39th AIAA Thermophysics Conference
25 - 28 June 2007, Miami, FL

AIAA 2007-4259

A Low Diffusion Particle Method for Simulating Compressible Inviscid Flows

Jonathan M. Burt[1] and Iain D. Boyd[2]
Department of Aerospace Engineering
University of Michigan, Ann Arbor, MI 48109

A new particle method is presented for the numerical simulation of compressible inviscid gas flows, through procedures which involve relatively small modifications to an existing direct simulation Monte Carlo (DSMC) algorithm. Implementation steps are outlined for simulations involving various grid geometries and for gas mixtures comprising an arbitrary number of species. The proposed method is compared with other numerical schemes through a series of one-dimensional and two-dimensional test cases, and is shown to provide a significant reduction in both artificial diffusion and statistical scatter effects relative to existing DSMC-based equilibrium particle methods.

I. Introduction

THE compressible Euler equations are regarded as the governing equations for a wide variety of gas dynamics problems, and numerical simulation schemes based on these equations are commonly used in a broad range of aerospace applications. Most of these schemes may be categorized as traditional computational fluid dynamics (CFD) methods, and involve direct numerical solution procedures for the governing equations through spatial discretization of various quantities over a computational grid.[1] Another category of numerical schemes use a molecular level approach more closely tied to kinetic theory, and model the gas using a large collection of representative particles. These "equilibrium" particle methods may be considered an extension for inviscid gas flows to the direct simulation Monte Carlo (DSMC) method of Bird.[2]

The DSMC method is generally recognized as the most mature and most frequently used simulation method for gas flows involving significant rarefaction and translational nonequilibrium effects. In a DSMC calculation, simulated particles, each representing a large number of real atoms or molecules, are moved through a computational grid and periodically collided following phenomenological molecular interaction models. The gas velocity distribution is represented here by the distribution of particle velocities within a cell, and moments of this distribution may be used to calculate bulk flow properties. Simulation results are then determined by averaging various properties among all particles in each cell, typically with further averaging over a large number of time steps. This provides a relatively simple and efficient alternative to direct numerical solutions to the Boltzmann equation for rarefied gas flow simulation, but cell size and time step limitations make the method prohibitively expensive for simulating most continuum flows of practical engineering interest.

Due to these issues of computational expense, the DSMC method is used almost exclusively to simulate rarefied flows for which the continuum assumptions underlying most CFD methods fail. In contrast, equilibrium particle methods are intended specifically for continuum flow simulation, and differ from DSMC primarily in procedures for updating particle velocities during each time step. In place of simulated binary collisions involving some fraction of particles in each cell, all particles may experience multiple collisions per time step – as in DSMC collision limiter schemes[3-5] – or collision procedures may be replaced by a resampling step, where new particle velocities are sampled from a Maxwellian distribution in such a way that total momentum and energy are conserved.[6,7] By enforcing local thermal equilibrium in every cell during each time step, these methods effectively reproduce characteristics of the Boltzmann equation at the equilibrium limit, and are assumed to provide results equivalent to a numerical solution of the compressible Euler equations.

Much of the interest in equilibrium particle methods has grown out of a desire for algorithms that can be accurately and efficiently applied to gas flows involving a wide range of Knudsen number regimes, as may occur in engineering problems involving small length scales, low density or high speed flows. These methods offer an

[1] Post-doctoral research fellow, AIAA member.
[2] Professor, AIAA associate fellow.

important advantage over traditional CFD methods for use in simulating such flows, as implementation in a hybrid code with the DSMC method is potentially far simpler than development of a hybrid CFD-DSMC code.[8] (In such a hybrid code, DSMC is used in nonequilibrium regions of the flowfield and an alternate method is used in continuum regions.) The similar use of information transport by particle advection between DSMC and equilibrium particle methods also allows for much stronger coupling between continuum and nonequilibrium regions than may be possible in a hybrid CFD-DSMC algorithm.[9] This characteristic of equilibrium particle methods should make them particularly useful for simulating unsteady flows which involve a wide range of Knudsen number regimes.

While a number of published equilibrium particle methods have shown promise for compressible inviscid flow simulation,[3-7] these methods are in general considerably less practical or desirable than CFD methods commonly used to simulate this type of flow. One main problem with existing equilibrium particle methods is their tendency to suffer from large numerical diffusion errors, including effects of artificial viscosity, thermal conductivity, and mass diffusion. These errors result from the free-molecular transport of mass, momentum and energy between adjacent cells, and have been analyzed in detail by Macrossan[10] and Breuer et al.[5] (The analysis of Macrossan is performed for the equilibrium flux CFD method of Pullin,[6] but is applicable to equilibrium particle methods as well.) As the artificial transport coefficients in these methods tend to scale linearly with cell size, a prohibitive number of cells may be required to reach an overall level of simulation accuracy comparable to that of CFD methods on a much coarser grid.

Another disadvantage of existing particle methods is that, under most conditions, they tend to be far more computationally expensive than many CFD schemes. This is due in part to effects of statistical scatter, which often require that results be averaged over a very large number of sampling time steps.[2] The prevalence of scatter also makes these methods poorly suited to unsteady flow simulation, as adequate precision in simulation results may force the user to employ ensemble averaging over multiple independent simulations, or to use a very large number of particles per cell.

In the following sections, a new particle method is proposed as an alternative means of simulating compressible inviscid flows. The method presented here is intended as an improvement over existing equilibrium particle methods, by significantly reducing both numerical diffusion effects and statistical scatter. Following a detailed description of simulation procedures, a series of test cases is used to demonstrate these properties and to evaluate the overall accuracy of the new method.

As a first test case, we consider a one-dimensional shock tube problem, and compare results from a simulation using the new method to results from two existing particle methods and an exact Riemann solver. In a modification to this flow, we then more clearly show the potential benefits of the new method by isolating the diffusive fluxes and comparing results among the different particle methods. A more complicated two-dimensional test case is then considered, where results are compared between simulations using the new method, an existing particle method, and a traditional CFD method. Finally, results are summarized and ideas for future work are presented.

II. Description of the Method

As with other equilibrium particle methods, numerical procedures in the proposed method are similar to those in the DSMC technique. The computational domain is divided into a large number of grid cells, and representative particles are tracked through these cells. During each time step, particles are moved through the grid according to assigned velocities, and bulk flow properties are computed by averaging quantities over all particles in each cell. The main difference between this new method and DSMC is in the procedures by which velocities used for particle movement are updated during each time step. Here standard DSMC collision routines are replaced by an alternate set of procedures, as described below.

A. Determination of cell quantities

In addition to standard DSMC values for position $\mathbf{X}_i$, velocity $\mathbf{U}_i$ and species number, particles here carry a bulk velocity vector $\mathbf{U}_{b,i}$ and a temperature T_i. As a first step in procedures to modify $\mathbf{U}_i$ values for subsequent use in particle movement, a number of cell-averaged quantities are computed and stored for each cell. These quantities are evaluated for a given cell during each simulation time step, and constitute instantaneous cell properties in the sense that no averaging is performed over multiple time steps. One vector and three scalar quantities are assigned to each cell: $\mathbf{U}_{cell}$ is the mass-averaged bulk velocity, taken as a mass-weighted average over particle bulk velocities $\mathbf{U}_{b,i}$; ζ is the average number of internal degrees of freedom among all particles in a cell; ρ is the mass density; and β is the thermal speed scale for the cell, defined as the inverse of the most probable thermal speed for a Maxwellian velocity distribution in a simple gas at the cell temperature T_{cell}. The thermal speed scale β is calculated as

$$\beta = \sqrt{\frac{\langle \mathrm{MW_i} \rangle}{2 Ru\, \mathrm{T_{cell}}}} \tag{1}$$

where

$$\mathrm{T_{cell}} = \langle \mathrm{T_i} \rangle + \frac{\langle \mathrm{MW_i} \rangle}{(3+\varsigma) Ru} \left(\frac{\mathrm{N_p}}{\mathrm{N_p}-1} \right) \left(\langle \mathbf{U}_{\mathrm{b,i}} \cdot \mathbf{U}_{\mathrm{b,i}} \rangle_{\mathrm{m}} - \langle \mathbf{U}_{\mathrm{b,i}} \rangle_{\mathrm{m}} \cdot \langle \mathbf{U}_{\mathrm{b,i}} \rangle_{\mathrm{m}} \right) \tag{2}$$

Here $\mathrm{N_p}$ is the number of particles in the cell, $\mathrm{MW_i}$ is the molecular weight of the assigned particle species, *Ru* is the universal gas constant, the operator $\langle\ \rangle$ denotes an unweighted average over all particles in the cell, and $\langle\ \rangle_{\mathrm{m}}$ denotes a mass-average. Note from the above definitions that $\mathbf{U}_{\mathrm{cell}}$ is equal to $\langle \mathbf{U}_{\mathrm{b,i}} \rangle_{\mathrm{m}}$.

