Hypersonic Flow over a Wedge with a Particle Flux Method

M. N. Macrossan*, M. V. Metchnik† and P. A. Pinto†

* *Centre for Hypersonics, School of Engineering, University of Queensland, St Lucia, 4072, Australia*
†*Steward Observatory, University of Arizona, 933 N Cherry Av, Tucson, AZ 85722-0065, USA*

Abstract. We have investigated the use of DSMC as a pseudo-Euler solver in the continuum limit by using a modification of Pullin's Equilibrium Particle Simulation Method (EPSM). EPSM is a particle-based method which is in effect the large collision rate limit of DSMC yet requires far less computational effort. We propose a modification of EPSM, the Particle Flux Method (PFM), which is intermediate between EPSM and a conventional finite volume continuum flow solver. The total mass, momentum and energy in each cell are stored. Flux particles are created at every time step and move in free flight over a short decoupling time step, carrying mass momentum and energy between cells. The new method has been demonstrated by calculating the hypersonic flow over a wedge, for which DSMC calculations are available. Because of an inherent dissipation, related to the cell size and time step, the shock was thicker than that found in the DSMC calculations, but the shock location was the same. PFM is not prohibitively expensive and may have some advantages over conventional continuum based flow solvers, in terms of robustness arising from its firm basis in the physics of molecular flow.

DSMC IN THE CONTINUUM LIMIT

At the RGD Symposium in 1980 two papers [1, 2] were presented which used DSMC in the near-continuum regime, where the flow is collision dominated. The cell size and decoupling time step were larger than the mean free path and mean collision time corresponding to the flow density. Thus a typical simulator particle would undergo many collisions at every time step, and the distribution of molecular velocities in each cell would be Maxwellian. In Ref. [2] the computational efficiency was improved by using a *collision limiting* procedure in which, after enough collisions were calculated in each cell to establish equilibrium, no more collisions were calculated in that cell for that time step; the further collisions can have no statistical effect on the resulting distribution of velocities.

Since the Euler equations can be derived from the Boltzmann equation under the assumption of an equilibrium distribution function, it was the aim of all these methods to produce a solution of the Euler equations. However, as noted by Merkle *et al.*[1] the finite decoupling interval and cell size give rise to an inherent dissipation. This dissipation is related to the mean distance simulator particles travel in each time step (their effective mean free path) [3, 4]. Alternatively, one can consider the distribution of velocities which determines the net fluxes between cells, which is an amalgam of truncated equilibrium distributions from which the viscous stress tensor and heat flux vector can be evaluated [5].

Equilibrium Particle Simulation Method

In the same year Pullin [6] published his infinite collision rate (or equilibrium) limit of DSMC, the Equilibrium Particle Simulation Method (EPSM). Pullin noted that in this limit it was not necessary to calculate collisions at all. Since the effect of collisions at each time step would be to establish an equilibrium distribution of velocities, with a mean and variance determined by the total momentum and energy of the particles in the cell, new velocities for the simulator particles can be generated statistically. That is, the collision phase of DSMC was replaced by a redistribution phase, in which new velocities for all the particles in a cell were selected from an equilibrium distribution. The mean and variance of the new particle velocities must match the values they had before the redistribution. Pullin [7] gives a method for generating the random set of velocity components with a specified mean and variance (see also Ref. [3] for Pullin's algorithm); another method is given in Ref. [8].

It is possible in EPSM to store separately the three components of velocity and each internal energy mode for each particle, as in DSMC. However, Pullin [6] noted that in 1D and 2D flows only 1 and 2 components (respectively) of molecular velocity need to be stored for each simulation particle. The thermal energy of any missing velocity component can be combined with molecular structure energy, such as rotational energy, and the mean value of this combined ‘internal energy’, appropriate to the equilibrium cell temperature, is stored. This tends to reduce the statistical scatter in the resulting flow temperature.

THE PARTICLE FLUX METHOD

Here we develop EPSM to improve its computational efficiency and reduce its statistical scatter. The resulting method is similar to a conventional finite-volume continuum flow solver with explicit time-stepping, in that the total mass, momentum and energy in each cell are stored and the cells exchange fluxes of mass, momentum and energy with neighboring cells. The flow state (density, mean velocity and temperature) is also stored for each cell.

