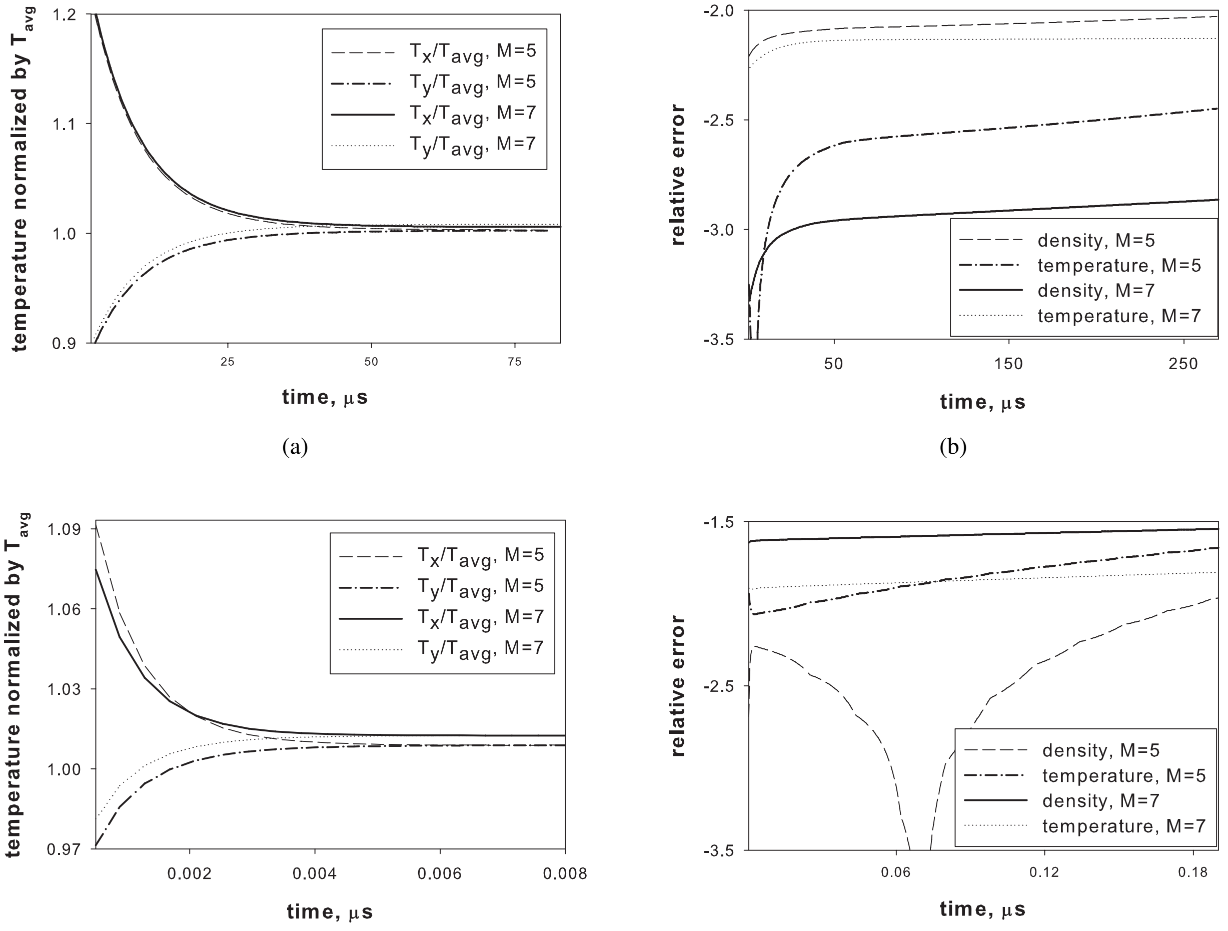


FIGURE 1. (a) Section $w = 0$ of the constant distribution function for $M = 11$ velocity cells. Artifacts at the edges of the discontinuous distribution are caused by insufficient resolution in the DG method. Mass, momentum and temperature are accurate to three digits. (b) Section $w = 0$ of the sum of two Maxwellian distributions in the case of $M = 7$ velocity cells. Mass and momentum of the approximate solution are accurate to three digits and temperature to two digits. (c) Section $w = 0$ of the sum of two constant distributions with spherical support in the case of $M = 7$ velocity cells. For the sum of constant distributions, mass and momentum of the discrete solution are accurate to two digits and temperature is accurate within 2%. The visible strong artifacts in the initial data are due to a coarse discretization. (d) Comparison of the steady-state DG velocity (DGV) solution to the exact solution in the case of mixing two Maxwellian streams. Jumps in the DGV solution at the velocity cell boundaries can be seen.