B. Lagrangian face velocity calculations

Particle velocities $\mathbf{U}_{\mathrm{i}}$ are updated in such a way that, over the time step interval Δt, each particle will maintain a constant position within a Lagrangian computational cell. This Lagrangian cell is coincident with a fixed Eulerian grid cell at the beginning of the time step, but moves and deforms according to local gas properties. To track the Lagrangian cell movement, we model each face of this cell as a massless and non-porous wall from which any colliding gas molecule will be specularly reflected. Thus, no mass can pass through a Lagrangian cell face, and no net momentum or energy may be transferred from this face to the surrounding gas. It can be shown that there is a unique face velocity, in the face-normal direction, for which both momentum and energy will be exactly conserved among all colliding gas molecules over the time period Δt. Moreover, this velocity is equal to the ratio of the total incident normal momentum flux, from both sides of the face, to the total incident mass flux. An informal proof of this property is provided in the appendix.

Once the quantities ζ, $\mathbf{U}_{\mathrm{cell}}$, ρ and β have been determined for all cells in the grid, scalar values $\mathrm{u_f}$ are computed for each cell face, where the corresponding Lagrangian face velocity is $\mathrm{u_f}\mathbf{n}$ and $\mathbf{n}$ is the outward normal unit vector for this face. As the magnitude of momentum exchange between the face and a colliding gas molecule scales with the normal component of the incident relative velocity, it follows that, in a coordinate system which moves at the constant velocity $\mathrm{u_f}\mathbf{n}$, the total incident normal momentum flux to the face must be zero. We can therefore determine $\mathrm{u_f}$ by assuming equilibrium conditions on either side of the face, and setting this flux to zero.

Consider a face which separates two cells, designated as cell 1 and cell 2, where $\mathbf{n}$ is the outward normal unit vector with respect to cell 1. Based on the kinetic theory solution for the unidirectional normal momentum flux in an equilibrium flow,[2] we define a scalar function $\mathrm{G(u_f)}$, such that

$$\begin{aligned} \mathrm{G(u_f)} = {} & \frac{\rho_1}{\beta_1^2}\left[\mathrm{s_1} \exp\left(-\mathrm{s_1^2}\right) + \sqrt{\pi}\left(1+\mathrm{erf}\left(\mathrm{s_1}\right)\right)\left(\frac{1}{2}+\mathrm{s_1^2}\right)\right] \\ & - \frac{\rho_2}{\beta_2^2}\left[\mathrm{s_2} \exp\left(-\mathrm{s_2^2}\right) + \sqrt{\pi}\left(1+\mathrm{erf}\left(\mathrm{s_2}\right)\right)\left(\frac{1}{2}+\mathrm{s_2^2}\right)\right] \end{aligned} \tag{3}$$

where $\mathrm{s_1} = \beta_1\left(\mathbf{U}_{\mathrm{cell,1}} \cdot \mathbf{n} - \mathrm{u_f}\right)$ and $\mathrm{s_2} = -\beta_2\left(\mathbf{U}_{\mathrm{cell,2}} \cdot \mathbf{n} - \mathrm{u_f}\right)$. The Lagrangian face velocity which enforces momentum and energy conservation then corresponds to $\mathrm{G(u_f)} = 0$.

As Eq. (3) cannot be inverted to provide a closed form solution, an iterative procedure is required to solve for $\mathrm{u_f}$. Here we use the secant method,[1] which requires two initial guesses for $\mathrm{u_f}$. The first guess $\mathrm{u_{f,0}}$ is set to the value of $\mathrm{u_f}$ for this face during the previous time step, while the second guess $\mathrm{u_{f,1}}$ is set to a slightly different value ($\mathrm{u_f}$ + 10 m/s). The iterative procedure is then performed as

$$\mathrm{u_{f,i+1}} = \mathrm{u_{f,i}} + \left(\mathrm{u_{f,i-1}} - \mathrm{u_{f,i}}\right)\left(\frac{\mathrm{G(u_{f,i})}}{\mathrm{G(u_{f,i})} - \mathrm{G(u_{f,i-1})}}\right) \tag{4}$$

and is repeated until $\left|\mathrm{u_{f,i+1}} - \mathrm{u_{f,i}}\right|$ is within some small tolerance. No more than five iterations are typically required to reach a tolerance of 10^{-5} m/s.

The procedure to find u_f is repeated for all faces of each cell in the computational grid. As the Lagrangian face velocity will be the same for both cells which are bounded by a given face, this velocity only needs to be calculated once per time step. In the example above, a value of u_f is stored as face information for cell 1, while a corresponding value of $-u_f$ is stored for cell 2.

Note that, in a gas mixture, the characteristic thermal speed scale β used in Eq. (3) will differ for each species, as more massive atoms or molecules tend to have smaller thermal speeds at a given temperature. It follows that the momentum flux relation used as a basis for Eq. (3) is strictly valid only for a simple gas. However, by analogy with the common lack of species distinctions in inviscid CFD calculations – where the mixture molecular weight is often used as an input value in the numerical solution to a single set of mass, momentum and energy conservation equations – we can assume that Eq. (3) provides a reasonable approximation for a gas mixture. While the above procedures could be modified so that unidirectional momentum fluxes are calculated individually for every species, by employing a single β value for each cell we can reduce both the memory requirements and computational expense for simulations involving a large number of species.

Also note that the evaluation of Eq. (3) requires information for cells on both sides of a face, so that some added information exchange is needed in a parallel implementation of these procedures. When used in parallel, an extra step is required following the determination of cell-averaged gas properties, where relevant values for any cells located on a given task boundary are sent to the task which shares this boundary. This information exchange is similar to procedures already used in any DSMC algorithm with capabilities for parallel domain decomposition, and should have little influence on overall simulation time or parallel efficiency.

C. Reassignment of particle bulk velocity and temperature

While no net momentum or energy can be transferred to the gas from a Lagrangian cell face, both momentum and energy may be exchanged through this face between the two neighboring cells. As a simple example, consider the case where one such face separates two cells with zero bulk velocity and equal pressure. The root of Eq. (3) then corresponds to $u_f = 0$, so that the face will not move. The normal velocity component of a gas molecule is reversed during a specularly reflecting collision with a stationary wall, so that molecules which collide with the face will gain momentum in the inward normal direction. This momentum difference, analogous to a “pressure” force on the Lagrangian cell, must be transferred from the cell on the opposite side of the face. While in this case there is no energy exchange between the two neighboring cells, there will be some transfer of gas translational energy when u_f is not equal to zero.

To account for any momentum and energy exchange through a Lagrangian face, all particle bulk velocity and temperature values are updated during each time step through the following procedure: First, for each face bounding a given Lagrangian cell, we calculate the total momentum $\Delta\mathbf{M}_j$ transferred to the cell through this face during the time step interval Δt. For a corresponding Eulerian face area A_j and a unidirectional incident normal momentum flux Φ_j in a coordinate frame which moves at the Lagrangian face velocity $u_{f,j}\ \mathbf{n}_j$, the momentum transfer $\Delta\mathbf{M}_j$ is calculated as

$$\Delta\mathbf{M}_j = -2\,A_j\,\Delta t\ \Phi_j\mathbf{n}_j \tag{5}$$

Here $\mathbf{n}_j$ represents the outward normal unit vector at the face, and for clarity no implied summations are used.

