Journal of Computational Physics **157,** 727–745 (2000)
doi:10.1006/jcph.1999.6397, available online at http://www.idealibrary.com on IDEAL®

Strategies for Efficient Particle Resolution in the Direct Simulation Monte Carlo Method

Keith C. Kannenberg and Iain D. Boyd

Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14853

Received August 17, 1998; revised May 10, 1999

The accuracy and computational cost of a direct simulation Monte Carlo simulation are directly related to the number of particles per cell. Optimal computational efficiency is achieved when the minimum number of particles needed for accurate resolution is used in each cell. Particle count is shown to scale proportionally with the inverse of gas density. This indicates that high density regions will tend to have few particles while low density regions are over resolved. Three methods of controlling the distribution of particles are presented—direct variation of particle weights, variation of time steps, and grid manipulation. A combination of time step variation and grid manipulation is indicated to be the most effective strategy. A sample plume expansion problem is used to demonstrate these strategies. Computational savings of up to an order of magnitude are observed.

Nomenclature

n	Number density
N_P	Number of particles per cell
W_P	Particle weight, ratio of real to simulated molecules
Δt	Time step
s	Time scale factor
Δx	Size of computational cell
λ	Mean free path
$\mathcal{V}$	Volume of computational cell
A_P	Planar area of an axisymmetric cell
R_C	Radial location of the centroid of an axisymmetric cell

INTRODUCTION

The direct simulation Monte Carlo method (DSMC) is a statistical particle method for the computation of nonequilibrium gas flows [1]. The gas is modeled at the microscopic

0021-9991/00 $35.00

level using particles which each represent a large number of physical molecules or atoms. The motion of these model particles and collisional interactions between them collectively reproduce the behavior of a macroscopic gas.

The accuracy of a DSMC simulation is directly related to the number of particles per computational cell. As the number of particles employed increases, the resolution of the computation improves both in terms of physical processes and macroscopic properties. The numerical error of macroscopic properties predicted by the DSMC method was investigated by Chen and Boyd [2]. Two components of error were discovered. One is due to the use of a finite sample size in calculating an average property. The second, called a bias, is due to the use of a finite number of particles in each cell to simulate collision processes. The limiting factor is computational cost which is, to first order, proportional to the total number of particles. When dealing with large flow problems which require many grid cells, it is generally necessary to use a small number of particles per cell in order to have a reasonable total for the simulation.

A minimum number of particles per cell is needed to resolve flow physics. Collisions are calculated between pairs of particles located within the same computational cell. Pairs of particles are selected and have a probability of collision such that bulk collision rates are obtained when averaged over many pairs. If the overall collision rate is low, due to a low gas density, for example, the probability of individual collisions will be correspondingly low. If the number of particles in a cell is small it is possible to lose low probability events. Twenty particles per cell is often considered an acceptable number to resolve fluid mechanics. More particles may be required for accurate resolution in certain circumstances, particularly when slow finite rate processes or species which have small mole fractions are considered [3].

A numerically optimal calculation is one in which the desired physics are resolved using a minimum total number of particles. This optimal situation, where each cell has the minimum number of particles for accurate resolution, is not generally achieved. Large variations in the number of particles per cell occurs due to density changes and results in over resolution of portions of the flow domain and wasted computational effort. It is therefore important to consider how the particle count in each cell is affected by various computational parameters. The distribution of particles and the computational cost of a simulation can be controlled through careful manipulation of these parameters.

The problem of controlling the distribution of particles in a particle simulation has been considered in previous studies. Lapenta and Brackbill considered duplication and consolidation of particles in particle-in-cell simulations of charged particles [4]. Various studies have considered the use of varying particle weights for DSMC simulations, particularly for axisymmetric flows [5, 1]. This work discusses several methods applicable to the DSMC technique as well as the benefits and drawbacks of each. Then a general strategy is presented which may be used to obtain a numerically efficient distribution for a variety of flows. A sample rarefied gas flow is considered which demonstrates the potential savings of computational time which may be observed using this approach.

PARTICLE COUNT SCALING

The number of particles per cell is related to the local number density of the gas by the relation

$$N_P = \frac{n\mathcal{V}}{W_P}. \tag{1}$$

W_P is the particle weight, a computational parameter which is the ratio between the number of physical molecules and computational particles in the cell. If the density is assumed to be fixed, the particle count can be increased by increasing the volume of cells or reducing the particle weight.