to be about 5.4 μs. In Figure 1d the cross-section of the steady state solution at $v = w = 0$ is compared to the exact Maxwellian distribution. Cross-sections for other velocity components show similar comparisons. One can see that the solutions coincide well, however, the DG velocity solution has noticeable jumps. In fact, these jumps are located on the boundaries of velocity cells and are expected in a DG method. As resolution increases, however, these jumps are expected to decrease in accordance with the order of the method. In Figure 2a the ratios of the directional temperatures T_x and T_y to the average temperature T_{avg} are shown. The value of the average temperature is calculated as $T_{\text{avg}} = 1/3T$ of the exact temperature. In particular, the value 1 of the ratio corresponds to a precise recovery of the directional temperatures in the equilibrium state. Simulations reach steady state at about 50 μs. One can see that simulations for $M = 5$ and $M = 7$ capture the correct directional temperature within 2% accuracy. Notice that conservation of mass and temperature are not explicitly enforced in the method. Rather, it is expected that by using higher order local DG basises one can conserve the first few moments within desired accuracy [11]. In Figure 2b logarithms to the base 10 of the relative error in conserving the density and temperature are shown. For $M = 5$ the density is accurate within two digits and for $M = 7$ within three digits. The temperature however is only conserved within two digits for both $M = 5$ and $M = 7$.



(a)

(b)

(c)

(d)

FIGURE 2. Relaxation of two Maxwellian streams (figures (a) and (b)) and relaxation of the sum of two constant distributions (figures (c) and (d)). (a) and (c) Graphs of T_x/T_{avg} and T_y/T_{avg} for numbers of velocity cells, $M = 5$ and $M = 7$. Here T_x and T_y are directional temperatures and $T_{\text{avg}} = T/3$, where T is the temperature of the exact solution. (b) and (d) Logarithmic plots of errors in mass and temperature for numbers of velocity cells, $M = 5$ and $M = 7$.

This is due to the fact that the resolution in velocity space is still insufficient to reach the asymptotic regime. Therefore, it is just a matter of coincidence that the temperature of the initial data is more accurately captured in the $M = 5$ case than in the more resolved $M = 7$ case. However, the graphs clearly show that the rate at which the density and the temperature are perturbed is slower for $M = 7$.

In the second simulation the initial data is discontinuous. Specifically, the initial distribution is a sum of two functions of the form

$$f_{\mathrm{X}}(\vec{v}) = \begin{cases} \rho h, & \text{if } |\vec{v} - \bar{\vec{v}}| \leq r, \\ 0, & \text{if } |\vec{v} - \bar{\vec{v}}| > r, \end{cases} \quad , \quad r = (T/5R)^{1/2}, \text{ and } h = (15\sqrt{5}/4\pi)(R/T)^{3/2}.$$

Here ρ, $\bar{\vec{v}}$ and T are given parameters. It is a straightforward exercise to verify that ρ, $\bar{\vec{v}}$ and T coincide with the mass density, bulk velocity and temperature of the distribution $f_{\mathrm{X}}(\vec{v})$, correspondingly. The gas is argon. The values of the macroparameters for the two artificial streams are $\rho_1 = 0.332$ kg/m^3, $\bar{\vec{v}}_1 = (106.0, 0, 0)$ m/s, $T_1 = 300$ K and $\rho_2 = 0.332$ kg/m^3, $\bar{\vec{v}}_2 = (-106.0, 0, 0)$ m/s, $T_2 = 300$ K. Graphs of two-dimensional cross-sections of the initial distribution by the plane $w = 0$ are given in Figure 1c for the case of $M = 7$. The mean time between molecular collisions is estimated about 0.39 ns. The simulations are performed for 190 ns. In Figure 2c the ratios of the directional temperatures to the average temperatures are presented. In this example, again, the approximation on the $M = 5$ cells is slightly better than the one on $M = 7$ cells. However in both cases the accuracy is within 2%. In Figure 2d the conservation of density and temperature are given. One can see that most of the error occurs at the initial stages, when the solution is discontinuous. As the solution evolves toward a Maxwellian and becomes smoother, the errors of the evaluation of the collision integral grow at a slower rate. The solver switches to the decomposition regime after just a few time steps and to the linearised regime at about 2 ns. Although the solution is still far from the local Maxwellian at 2 ns, the mass, momentum and temperature are conserved quite satisfactory in the linearised regime. However, in this case, the simulations are at least one hundred times faster than in the full non-linear regime for $M = 7$.