To determine the momentum flux Φ_j we consider the cell as an unbounded equilibrium reservoir, and compute the unidirectional momentum flux through a plane which moves at velocity $u_{f,j}\ \mathbf{n}_j$ through this reservoir. Following the momentum flux expression[2] used as a basis for Eq. (3), we find

$$\Phi_j = \frac{1}{2\sqrt{\pi}}\frac{\rho}{\beta^2}\left[s_j\exp\left(-s_j^2\right) + \sqrt{\pi}\left(1+\operatorname{erf}\left(s_j\right)\right)\left(\frac{1}{2}+s_j^2\right)\right] \tag{6}$$

where $s_j = \beta\ \left(\mathbf{U}_{\text{cell}}\cdot\mathbf{n}_j - u_{f,j}\right)$. The corresponding energy transfer to the cell ΔE_j is then computed as

$$\Delta E_j = -2\,A_j\,\Delta t\ u_{f,j}\,\Phi_j \tag{7}$$

Once $\Delta\mathbf{M}_j$ and ΔE_j have been determined for all faces which bound the cell, particle bulk velocity and temperature values may be updated to account for momentum and energy exchange through Lagrangian faces between this and all neighboring cells. Each particle in the cell is assigned a new bulk velocity $\mathbf{U}_{b,i}$ such that

$$\mathbf{U}_{b,i} = \mathbf{U}_{cell} + \frac{1}{\rho V_{cell}} \sum_{j=1}^{N_f} \Delta\mathbf{M}_j \tag{8}$$

where V_{cell} is the cell volume and N_f is the total number of faces. Each particle is then given a new temperature

$$T_i = T_{cell} + \frac{2}{(3+\varsigma) n k_B} \left[\frac{1}{V_{cell}} \sum_{j=1}^{N_f} \Delta E_j - \frac{1}{2} \rho \left(\mathbf{U}_{b,i} \cdot \mathbf{U}_{b,i} - \mathbf{U}_{cell} \cdot \mathbf{U}_{cell} \right) \right] \tag{9}$$

where k_B is Boltzmann's constant, n is the cell number density, and T_{cell} is given by

$$T_{cell} = \frac{\rho}{2 n k_B \beta^2} \tag{10}$$

Note that the newly assigned bulk velocity and temperature will be equal for all particles in the cell.

For axisymmetric simulations, a modification to Eq. (8) is required to account for the fact that A_j tends to be larger for cell faces positioned further from the central axis. In this case $\mathbf{U}_{b,i}$ values are calculated as

$$\mathbf{U}_{b,i} = \mathbf{U}_{cell} + \frac{1}{\rho V_{cell}} \left[\sum_{j=1}^{N_f} \Delta\mathbf{M}_j + 2\Delta t\, \Phi_{axi}\, \hat{\mathbf{r}} \sum_{j=1}^{N_f} A_j \left(\mathbf{n}_j \cdot \hat{\mathbf{r}} \right) \right] \tag{11}$$

where $\hat{\mathbf{r}}$ is the unit normal vector in the outward radial direction, and

$$\Phi_{axi} = \frac{1}{2\sqrt{\pi}} \frac{\rho}{\beta^2} \left[s_{axi} \exp\left(-s_{axi}^2\right) + \sqrt{\pi} \left(1 + \operatorname{erf}\left(s_{axi}\right)\right) \left(\frac{1}{2} + s_{axi}^2 \right) \right] \tag{12}$$

where $s_{axi} = -\beta\, \mathbf{U}_{cell} \cdot \hat{\mathbf{r}}$. As can be shown from procedures described below, in an axisymmetric simulation all particles will move only within the plane defined by the grid. The second bracketed term in Eq. (11) therefore accounts for all out-of-plane movement of molecules in the flow being simulated. As particle position and velocity components do not need to be continually projected onto the grid plane, and as the azimuthal curvature of cell faces may be ignored, the use of Eqs. (11) and (12) potentially reduces both the expense and complexity of axisymmetric particle movement procedures relative to those used in existing DSMC-based particle methods.

D. Velocity reassignment for particle movement

Following the reassignment of $\mathbf{U}_{b,i}$ and T_i values for all particles in each cell, the velocity vectors $\mathbf{U}_i$ used for particle movement must be updated. As a first step in the determination of new particle velocities for a given cell, we calculate the locations of the corresponding Lagrangian cell vertices at the end of the current time step. Let $\mathbf{X}_{L,j}$ represent the final location of a Lagrangian cell vertex which is coincident with the Eulerian cell vertex $\mathbf{X}_{E,j}$ at the beginning of the time step. In a two-dimensional (either planar or axisymmetric) simulation, the vertex $\mathbf{X}_{E,j}$ will mark the intersection of two cell faces, given here by the outward normal unit vectors $\mathbf{n}_j$ and $\mathbf{n}_{j+1}$. It follows that the two unknown coordinates in $\mathbf{X}_{L,j}$ may be determined as a solution to the simultaneous equations

$$\begin{aligned} \left(\mathbf{X}_{L,j} - \mathbf{X}_{E,j} \right) \cdot \mathbf{n}_j &= u_{f,j} \Delta t \\ \left(\mathbf{X}_{L,j} - \mathbf{X}_{E,j} \right) \cdot \mathbf{n}_{j+1} &= u_{f,j+1} \Delta t \end{aligned} \tag{13}$$

Once $\mathbf{X}_{L,j}$ values have been computed for all $j \in [1, N_f]$, particle velocities are modified so that each particle maintains the same relative position in the Lagrangian cell defined by the vertices $\mathbf{X}_{L,j}$ as in the Eulerian cell defined by the vertices $\mathbf{X}_{E,j}$. For a triangular cell, the final particle position $\mathbf{X}_i^{'}$ at the end of the time step can be computed as a weighted sum of the three $\mathbf{X}_{L,j}$ values

$$\mathbf{X}_i^{'} = \sum_{j=1}^{N_f} \frac{\left(\mathbf{X}_i - \mathbf{X}_{E,k}\right)\cdot \mathbf{n}_k}{\left(\mathbf{X}_{E,j} - \mathbf{X}_{E,k}\right)\cdot \mathbf{n}_k} \mathbf{X}_{L,j} \tag{14}$$

where $N_f = 3$ is the total number of cell faces, $\mathbf{X}_i$ is the initial particle position, $\mathbf{n}_k$ is the outward normal unit vector for the face opposite vertex j, and $\mathbf{X}_{E,k}$ is the location of an Eulerian cell vertex along this face. (The same relation holds between $\mathbf{X}_{E,k}$ and $\mathbf{n}_k$ in Eq. (14) as between $\mathbf{X}_{E,j}$ and $\mathbf{n}_j$ in Eq. (13).) All relevant vectors in Eqs. (13) and (14) are shown for a representative triangular cell in Fig. (1).

When non-Cartesian quadrilateral cells are used in a two-dimensional simulation, each cell is temporarily divided into two triangular cells, and the summation in Eq. (14) is performed over only the three vertices bounding the triangular cell in which a particle is located. A similar procedure can be performed for three-dimensional simulations, except in this case three simultaneous equations are solved in place of Eqs. (13) to find the three unknown coordinates in $\mathbf{X}_{L,j}$, and cells are temporarily divided into tetrahedra in the evaluation of Eq. (14).

When a Cartesian grid is used, a simpler alternate procedure may be employed in place of Eq. (14) to determine the final particle position. In the two-dimensional case, $\mathbf{X}_i^{'}$ is computed as a weighted sum over all four Lagrangian cell vertex locations $\mathbf{X}_{L,j}$, where the coefficient of each $\mathbf{X}_{L,j}$ value is given by an area ratio. The numerator in this ratio is the area of a rectangle aligned with the coordinate axes and bounded on two corners by $\mathbf{X}_i$ and the Eulerian cell vertex opposite $\mathbf{X}_{E,j}$, and the denominator is the projected cell area within the grid plane. For a three-dimensional Cartesian grid, $\mathbf{X}_i^{'}$ is computed as a weighted sum of the eight bounding Lagrangian vertices $\mathbf{X}_{L,j}$, where each coefficient is proportional to the volume of the cuboid bounded by $\mathbf{X}_i$ and the Eulerian cell vertex opposite $\mathbf{X}_{E,j}$.

After the final particle location $\mathbf{X}_i^{'}$ is determined, the particle velocity is reassigned to a new value $\mathbf{U}_i = (\mathbf{X}_i^{'} - \mathbf{X}_i)/\Delta t$ which is then used in standard DSMC procedures for particle movement.