The fluxes are carried between cells by simulator particles and we refer to this method as the *Particle Flux Method* (PFM). PFM differs from DSMC and EPSM in that the flux particles are continuously created as they are needed and discarded after they are moved. At each time step, in each cell, N_f flux particles, are created with positions uniformly distributed over the cell volume. These flux particles each represent a fraction $1/N_f$ of the current total mass in the cell. The flux particles are given velocities selected from an equilibrium distribution with a mean equal to the known flow velocity in the cell and a variance RT_e, where R is the ordinary gas constant and T_e is the (equilibrium) cell temperature. The flux particles then move in free molecular flight for the time Δt. If a particle moves from one cell to another its mass, momentum and energy are deducted from its old cell and added to its new cell. Once this is done the particle is discarded. The number of flux particles could vary from cell to cell, but it is convenient to use the same number N_f for each cell. The computer memory required for the particles (a few hundred particles) is much less than that required for particles in DSMC or EPSM.

Because the same number of flux particles is used for every cell, the statistical scatter is independent of the flow density, a significant difference from EPSM and DSMC. The greatest advantage of using the same number of flux particles in each cell is that it is not necessary to generate a new set of random velocities in every cell at every time step. A single set of N_f thermal velocities (with zero mean and unit variance) can be generated at each time step. For each cell these thermal velocities are scaled by the standard deviation $(RT_e)^{1/2}$ and are added to the mean cell velocity to produce the individual flux-particle velocities. In the calculations reported here there are approximately 50,000 cells so that the number of random thermal velocities which must be generated is reduced by a factor of 50,000, a significant saving in CPU time. In a typical EPSM (or DSMC) calculation on this same grid there might be about 20 particles per cell for a total of $\sim 10^6$ simulator particles and this number of new velocities would be generated at each time step. Although PFM is much less CPU intensive than EPSM, which in turn is much less CPU intensive than collision-limited DSMC, it is more CPU intensive than a conventional continuum Euler solver.

Here we restrict the time step in PFM so that no particle travels more than one cell width in one time step, and fluxes flow between immediately adjacent cells only. The particles travel in all directions so that each cell may exchange fluxes directly with *any* neighboring cell; unlike conventional continuum solvers the flux calculations are not 1D procedures which are applied independently in each direction on the grid. Because the flux particles are uniformly distributed across each cell when created, some density gradient information is lost in PFM.

FLOW OVER A WEDGE

PFM was used to calculate the hypersonic flow of argon ($\gamma = 5/3$) over a 46° and 42° wedge with freestream Mach number ($M_\infty = 15.64$) matching the DSMC and Navier-Stokes calculations of Bondar *et al.* [9]. A specularly reflecting surface boundary condition was used. The grid for the 46° wedge was 245 $\times$ 196 and that for the 42° wedge was 245 $\times$ 217. In each case Δx was $L/98$ and $\Delta y/\Delta x$ was equal to the tangent of the wedge half-angle δ_w. The flow begin with an impulsive start and the statistical scatter was reduced by accumulating flow samples after steady state was reached

Fig. 1 shows the time variation of shock stand-off distance on the stagnation streamline, and the temperature in a test cell adjacent to the sharp corner of the wedge. These show that steady state cannot safely be assumed until an elapsed time of $tV_\infty/L > 30$, where V_∞ is the freestream flow speed and L is the length of the wedge in the x-direction. In order to reduce the computational effort, the calculation began with $N_f = 20$ particles, which was increased in steps until time $tV_\infty/L = 30$. Then N_f remained constant while flow samples were accumulated until a elapsed simulation

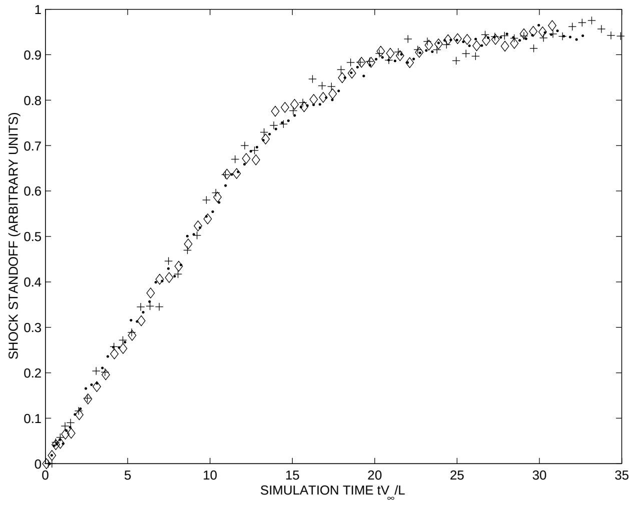

FIGURE 1. Unsteady development of the flow (46° wedge) shown by (a) the shock stand-off distance and (b) the temperature in a cell adjacent to the top corner of the wedge. •, $N_f = 236$; ⋄, $N_f = 236$; +, $N_f = 100$.