CONCLUSION

We applied the discontinuous Galerkin approach to the discretization of the Boltzmann equation in the velocity space using the symmetric bilinear form of the Galerkin projection of the collision operator. The time-independent kernel of the bilinear operator carries the information about the geometry of the velocity discretization and about the collision model. Evaluation of the kernel is implemented using an MPI parallelization algorithm that scales to a large number of processors. Algorithms for automatic linearisation and perturbation decomposition were proposed and implemented in the method. The discontinuous Galerkin approach was applied to the problem of spatially homogeneous relaxation. Discretizations with $n = 9$, 15 and 21 degrees of freedom per velocity dimension were tested. The method shows good conservation properties in both non-linear and linearised regimes. One obstacle to increasing the velocity resolution is that the amount of storage for the collision kernel and the number of arithmetic operations to evaluate the collision integral in the fully nonlinear regime grow as $O(n^5)$ and $O(n^8)$, respectively. The authors are developing algorithms for an efficient parallelization of evaluation of the collision integral to up to tens of thousands of processors. This should allow for practical computations for $n = 100$ degrees of freedom per velocity dimension using modern high performance computers. There are several unique features of the new method that were not fully explored in the present work. Galerkin techniques are known to give the best results when the approximation spaces are chosen to fit the problem. In the present implementation of the method, however, off-shelf basis functions were used. It will therefore be interesting to explore the bilinear form of (8) to study the properties of the collision operator and find approximation spaces that are mimicking analytical and statistical properties of the problem. Because of the very close connection of Galerkin methods to the method of moments both approaches will benefit from the analysis of (8). This will be the authors' future work.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the NRC Resident Research Associateship Program and the U.S. Air Force Office of Scientific Research monitored by F. Fahroo.

REFERENCES

1. G. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, 1994.
2. M. I. Gallis, and J. R. Torczynski, *Phys. Fluids* **23**, 030601 (2011).
3. H. Struchtrup, *Macroscopic Transport Equations for Rarefied Gas Flows Approximation Methods in Kinetic Theory*, Interaction of Mechanics and Mathematics Series, Springer, Heidelberg, 2005.
4. V. Aristov, *Direct Methods for Solving the Boltzmann Equation and Study of Nonequilibrium Flows*, Fluid Mechanics and Its Applications, Kluwer Academic Publishers, c, 2001.
5. A. V. Bobylev, and S. Rjasanow, *Eur. J. Mech. B/Fluids* **18**, 869–887 (1999).
6. F. G. Tcheremissine, *Comp. Math. Math. Phys.* **46**, 315–329 (2006).
7. B. I. Green, and P. Vedula, "Validation of a collisional Lattice Boltzmann Method," in *20th AIAA Computational Fluid Dynamics Conference, 27-30 June 2011, Honolulu Hawaii*, AIP Conference Proceedings 3403, American Institute of Physics, 2011, p. 14.
8. R. O. Fox, and P. Vedula, *Ind. Eng. Chem. Res.* **49**, 5174–5187 (2010).
9. M. K. Gobbert, and T. S. Cale, *J. Sci. Comput.* **30**, 237–273 (2007).
10. M. Kogan, *Rarefied Gas Dynamics*, Plenum Press, 1969.
11. A. Alekseenko, N. Gimelshein, and S. Gimelshein, *Int. J. Comput. Fluid D.* **to appear** (2012).