Application of the above procedures to various test cases has shown that, for certain flows and grid geometries, there is some potential for nonphysical oscillations to develop in a number of different flow properties. These oscillations have been traced to inhomogeneous particle concentrations within a cell, as may result if overlapping regions occur among two or more Lagrangian cells. It has been found that the oscillations can be completely suppressed if particles are periodically repositioned at random locations within the assigned Eulerian cell. Thus, prior to the determination of new particle velocities $\mathbf{U}_i$ as described above, a small fraction of randomly selected particles are given a new random location $\mathbf{X}_i$ in the cell. While the selection probability for each particle can generally be set to zero with no adverse effects, a probability of 10% is used here to insure sufficient damping of any oscillations.

E. Other modifications to DSMC

Aside from the steps outlined above, the only required modifications to standard DSMC procedures are in the calculation of simulation output quantities, and in the generation of new particles at inflow boundaries. In standard DSMC procedures to compute cell-based output values, translational temperature is evaluated as a function of mean and mean-squared particle velocity components in each cell, and rotational temperature is determined as a function of the average particle rotational energy. Following equilibrium assumptions on which the proposed method is based, both temperatures here are equal, and are computed as the average of assigned temperatures T_i among all particles in the cell over one or more time steps. Cell bulk velocity values are also computed differently here; this velocity is found as a mass-weighted average of particle bulk velocities $\mathbf{U}_{b,i}$, not an average among velocities $\mathbf{U}_i$ used in particle movement.

As mentioned above, modifications are also required to DSMC routines for the generation of particles at inflow boundaries. The following formula is used to determine the number of new particles N_{gen} which are generated per time step at a cell face along an inflow boundary:

$$N_{gen} = int\left(R - \frac{A_j \Delta t}{W_p} n_{in} \mathbf{U}_{in} \cdot \mathbf{n}_j \right) \tag{15}$$

Here $\mathbf{n}_j$ is the face outward unit normal vector, A_j is the face area, W_p is the particle numerical weight (the number of real atoms or molecules represented by each particle), n_{in} is the inflow number density, $\mathbf{U}_{in}$ is the inflow bulk velocity, R is a random number in [0,1], and the operator *int* rounds down to the nearest positive integer. Note that, unlike in DSMC, no particles may be generated along any inflow boundary with an outward-directed bulk velocity.

All newly generated particles are assigned a velocity $\mathbf{U}_i$, bulk velocity $\mathbf{U}_{b,i}$ and temperature T_i equal to the corresponding prescribed inflow values. Inflow properties are also used in the evaluation of Eqs. (3) and (4) to find the Lagrangian face velocity for any face located along an inflow boundary.

III. Test Cases and Simulation Results

While the method outlined above is intended to provide comparable results to a numerical solution of the compressible Euler equations, it does not involve a direct solution procedure for the governing equations, as in traditional CFD methods, or a physically motivated solution procedure for the Boltzmann equation at the equilibrium limit, as in other DSMC-based equilibrium particle methods. Instead, the proposed method is based on a macroscopic representation of molecular motion, following a set of arguments from kinetic theory: First, we note that transport coefficients tend to scale with the gas mean free path, so that an inviscid flow can be thought of as the limiting case for a dilute gas where the mean free path becomes infinitesimally small. In an inviscid gas flow simulation involving a fixed computational grid, the ratio of the local mean free path to the cell size then approaches zero.

As random (Brownian) motion of gas molecules occurs over length scales comparable to the mean free path, it follows that, on the scale of a grid cell, this random motion should be completely suppressed. On such macroscopic scales, individual molecules are therefore transported along gas streamlines at the local bulk velocity. In the proposed method, representative particles are in fact transported along streamlines, as the gas thermal energy is considered separately from particle velocities through the use of assigned particle temperatures. This contrasts with other equilibrium particle methods, where thermal and bulk translational energy are not considered independently, and where random particle motion occurs over length scales comparable to the cell size.

More generally, the method presented here is based on the well-documented Lagrangian description of fluid flows,[1,11,12] and is in some ways similar to a number of CFD schemes which involve remapping steps between Lagrangian and Eulerian coordinates.[13,14] This method can in fact be viewed as an extension to the equilibrium flux method of Pullin[6] in Lagrangian coordinates, where diffusive flux components are suppressed, and where relevant cell information is divided into discrete packets and carried by particles through the grid. The use of separate particle velocities for momentum transport and particle movement, as well as a similar particle tracking algorithm based on deformation of a Lagrangian grid, has been previously implemented in a particle-in-cell method of Brackbill and Ruppel.[15]

Still, the procedures by which flowfield information is transferred through the grid differ significantly from those used in existing numerical schemes. This leaves some doubt regarding both the effectiveness and the theoretical foundations of the new method, and compels a detailed comparison with existing methods. Such a comparison is performed using a series of one-dimensional and two-dimensional test cases, through which we demonstrate the overall accuracy of the new method and its advantages over existing equilibrium particle methods.

A. Shock tube simulation

The first test case involves simulations of a one-dimensional shock tube problem, where two equilibrium regions of zero bulk velocity, with potentially different pressures, densities and temperatures are initially separated by a non-porous membrane. After the membrane is instantaneously removed, a series of unsteady flow structures spread through the tube in both directions. The test case chosen here has been used by Laney[1] to evaluate and compare a number of different CFD schemes, and involves initial density and pressure ratios of $\rho_L/\rho_R = 8$ and $P_L/P_R = 10$. We use argon gas on both the left and right sides of the tube, and assume an initial pressure of $P_L = 1$ atm and temperature of $T_L = 273$ K on the left side.

At an elapsed time of $t = 6.5\times10^{-6}$ s, results from the new low diffusion particle method (labeled "LD" for convenience) are compared with results from simulations using the equilibrium particle simulation method (EPSM) of Pullin[6] and the DSMC collision limiter scheme of Titov, Zeifman and Levin.[4] All three simulations are carried out using a modified version of the DSMC code MONACO,[16] and all are performed on a 200×1 cell grid with specularly reflecting boundaries on all sides. Each simulation uses a time step interval of $\Delta t = 3.25\times10^{-8}$ s, with an initial population of about 4000 particles per cell on the left side and 500 on the right side. Exact Riemann solver results are also used for comparison.[1]

Gas density profiles for this test case are shown in Fig. (2), with density values normalized by the ambient density ρ_L on the left side of the grid. Here the initial discontinuity is located at x = 0.005 m, on a computational grid that extends from x = 0 m to x = 0.01 m. As shown in the exact Riemann solver results, the flow is characterized by a right-running shock wave, a contact discontinuity, and a wide left-running expansion fan. The LD results agree very well with results from the other two particle methods, and all three agree reasonably well with the expected density profile from the exact Riemann solver. The most noticeable difference between the simulation results is in the level of statistical scatter; the scatter is roughly two orders of magnitude smaller in the results from the LD simulation. The same general trends are displayed in Fig. (3) where the exact solution for the temperature profile is compared with results from all three simulation methods. All temperatures in Fig. (3) are normalized by the ambient temperature T_L to the left of the initial discontinuity.

In a second test case, we consider a shock tube problem with ρ_L/ρ_R = 8 and P_L/P_R = 1, where all other parameters and input values are the same as for the previous case. As the pressure is initially uniform across the entire grid, all transport of mass, momentum and energy will occur only through molecular diffusion. It follows that, for an inviscid flow, there should be no change in flow properties over time. Normalized density profiles at $t = 6.5\times10^{-6}$ s are shown in Fig. (4). Excellent agreement is observed here between the LD results and the exact Riemann solver profile, while results from the other two particle methods show considerable diffusion. Similar trends are observed in Fig. (5), which shows the corresponding normalized temperature profiles.