In order to accurately model collisions using a statistical approach, the size of a computational cell in a DSMC simulation should be on the order of the local mean free path [1, pp. 214–216]. The volume of a cell can then be related to the density by the fact that the mean free path is inversely proportional to the number density. If each dimension of a cell is Δx and this is varied proportionally with the mean free path then this cell size will be inversely proportional to the density:

$$\Delta x \propto \lambda \propto n^{-1}. \tag{2}$$

The volume of a cell is a function of the cell size. The form of this relation depends on the dimensionality of the problem being simulated.

Two-Dimensional Flows

In a two-dimensional calculation, the cells are planar polygons and the third dimension of the cell is taken to be unity. The cell volume is given by

$$\begin{aligned} \mathcal{V} &\propto \Delta x \times \Delta x \times 1 \\ &\propto \lambda^2 \\ &\propto n^{-2}. \end{aligned} \tag{3}$$

Inserting this expression for cell volume into that for number of particles, Eq. (1), gives the following relation between number of computational particles and flow density (assuming constant particle weight):

$$N_P \propto n^{-1}. \tag{4}$$

The number of particles and thus the resolution decreases linearly as the density increases.

Three-Dimensional Flows

In a three-dimensional calculation, the cells are polyhedrons. The cell volume is thus given by

$$\begin{aligned} \mathcal{V} &\propto \Delta x \times \Delta x \times \Delta x \\ &\propto \lambda^3 \\ &\propto n^{-3}. \end{aligned} \tag{5}$$

Using this expression with Eq. (1) gives the following relation between number of computational particles and flow density (assuming constant particle weight):

$$N_P \propto n^{-2}. \tag{6}$$

The resolution decreases quadratically as density increases.

Axisymmetric Flows

In axisymmetric calculations, the cells are described by polygons in the simulation plane. The cell volume is that of the annular region formed by rotating the bounding polygon through 360 degrees,

$$\mathcal{V} = 2\pi R_C A_P, \tag{7}$$

where R_C is the radial location of the centroid of the cell and A_P is the planar area.

The planar area of the cell scales as the square of the local cell size. For the majority of cells the radial location of the centroid is independent of cell size so the cell volume has the same scaling as in the two-dimensional case, Eq. (3). Consequently, the particle count scales as the inverse of the density, Eq. (4).

Cells in the vicinity of the axis of symmetry have a different scaling. The centroid of a cell adjacent to the axis is located approximately one-half of a cell dimension away from the axis. The position of the centroid is then proportional to the cell size in near axis cells. This gives the following scaling for this class of cells

$$\begin{aligned} \mathcal{V} &= 2\pi R_C A_P \\ &\propto \Delta x \times (\Delta x \times \Delta x) \\ &\propto \lambda^3 \\ &\propto n^{-3}. \end{aligned} \tag{8}$$

This leads to the number of particles scaling with the inverse of the square of the density as in three-dimensional flows, Eq. (6),

$$N_P \propto n^{-2}. \tag{9}$$

Resolution Difficulties

The results for two- and three-dimensional flows both show the interesting result that the number of computational particles in a cell varies inversely with the number density at that point. This counter-intuitive result indicates that the flow field is more accurately resolved due to a larger sample size in parts of the domain that have the lowest density. The greater resolution is both in terms of physical processes and statistical accuracy.

In a typical problem, the greatest amount of physics (collisions and chemistry) occur in high density regions. In order to correctly model the flow it is necessary to resolve flow physics occurring in these regions. The particle weight for the calculation must be set to a low number so that there will be a reasonable number of particles in these high density regions. As a result, the more rarefied portions of the flow field will have a large number of particles and will be over resolved. More computational time is spent calculating the low density regions than is needed. This results in less efficient calculations.

IMPROVING PARTICLE DISTRIBUTION

From a computational standpoint, an optimal distribution of particles among cells is one where each cell has a sufficient number to resolve flow physics and provide an acceptable

sample size and no greater. This would ensure an accurate solution without wasted computational effort due to over resolution. Due to the scaling of particle count with density, only a flow with uniform number density throughout would naturally have this type of ideal distribution.

There are various ways in which the particle count distribution can be adjusted to approach the optimal case. For example, variable time step and particle weight scaling each affect the particle count. Cell sizes and shapes can be selected so as to improve the distribution. These three methods are examined in order to develop a general strategy. In each case, an estimate of the density field is used to set the parameter of interest. This initial estimate can be based on knowledge of the flow field or a preliminary simulation performed with a reduced number of particles.