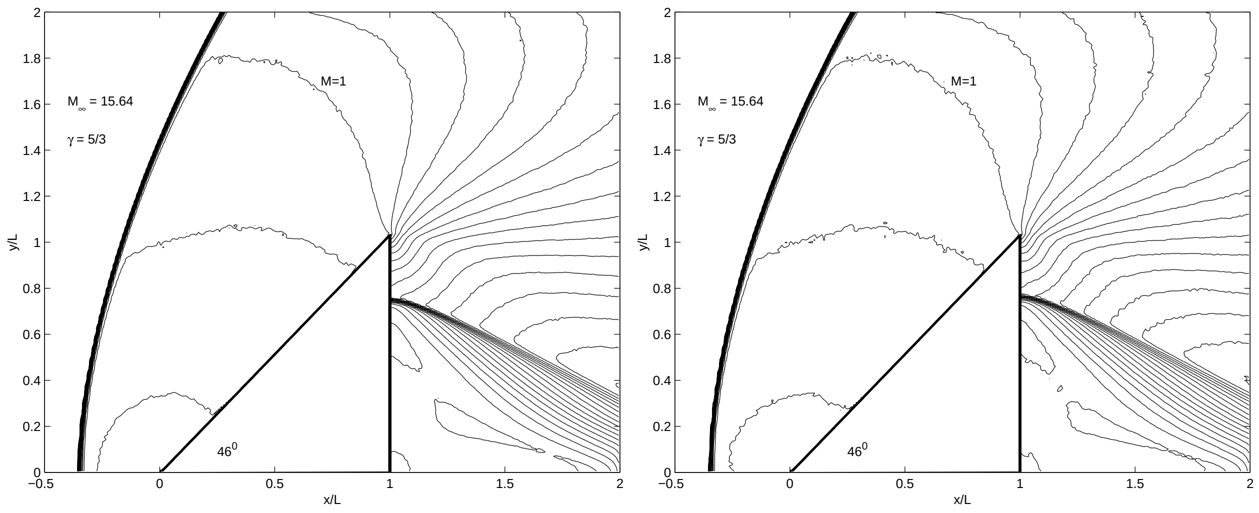

FIGURE 2. Mach number contours ($\delta M = 0.3$) for flow over a 46° wedge, calculated with the particle flux method with different maximum numbers of flux particles. (a) $N_f = 236$, (b) $N_f = 100$. Grid 245 × 196.

time $tV_\infty/L = 35$. Fig. 2 shows the time-averaged flow calculated with $N_f = 236$ flux particles, and also that calculated with $N_f = 100$. The later case required less CPU time and produced results virtually identical to those for the larger number of flux particles.

Fig. 3 shows the stagnation streamline density profile compared with the DSMC results of Bondar *et al.* [9]. In a PFM calculation scales with the cell size; the non-equilibrium nature of the shock and its physical correct structure cannot be captured. However, the figure shows that the location of the shock, the shock-stand off distance, is in good agreement with the DSMC calculations. Fig. 4 compares the flow over the 46° wedge with that over a 42° wedge. The figure caption gives the shock standoff distance for both flows, determined from the point where the flow density was mid-way between its freestream value and the theoretical value behind a normal shock.

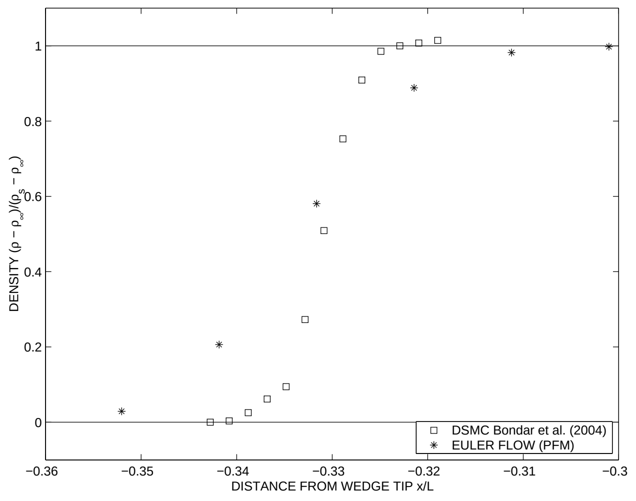

FIGURE 3. Normalized density profile $(\rho - \rho_\infty)/(\rho_s - \rho_\infty)$ along stagnation line for the 46° wedge flow. Euler flow calculated with PFM ($N_f = 236$) compared with the DSMC calculations of [9]; ρ_s is the theoretical density behind a normal shock. $M_\infty = 15.64$, $\gamma = 5/3$. The shock's thickness for PFM is too large but its location agrees with the DSMC simulation.