As expected, very little movement is experienced by Lagrangian cell faces in the LD simulation. The initial discontinuity at x = 0.005 m is therefore preserved nearly perfectly over time, which allows for the physically accurate simulation results shown in Figs. (4) and (5). In contrast, the free-molecular transport of particles across cell boundaries experienced in the other two simulations provides for substantial artificial diffusion and gives unphysical results. While we can find no previous application of a similar constant-pressure shock tube case to equilibrium particle methods, the poor results for both the EPSM method and the DSMC collision limiter scheme indicate the same numerical diffusion errors examined through simulations of Breuer et al.[5] and through the analysis of Macrossan.[10]

Note that a similar preservation of discontinuities in the LD simulation is not observed in the results from the first test case, shown as Figs. (2) and (3). The “smearing” effect in LD results of Figs. (2) and (3) for the right-running shock and contact discontinuity occur as a result of interpolation between Lagrangian and Eulerian cells. While any discontinuity between adjacent cells will be exactly preserved during procedures outlined above for Lagrangian cell transport, the cell properties calculated at the beginning of each time step depend on assigned quantities for all particles in the corresponding Eulerian grid cell. When some partial overlap occurs between Eulerian and Lagrangian cells, as when particles in an Eulerian cell have been located within two or more Lagrangian cells during the previous time step, some smearing of flowfield discontinuities may be expected.

It follows that this smearing effect can be avoided, or at least greatly suppressed, by either careful grid construction or time step determination. If the time step interval is roughly equal to the ratio of the cell size to the local bulk velocity, then during each time step most particles in a Lagrangian cell can be transported to a single Eulerian grid cell. This reduces interpolation effects and allows any discontinuities or strong gradients to be preserved with far greater accuracy than is likely possible through existing equilibrium particle methods. Likewise, any transverse discontinuities in a multi-dimensional flow may be accurately preserved by aligning some cell faces with gas streamlines. This cell face alignment can generally be performed by using a structured grid, as is commonly required for CFD simulations of flows involving shocks or other discontinuities.

B. Ramp-channel flow simulation

As an additional test case, we choose a steady state supersonic N_2 gas flow over a 15° inclined ramp and two-dimensional planar channel, following simulations of Kao, Liou and Chow.[17] This case has been used to evaluate a number of different CFD methods for compressible inviscid flows,[12,18-20] due in part to the complexity of the flow and the difficulty of accurately resolving certain flowfield structures. In particular, a Mach reflection occurs where an oblique shock from the ramp leading edge interacts with the upper wall of the channel. The desired sharp resolution of a slip line which originates at the Mach reflection triple point may be considered a challenging test of overall simulation accuracy. Simulation accuracy may also be assessed by gauging the width, position and orientation of other flowfield structures, including weak regular shock reflections and a centered expansion fan.

Simulations are performed on a structured grid with 300×100 quadrilateral cells and outer dimensions of 3×1 m, as shown in Fig. (6). Wall boundaries are used along the upper and lower edges of the grid, and inflow conditions along the left edge include a Mach number of Ma = 1.8, temperature T_o = 273 K and pressure P_o = 1 atm. LD and EPSM simulations are run using a constant time step of $1.5\text{x}10^{-5}$ s, with a minimum of about 20 particles per cell at steady state. Sampling is performed over 1000 time steps for both simulations, following a startup period of 4000 time steps. For comparison, an additional simulation is performed on the same grid using a standard CFD method to solve the compressible Euler equations.[21] The CFD calculations employ a second order point implicit solver with modified Steger-Warming flux vector splitting, where the maximum CFL number is set to 10.

Mach number contours for the above three simulations are shown in Figs. (7a) through (7c). In comparing results from the different methods, first note the large reduction in statistical scatter for Mach number contours from the LD simulation relative to those from the EPSM simulation. While a comparable level of scatter can be attained for both methods, this has been found to require that roughly 10 times more sampling time steps be used in the EPSM simulation.

Shock locations and shock angles are found to be nearly identical for all three simulations. While the length of the Mach stem is also nearly identical for the three simulations, significant differences can be observed in the region downstream of the Mach reflection triple point. As described above, accurate resolution of the slip line in this region is considered a challenging test of simulation accuracy. Although the desired transverse discontinuity in Mach number is not exactly preserved in any of the simulations here, this discontinuity is smeared over a relatively small number of cells in the LD simulation, and little transverse variation in Mach number is observed through most of the region downstream of the Mach stem. In contrast, the CFD results show considerably less resolution in the slip line, while the slip line seems to be entirely absent in Mach number contours from the EPSM simulation.

Another distinguishing feature of the LD results is in the width of shocks. For all shocks modeled in these simulations, the shock thickness tends to be about 50% smaller in the LD results shown in Fig. (7a) than in Figs. (7b) and (7c). As shocks should be infinitesimally thin in a truly inviscid flow, the reduction in shock thickness can be viewed as an indication of increased accuracy of the LD results relative to results from the EPSM simulation. The centered expansion fan shown in Figs. (7) is also slightly better resolved in the LD simulation, as iso-contour lines are more closely directed toward the downstream edge of the inclined ramp.

One unphysical feature is however observed in the LD results of Fig. (7a). Downstream of the shock reflection at the lower wall, we find a narrow region of large transverse Mach number gradients, where the Mach number is significantly reduced at points along the wall. While the cause of this feature is not entirely clear, one possible explanation is as follows: Wall boundaries are modeled in the LD simulation by imposing a Lagrangian face velocity of zero for all cell faces located along a wall. This in effect creates a symmetry condition at the wall, and suppresses any transverse gradients within neighboring cells. When a weak oblique shock is reflected through one of these cells, the lack of strong transverse gradients may allow the reflected shock to be treated as a normal shock at the point of intersection with the wall, resulting in a reduction in both Mach number and gas density. As effects of diffusion in the transverse direction are very small here, the large unphysical gradients at the wall tend to continue for some distance downstream.

Some reduction in Mach number along the lower wall is also observed in the CFD results of Fig. (7c), although the effect is considerably smaller. From the above discussion of other flow characteristics, it seems clear that artificial diffusion effects are generally more prominent in the CFD simulation, so it can be argued that any occurrence of this same error in the CFD simulation has been suppressed by the influence of diffusion. In fact, previous CFD simulations of this flow, which seem overall to be less diffusive, show Mach number contours along the lower wall which are more in line with those from the LD simulation.[17]

Note that the CFD algorithm employed here has not been optimized for flows involving complex shock-boundary interactions. The CFD results are therefore intended mainly for the general evaluation of flowfield structures, not for rigorous comparison with LD method results. To more accurately characterize these structures, and to provide a potentially better overall representation of the flowfield for comparison with LD results, an additional CFD simulation has been performed using a refined grid with 600×200 quadrilateral cells. Mach number contours from this simulation are shown in Fig. (7d).

In comparing the LD simulation results of Fig. (7a) with Fig. (7d), we find relatively good overall agreement, with similarly narrow shocks and a well-defined slip line. However, one important difference is observed: The Mach stem is positioned slightly further (1.3 cm) upstream and is about 15% longer in Fig. (7d) than in Fig. (7a). As a result of this difference, subsequent shock reflections also occur slightly further upstream in the LD simulation. Still, the Mach stem height and location in the LD results is within the expected range based on published CFD simulations of this flow,[12,17-20] and the agreement observed in these characteristics between LD and CFD calculations employing the same grid gives us reasonable confidence in the overall accuracy of the LD simulation.

Pressure contours for the ramp-channel flow simulations are shown in Figs. (8), where contour values are normalized by the static pressure at the inflow boundary P_o. Figure (8c) shows results from the CFD simulation using the refined 600×200 cell grid, and is included here to represent the desired flowfield characteristics for comparison with the LD results in Fig. (8a). Corresponding contour plots for the normalized gas density are shown in Figs. (9a) through (9c).

Similar trends are observed in both sets of figures as in the Mach number contour plots. As found in Figs. (7a) and (7b), effects of statistical scatter are much more noticeable in the EPSM pressure and density contours than in the those from the LD simulation, while shock thickness is significantly reduced in the LD results. In addition, the density difference across the slip line is similarly resolved in the LD simulation as in the CFD simulation employing the refined grid, while this density difference is nearly indiscernible in the EPSM results.

One very important simulation property which has not yet been discussed is overall computational expense. A general estimate of computational expense in particle methods may be taken by assessing the average calculation time per particle and per time step. By this measure, we find that the LD method is about 55% as expensive as the DSMC collision limiter scheme used in the shock tube simulations, and about 145% as expensive as the EPSM method. While these figures might seem discouraging, it should be emphasized that comparable accuracy for an EPSM or collision limiter simulation may require far more particles and a much finer grid, while comparable precision may require sampling over a much larger number of time steps. After consideration is made for differences in the magnitude of scatter and diffusion effects, the LD method should therefore permit a large reduction in overall simulation time when given levels of accuracy and precision are desired.