Varying Particle Weights

The most direct method of adjusting the particle count in each cell is to vary the particle weight across the computational domain. Then, in each cell, simulation particles represent different numbers of physical atoms or molecules. A particle crossing between cells with different weights creates a discontinuity in mass flow across the interface. This can be accounted for by allowing a probability that a particle is either cloned (if the particle weight decreases across the interface) or destroyed (if the weight increases) when it crosses the interface. If particle weights are assigned such that they have the correct density dependence, the variation of particle count with density can be eliminated.

Varying the particle weight is the most direct method of improving resolution. Equations (4) and (6) show that the main difficulty in terms of particle count is the decrease in number of particles with increasing density. Applying particle weights with the following density dependences would counter this effect:

$$\begin{aligned} W_P &\propto n^{-1}, \qquad \text{2D} \\ &\propto n^{-2}, \qquad \text{3D}. \end{aligned} \tag{10}$$

Applying an inverse density based particle weight scaling can have significant impact on the statistical accuracy of a simulation. The use of geometric weights for axially symmetric flows was investigated by Deng *et al*. [5]. For application to flow past a blunt-faced cylinder, the effects of weights on physical accuracy and computational performance were analyzed. These studies showed that weights lead to additional computational expense and greater statistical fluctuations. Similar studies are given in Bird [1, p. 372].

This type of scaling is a particular concern for highly compressive flows such as those generated by a hypersonic blunt body. In order to counteract the density rise caused by compression in front of the body, the particle weight must drop rapidly. This produces a significant amount of cloning in the vicinity of the body. The presence of many identical particles in a region which is likely to have an important physical effect on the flow field can seriously degrade the accuracy of the calculation. At best there will be less resolution of this region than expected. In order to avoid this problem, it is suggested that the particle weight generally not decrease in the direction of the main flow.

Under certain conditions, particle weights can be an effective means of improving particle resolution. These flows are typically relatively high density expanding flows. A high density ensures collisional variation of cloned particles while expanding flows avoid the problem

of significant cloning. This method should be used with caution and only when necessary. It can rarely be used to offset the entire effect of density variations on particle count without causing significant statistical problems.

Varying Time Steps

A finite time step is used in the DSMC method to decouple particle movement from collisions. The magnitude of the time step should be a small fraction of the mean collision time, a quantity which is inversely proportional to the density. In flows where there is a large variation in density, there is a commensurate variation in the mean collision time. If a single time step is used for the entire flow domain, it is limited to the minimum value corresponding to the highest density region. This results in a waste of computational effort. Many small time steps are calculated in low density regions where the mean collision time is high when a single larger step would be sufficient. By varying the time step throughout the flow domain this inefficiency can be eliminated. In the process, a more uniform particle count throughout the flow field can be obtained.

The primary restriction on the time step is that it be a small fraction of the mean collision time. A reasonable choice for time steps is a constant fraction K of the local collision time,

$$\Delta t = K t_{col}. \tag{11}$$

We can define a local time scale s which is the ratio between the local time step, Δt, and a reference time step for the simulation, Δt_{ref}, which is based on a reference density and mean collision time. Since collision time is inversely proportional to density, the time scale s can be related to the local density,

$$s = \frac{\Delta t}{\Delta t_{ref}} \propto n^{-1}. \tag{12}$$

When different time steps are used, a single iteration of the DSMC algorithm no longer represents the same amount of physical time in each cell. This disparity in elapsed time is accounted for by effectively weighting all particles by the time scale factor [6]. This variation in time step causes the effective particle weight to be a function of the flow density,

$$\begin{aligned} W_P &= W_{P,ref} \times s \\ &\propto n^{-1}. \end{aligned} \tag{13}$$

Inserting this expression into that for the number of particles per cell (Eq. (1)), gives the result

$$\begin{aligned} N_P &= \frac{n\mathcal{V}}{W_P} \\ &\propto n^2\mathcal{V}. \end{aligned} \tag{14}$$

The volume of cells was found to vary inversely with density. The effect of varying the simulation time step according to this rule is to reduce the dependence of particle count on density. Applying the earlier results for cell volume, Eqs. (3) and (5), gives the following

expressions for particle count:

$$W_P \propto 1, \qquad \text{2D}$$
$$\propto n^{-1}, \qquad \text{3D}. \tag{15}$$

The dependence on density is eliminated for two-dimensional flows and reduced for three-dimensional problems. The behavior of axisymmetric flows is again the same as in two dimensions away from the flow axis and three dimensions near to the axis.