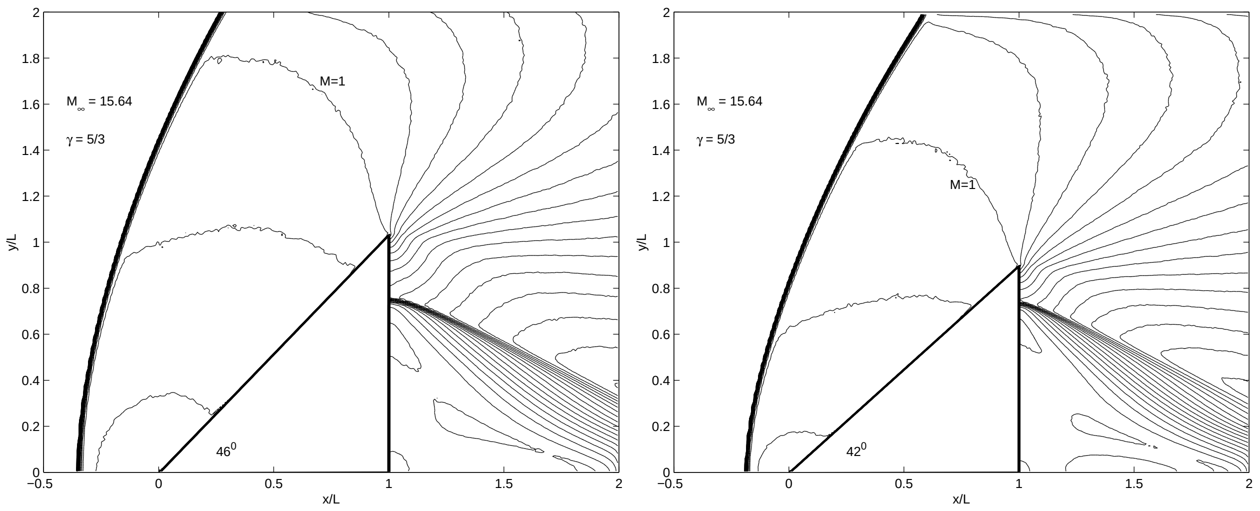

FIGURE 4. Mach number contours ($\delta M = 0.3$) for flow over a wedge. $N_f = 236$. Time average for $30 < tV_\infty/L < 35$. (a) Wedge angle $\delta_w = 46°$. Grid 245 × 196. Shock-standoff $\Delta/L = 0.334$, $\cos\delta_w\,(\Delta/L) = 0.232$ (b) $\delta_w = 42°$. Grid 245 × 217. $\Delta/L = 0.175$, $\cos\delta_w\,(\Delta/L) = 0.130$

Weighted mass particles

A variation of PFM was investigated in which weighted mass particles were used in every cell at every time step. Thermal velocity space was discretized and each particle was assigned a weight w corresponding to a small range of the Maxwell distribution and its velocity was an average of this range of velocities. A radially symmetric set of these thermal velocities is shown in Fig. 5 for the first quadrant of the thermal velocity plane c_x-c_y. Each quadrant in the thermal velocity plane was a mirror image of that shown in Fig. 5 giving a total of 236 velocities. The thermal speeds and weights are shown in Table 1.