To quantify this potential reduction in simulation time, various results are compared between LD and EPSM simulations of the ramp-channel flow. Based on maximum Mach number gradients, we estimate that comparable resolution of the slip line requires a reduction in cell size by a factor of about 2.6 in the EPSM simulation, and a similar cell size reduction is required for comparable shock wave thickness. It follows that 2.6 times more time steps are needed in the EPSM simulation before sampling can begin, and that about 6.8 times more particles must be used to have the same number of particles per cell as in the LD simulation. As mentioned above, a similar level of scatter in output values requires roughly 10 times more sampling time steps in the EPSM simulation. Taking these factors together, and considering the approximately 45% efficiency increase in the EPSM simulation when measured per particle per time step, we find that the total EPSM simulation expense will be about 20 times greater for comparable levels of overall accuracy and precision as in the LD simulation. By the same standards, a simulation employing a DSMC collision limiter scheme may be estimated to take roughly 50 times longer than the LD simulation.

A general comparison of computational expense between LD and CFD simulations is made more difficult by inherent differences between particle and continuum flow solvers. For example, the inherent scatter in particle methods prevents determination of solution convergence by residual tracking, while optimization for efficiency and accuracy may be governed by a different set of numerical parameters (such as the number of particles per cell, or the number of sampling time steps) in particle methods. Still, we expect that a typical LD simulation will be several times slower than a fully optimized simulation of the same flow using an implicit high-order CFD scheme. It should be noted however that the CFD simulation of the ramp-channel flow on the refined grid was roughly 16 times as expensive as the corresponding LD simulation. The CFD simulation required about 24 CPU hours running in parallel on 16 processors, while the LD simulation was completed in slightly less than 1.5 CPU hours on 8 processors in the same cluster.

IV. Summary and Conclusions

A new particle method has been presented for the simulation of compressible inviscid gas flows. This method, referred to for convenience as the low diffusion (LD) particle method, is closely based on the DSMC method for rarefied gas flow simulation, and may be implemented through a series of minor modifications to an existing DSMC algorithm. Unlike other DSMC-based simulation techniques intended for the same class of flows, the proposed LD procedures allow representative particles to move in a physically realistic manner over cell-based macroscopic length scales much greater than the local mean free path. The intended reduction in numerical diffusion is evaluated through simulations of a one-dimensional unsteady shock tube flow and a more complicated two-dimensional ramp-channel flow.

The level of general agreement between results from ramp-channel flow simulation using LD and CFD methods may be seen as compelling evidence that the LD method does effectively simulate the underlying physics behind the compressible Euler equations, while a comparison between LD and EPSM results for this flow demonstrates considerable reduction in both scatter and diffusion effects when the LD method is used. This reinforces findings from the one-dimensional shock tube simulations, where LD results are found to agree reasonably well with the exact solution, and where clear advantages are observed over other equilibrium particle methods in both numerical diffusion effects and statistical scatter. Together these results can be seen to demonstrate the potential of the LD method as a practical alternative to CFD methods for the simulation of compressible inviscid flows.

Even when we assume that the LD method is considerably less efficient than modern CFD schemes in common use, the LD method still offers advantages over CFD that may make it more desirable for certain types of gas flow calculations. In particular, the LD method can be implemented in an existing DSMC code using a relatively small number of modifications, so the ultimate implementation of a hybrid code involving both LD and standard DSMC procedures may be far simpler than development of a hybrid CFD-DSMC code. As mentioned in the introduction, a hybrid LD-DSMC approach also allows for potentially much stronger coupling at the interface between continuum and nonequilibrium regions than is likely possible in a hybrid CFD-DSMC algorithm,[8,9] so such an all-particle hybrid approach may be particularly suitable for unsteady flow simulation.

Expected future work will involve the development, implementation and evaluation of procedures for hybrid LD-DSMC gas flow simulation. We also plan to modify the LD method for the simulation of viscous compressible flows, over the Knudsen number range for which the Navier-Stokes equations are valid, by introducing a random walk model into the numerical procedures.

Appendix

In the following, we show that there is a unique Lagrangian face velocity for which total momentum and energy are both exactly conserved. The face is modeled as a massless wall from which any colliding gas molecule will be specularly reflected. As only the normal components of momentum and energy for a colliding molecule are altered during the collision, we can consider this as a one-dimensional problem with unit cross-sectional area, where u_f is the face velocity, u_i is the incident molecule velocity and v_i is the post-collision molecule velocity. First, specular reflection requires that $v_i - u_f = -(u_i - u_f)$, so that

$$v_i = 2u_f - u_i \tag{A1}$$

To enforce momentum conservation, we need to satisfy

$$\sum m_i u_i = \sum m_i v_i \tag{A2}$$

where m_i is the mass per molecule, and the summations are carried out over all colliding molecules, from either side of the face, over the time step interval Δt. Substitution of Eq. (A1) into Eq. (A2) then gives

$$u_f = \frac{\sum m_i u_i}{\sum m_i} \tag{A3}$$

It follows that u_f must equal the ratio of the time-averaged net momentum flux through the face, $\sum m_i u_i / \Delta t$, to the corresponding mass flux $\sum m_i / \Delta t$. The total energy addition to the flow over the time step Δt may be given as

$$\Delta E = \frac{1}{2}\sum m_i v_i^2 - \frac{1}{2}\sum m_i u_i^2 \tag{A4}$$

By substituting Eq. (A1) into Eq. (A4), then substituting the right side of Eq. (A3) for u_f, we can show that $\Delta E = 0$, so that energy as well as momentum will be exactly conserved.

References

[1]Laney, C. B., *Computational Gasdynamics*, Cambridge University Press, 1998.

[2]Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, 1994.

[3]Bartel, T. J., Sterk, T. M., Payne, J. L., and Preppernau, B., "DSMC Simulation of Nozzle Expansion Flow Fields," AIAA Paper 94-2047, 1994.

[4]Titov, E. V., Zeifman, M. I., and Levin, D. A., "Application of the Kinetic and Continuum Techniques to the Multi-Scale Flows in MEMS Devices," AIAA Paper 2005-1399, 2005.

[5]Breuer, K. S., Piekos, E. S., and Gonzales, D. A., "DSMC Simulations of Continuum Flows," AIAA Paper 95-2088, 1995.

[6]Pullin, D. I., "Direct Simulation Methods for Compressible Inviscid Ideal-Gas Flow," *Journal of Computational Physics*, Vol. 34, 1980, pp. 231-244.

[7] Macrossan, M. N., Metchnik, M. V., and Pinto, P. A., "Hypersonic Flow over a Wedge with a Particle Flux Method," *Proceedings of the 24th International Symposium on Rarefied Gas Dynamics*, American Institute of Physics, 2005, pp. 650-655.

[8] Schwartzentruber, T. E., and Boyd, I. D., "A Hybrid Particle-Continuum Method Applied to Shock Waves," *Journal of Computational Physics*, Vol. 215, 2006, pp. 402-416.

[9] Macrossan, M. N., "A Particle-Only Hybrid Method for Near-Continuum Flows," *Proceedings of the 22nd International Symposium on Rarefied Gas Dynamics*, American Institute of Physics, 2001, pp. 388-395.

[10] Macrossan, M. N., "The Equilibrium Flux Method for the Calculation of Flows with Non-equilibrium Chemical Reactions," *Journal of Computational Physics*, Vol. 80, 1989, pp. 204-231.

[11] White, F. M., *Fluid Mechanics*, McGraw-Hill, Boston, 1999.

[12] Loh, C. Y., and Hui, W. H., "A New Lagrangian Method for Time Dependent Inviscid Flow Computation," AIAA Paper 97-1974, 1997.

[13] Rider, W. J., "A Review of Approximate Riemann Solvers with Godunov's Method in Lagrangian Coordinates," *Computers and Fluids*, Vol. 23, No. 2, 1994, pp. 397-413.

[14] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, 1979, pp. 101-136.