Any number of sets of such velocity points, and corresponding weights, could be constructed. A measure of the goodness of a set of thermal velocities and weights is given by evaluating the flux-related integrals

$$\int_0^\infty c_x dc_x \approx \sum wc_x, \quad \int_{-\infty}^\infty c_x^2 dc_x \approx \sum wc_x^2, \quad \int c_x^3 dc_x \approx \sum wc_x^3 \text{ and } \quad \int_0^\infty \left(c_x^2 + c_y^2\right)^{1/2} dc \approx \sum w\left(c_x^2 + c_y^2\right)^{1/2}.$$

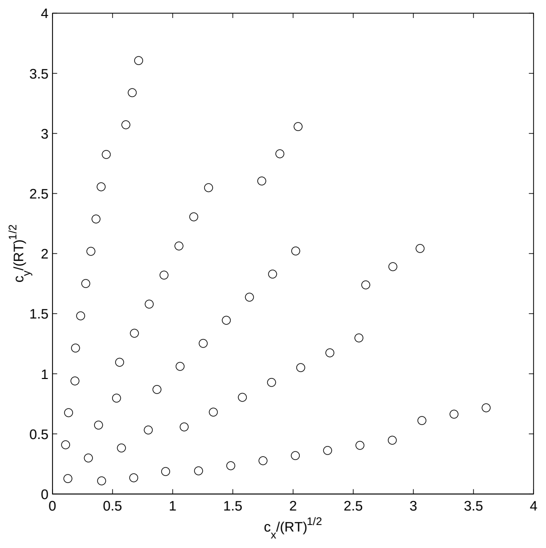

FIGURE 5. Thermal velocity points (first quadrant only shown). The relative weights of the points depend on the magnitude of the thermal speed and the spacing of the points (see Table 1).

TABLE 1. Fractional (weighted) masses corresponding to Fig. 5. The mass of each flux particle is $w\rho V$, where V is the cell volume.

$c/(RT_e)^{1/2}$	0.1819	0.4235	0.6895	0.9590	1.2297	1.5010	1.7725
# points	4	12	16	16	20	20	20
$w \times 10^3$	9.1339	8.4830	9.1464	10.2579	7.8500	6.6296	5.028
$c/(RT_e)^{1/2}$	2.0442	2.3160	2.8598	3.1318	3.1318	3.4039	3.6760
# points	20	20	20	20	16	16	16
$w \times 10^3$	3.4578	2.1692	1.2463	0.6577	0.3993	0.1787	0.0738

The errors in these integrals, compared with the theoretical values, were in this case 0.8%, 0, -0.6% and 0.2% respectively. Note that the velocities and weights have been chosen so that the mean velocity is zero and the second integral, $\int c_x^2 dc_x = \int c_y^2 dc_y = RT_e$ is exact. These same conditions are satisfied by the random thermal velocities that are generated once every time step in the version of PFM described above. The x-velocity of a flux particle in a cell is then given by $v_x = u_x + (RT_e)^{1/2} c_x$, where u_x and T_e are the flow velocity and temperature in the cell.

The flow calculated using these weighted velocities is shown in Fig. 6. There is a significant difference in the Mach number contours, particularly near the shock, compared with using new random thermal velocities at each time step. It appears therefore that the small errors in the representation of the Maxwell distribution accumulate during the simulation.

The CPU times are shown in Table 2, which show that the weighted particle method is the slowest. This was partly because it required a greater number of time steps but mainly because 236 particles were used throughout the entire computation. For the random velocity method N_f was significantly smaller during the flow development. We have not investigated whether a smaller number of weighted particles with a better choice of fixed velocities and weights can give results of similar quality to those obtained with random velocities.

TABLE 2. CPU time for PFM. Simulation time $t/V_\infty/L = 35$. Intel XEON 2GHz processor.

Wedge angle	N_f	Total # time steps	Thermal velocities	Particles	CPU time (s)
46°	100	6,670	random	equal masses	6,850
46°	236	6,803	random	equal masses	15,032
46°	236	7,503	fixed	weighted masses	24,160
42°	236	7,259	random	equal masses	22,325