[15] Brackbill, J. U., and Ruppel, H. M., "FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions," *Journal of Computational Physics*, Vol. 65, 1986, pp. 314-343.

[16] Dietrich, S., and Boyd, I. D., "Scalar and Parallel Optimized Implementation of the Direct Simulation Monte Carlo Method," *Journal of Computational Physics*, Vol. 126, 1996, pp. 328-342.

[17] Kao, K., Liou, M., and Chow, C., "Grid Adaptation Using Chimera Composite Overlapping Meshes," *AIAA Journal*, Vol. 32, No. 5, 1994, pp. 942-949.

[18] Hui, W. H., Li, P. Y., and Li, Z. W., "A Unified Coordinate System for Solving the Two-Dimensional Euler Equations," *Journal of Computational Physics*, Vol. 153, 1999, pp. 596-637.

[19] Chit, O. J., Omar, A. A., Asrar, W., and Hamdan, M. M., "Development of Gas-Kinetic BGK Scheme for Two-Dimensional Compressible Inviscid Flows," AIAA Paper 2004-2708, 2004.

[20] Jin, C., and Xu, K., "A Unified Moving Grid Gas-Kinetic Method in Eulerian Space for Viscous Flow Computation," *Journal of Computational Physics*, Vol. 222, 2007, pp. 155-175.

[21] Scalabrin, L. C., and Boyd, I. D., "Development of an Unstructured Navier-Stokes Solver for Hypersonic Nonequilibrium Aerothermodynamics," AIAA Paper 2005-5203, 2005.

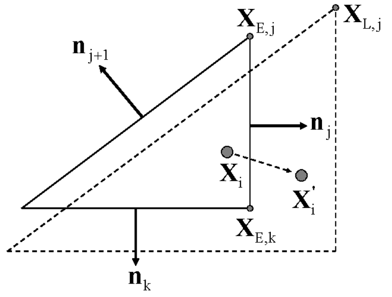

Figure 1. Diagram of vectors used in procedures for particle velocity reassignment. Dashed lines represent Lagrangian cell faces at the end of the current time step.

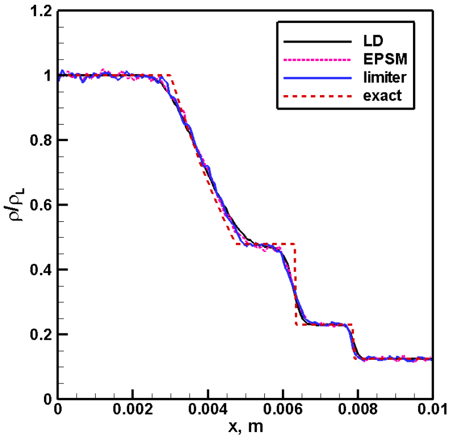

Figure 2. Normalized density profiles at t = 6.5×10^{-6} s for a shock tube problem with $\rho_L/\rho_R = 8$ and $P_L/P_R = 10$.

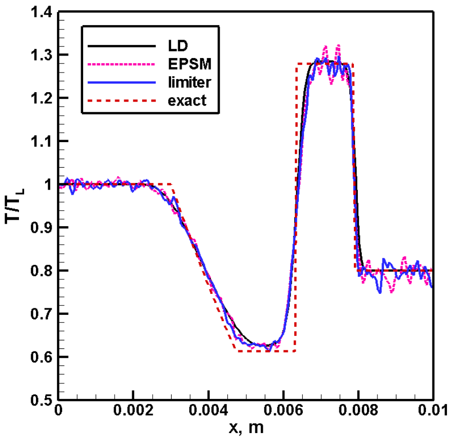

Figure 3. Normalized temperature profiles for the same case as Fig. 2.

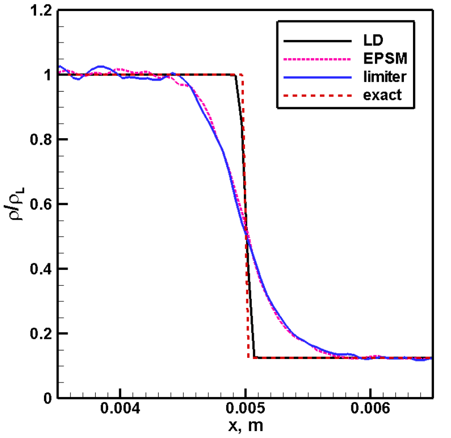

Figure 4. Normalized density profiles at t = 6.5×10^{-6} s for a shock tube problem with $\rho_L/\rho_R = 8$ and $P_L/P_R = 1$.

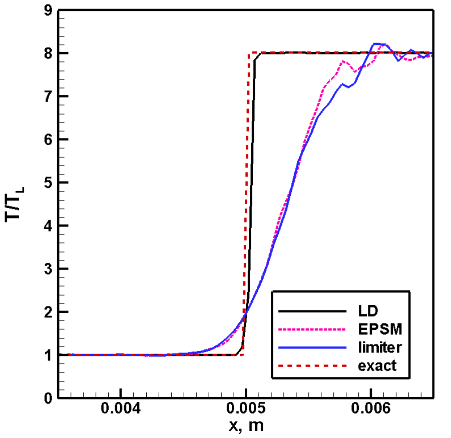

Figure 5. Normalized temperature profiles for the same case as Fig. 4.

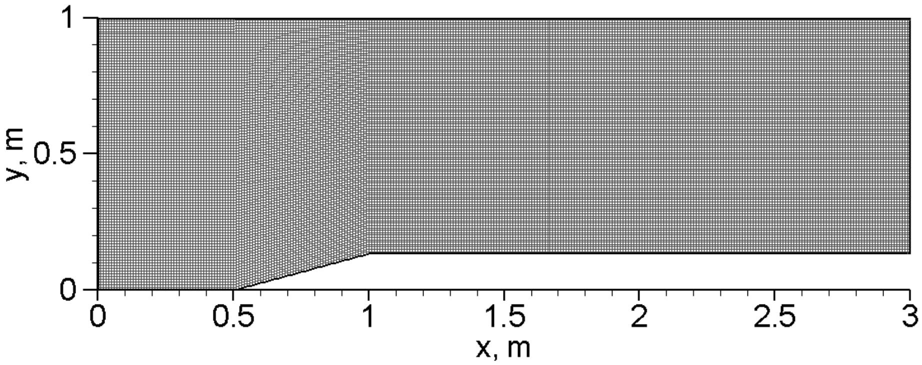

Figure 6. Grid used in ramp-channel flow simulations.

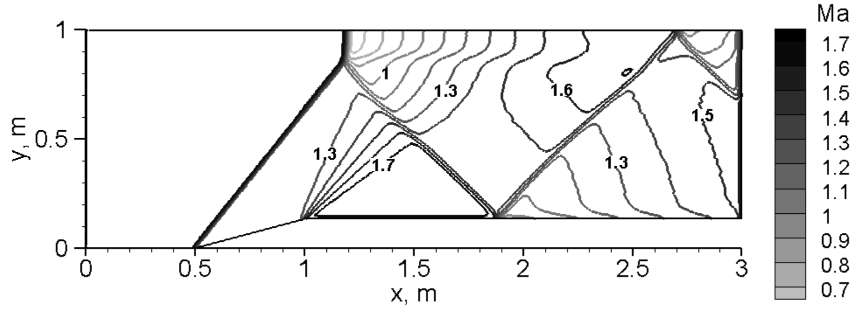

Figure 7a. Mach number contours from an LD simulation.

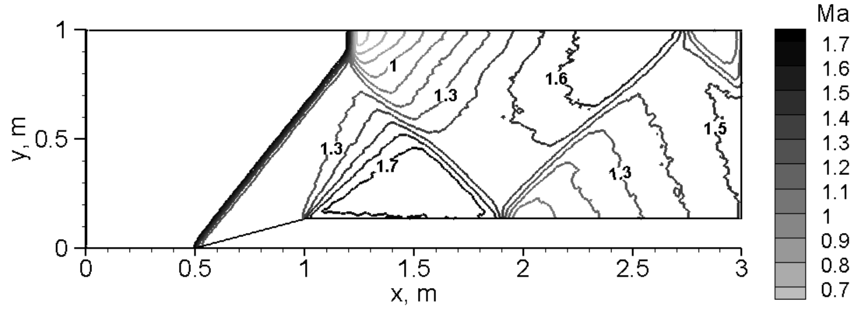

Figure 7b. Mach number contours from an EPSM simulation.