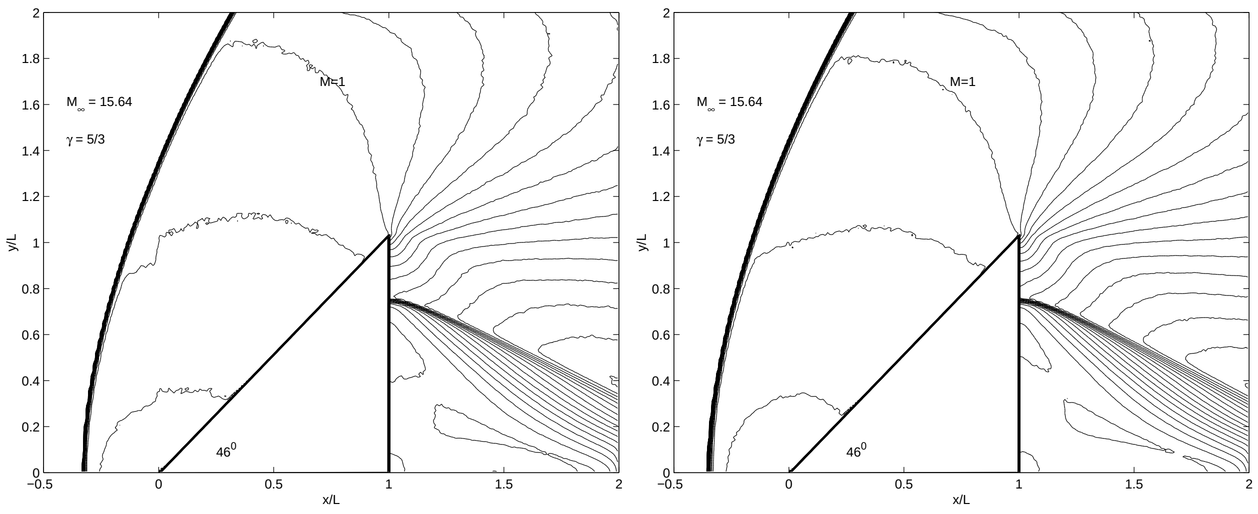

FIGURE 6. (a) Flow results for 236 weighted particles/cell with fixed velocities compared with (b) the standard case using 236 random thermal velocities/cell at each step (as in Fig. 2).

DISCUSSION AND CONCLUSIONS

We have investigated the use of DSMC as a pseudo-Euler solver in the continuum limit by using a modification of Pullin's Equilibrium Particle Simulation Method (EPSM) [6]. EPSM is a particle based method which is the large collision rate limit of DSMC. EPSM requires far less computational effort than DSMC but is nevertheless still an expensive method compared to conventional continuum Euler solvers. We proposed a modification of EPSM, the Particle Flux Method, which is intermediate between EPSM and a conventional finite volume continuum flow solver. Fluxes of mass, momentum and energy are carried between cells by continually created particles in free flight.

The new method has been demonstrated by calculating the hypersonic flow over a wedge. The location of the bow shock agreed with the DSMC calculations of Ref. [9]. PFM and EPSM (and DSMC if the time step is greater than the mean collision time) display an inherent dissipation which arises from the free flight of the particles for an effective mean time between collisions equal to the time step. Hence the shock calculated with PFM was much thicker than that found in the DSMC calculations. The use of 100 flux particles/cell gave a satisfactory resolution of the flow. The CPU time of less than 2 hours was not prohibitive. PFM may have some advantages over conventional continuum based flow solvers, in terms of robustness arising from its firm basis in the physics of molecular flow.

REFERENCES

1. Merkle, C. L., Behrens, H. W., and Hughes, R. D., "Application of the Monte-Carlo Simulation Procedure in the Near Continuum Regime," in *Rarefied Gas Dynamics*, edited by S. S. Fisher, Prog. Astro. Aero. v74, AIAA, New York, 1981, pp. 256–268.
2. Lengrand, J. C., Raffin, M., and Allègre, J., "Monte-Carlo Simulation Method Applied to Jet Wall Interactions under Continuum Flow Conditions," in *Rarefied Gas Dynamics*, edited by S. S. Fisher, Prog. Astro. Aero. v74, AIAA, New York, 1981, pp. 994–1006.
3. Macrossan, M. N., Some developments of the equilibrium particle simulation method for the direct simulation of compressible flows, NASA Contractor Report CR 198175, Institute for Computer Applications in Science and Engineering, N.A.S.A. Langley Research Center, Hampton, Virginia (1995).
4. Macrossan, M. N., *J. Comput. Phys.*, **80**, 204–231 (1989).
5. Macrossan, M. N., and Oliver, R. I., *Int. J. Numer. Meth. Fluids*, **17**, 177–193 (1993).
6. Pullin, D. I., *J. Comput. Phys.*, **34**, 231–244 (1980).
7. Pullin, D. I., *J. Statist. Comput. Simul.*, **9**, 303–309 (1979).
8. Macrossan, M. N., "A particle simulation method for the BGK equation," in *Am. Inst. Phys. Conf. Proc. v. 585*, edited by T. Bartel and M. Gallis, (Rarefied Gas Dynamics 22), New York, 2001, pp. 426–433.
9. Bondar, Y. A., Markelov, G. N., Gimelshein, S. F., and Ivanov, M. S., AIAA Paper 2004-1183 (2004).