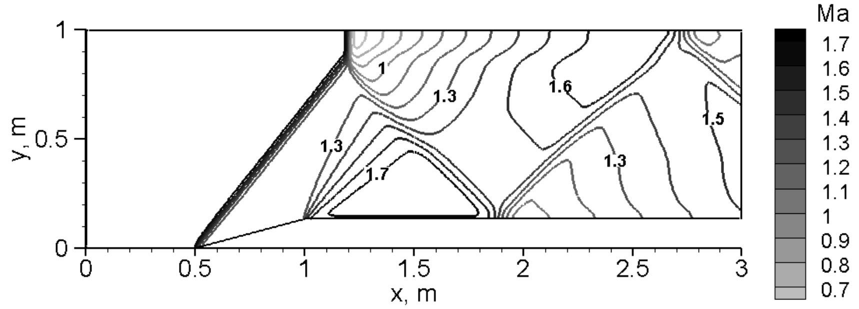

Figure 7c. Mach number contours from a CFD simulation, using the same 300×100 cell grid as LD and EPSM simulations.

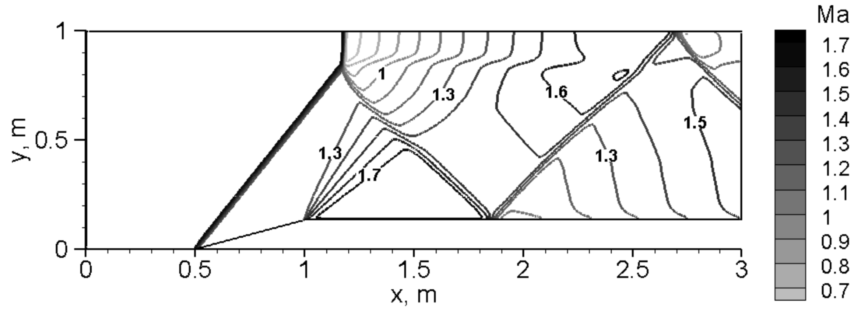

Figure 7d. Mach number contours from a CFD simulation, using a refined grid with 600×200 cells.

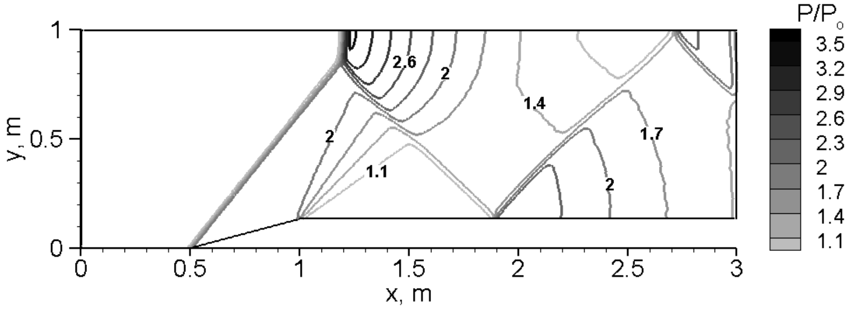

Figure 8a. Pressure contours from the LD simulation.

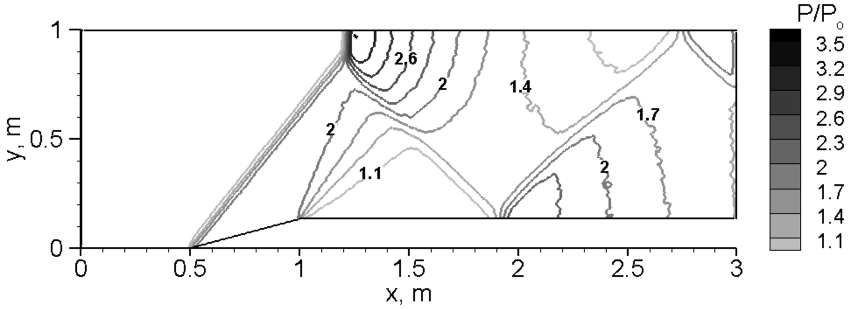

Figure 8b. Pressure contours from the EPSM simulation.

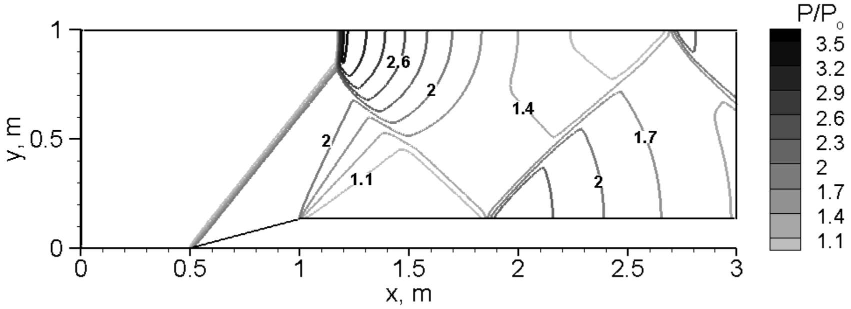

Figure 8c. Pressure contours from the CFD simulation using a refined grid.

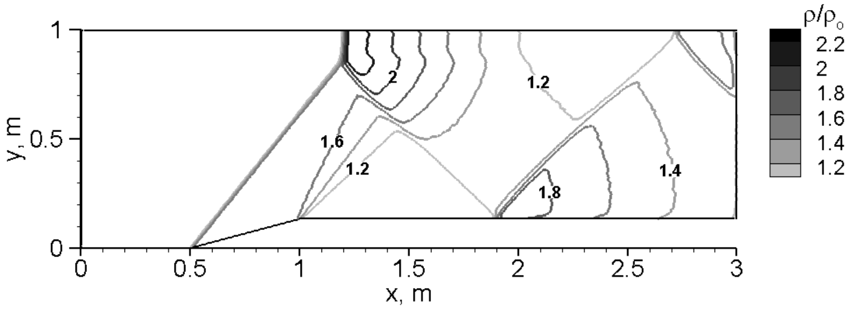

Figure 9a. Density contours from the LD simulation.

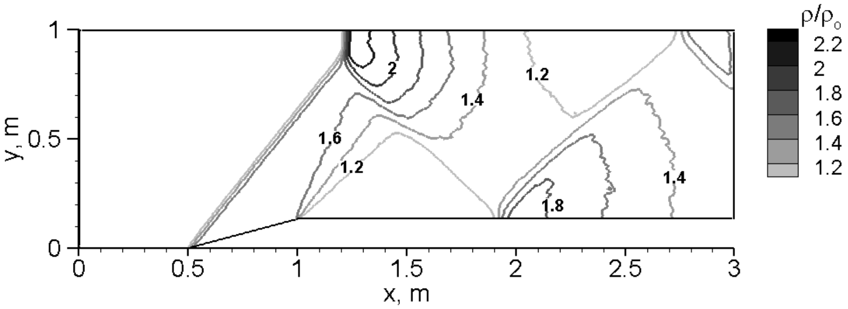

Figure 9b. Density contours from the EPSM simulation.

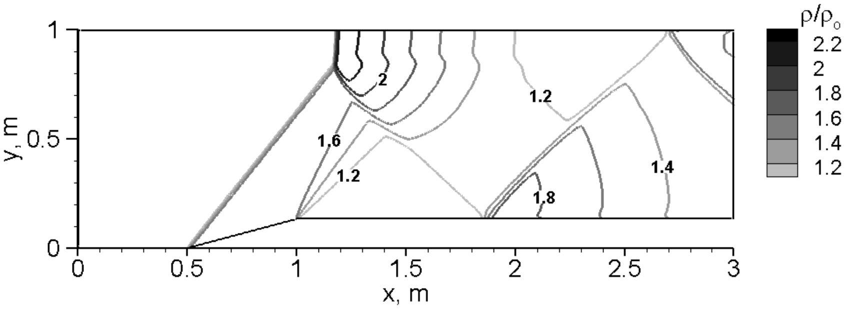

Figure 9c. Density contours from the CFD simulation using a refined grid.