The use of variable time scaling according to this scheme can thus reduce or eliminate particle resolution difficulties as well as improving the convergence and efficiency of the calculation. This variation of time steps can easily be achieved if the cells in the computational domain are sized proportionally to the mean free path. The cell size will be proportional to the inverse of the density and thus be proportional to the mean collision time. The cell size can then be used to scale the time steps directly. Cell size can be measured by the minimum altitude of the cell. If cell stretching is employed (see below) the cell dimension which is directly scaled to the mean free path should be used to determine the time step.

Grid Manipulation

From the point of view of the physical correctness of the DSMC method, the ideal grid consists of a set of cells whose dimensions are some fraction of the local mean free path. The scale length of flow gradients will be on the order of the mean free path and therefore this size requirement ensures that gradients are properly resolved [1]. Smaller grid spacing may be necessary to provide sufficient spatial resolution in some cases.

In many flows it is clear from the geometry that the primary gradients will be in a specific direction. In these circumstances it is reasonable to relax restrictions on cell dimensions in the directions other than that of the primary gradients. Several mean free paths can be used rather than one or smaller. Significant improvements in particle distribution can be obtained at the cost of some spatial resolution.

Consider a two-dimensional flow with no weight or time scaling which is compressed at a wall. Since the particle count varies as the inverse of the density (Eq. (4)), the compression will cause a decrease in particle count at the wall. If cells in the vicinity of the wall are stretched in the direction parallel to the wall by a factor of 10, this increase in cell volume will counteract the effect of the density rising by the same factor. If the flow is such that the properties vary slowly along the wall (e.g., bulk flow normal to the surface) the reduction in spatial resolution will be inconsequential.

In three dimensions, the cells can often be stretched in two directions. Increasing each dimension by an order of magnitude leads to a hundredfold increase in volume, compensating for a two order of magnitude increase in density.

This type of grid manipulation can be particularly important in axisymmetric problems. Cells near the axis tend to suffer from low particle counts due to three-dimensional type scaling and are thus difficult to resolve accurately. Stretching cells in the radial direction has a squared effect on particle count due to the resultant movement of the cell centroid away from the axis.

Cells can be of any geometric shape as long as they satisfy the size requirement. From a practical point of view, cells are typically quadrilateral or triangular in two-dimensional and axisymmetric flows and hexahedral or tetrahedral in three dimensions (structured and

unstructured grids, respectively). While unstructured grids provide the most flexibility for meshing complex domains, structured cells are more easily stretched to increase the particle count and improve the distribution. In general, a hybrid grid which combines the advantages of both makes the most effective grid.

General Strategies

A combination of grid manipulation and variable time scales will often be the best means of controlling the distribution of particles. Time scaling can eliminate one level of density dependence while not impacting the accuracy of the simulation. Creative use of grid stretching can compensate for a large portion of the remaining dependence. Some spatial resolution is sacrificed but this can be managed so as to not adversely affect the overall macroscopic results.

In certain cases it is not desirable to strictly scale the cells by the mean free path. For example, flows which contain very low density regions as well as regions of moderate density are not well suited to strict mean free path scaling [7]. The extremely low density regions have a very large mean free path which would suggest very large cells. Reasonable spatial resolution would suggest using cells many times smaller. At the other extreme, it is not necessary to resolve at the level of the mean free path in near-continuum, small gradient flows. An example of this is the throat of a low density nozzle expansion [8]. Resolving this type of flow to the level of one mean free path may be unnecessary and certainly very expensive.

It is not necessary for an efficient calculation to have a completely uniform distribution. In some cases excessive efforts to obtain this may affect the macroscopic results of the calculation either by reduced spatial or statistical resolution. However, attention to this issue is extremely important when large engineering simulations are performed in order to maintain a reasonable cost.

DEMONSTRATION RESULTS

The scaling of particle count and the strategies for improving the distribution are demonstrated by considering a model problem. Figure 1 shows a schematic of the configuration. An expanding gas plume from a nozzle impacts on a surface placed downstream. The initial expansion of the plume followed by compression at the surface results in a range of densities which is suitable for examining resolution and scaling. Although this is a fairly simple flow, the results should be extensible to more complex problems.

The methods outlined above will be applied to this plume problem with the goal of approaching a uniform distribution of particles in the domain. Both two-dimensional and axisymmetric cases will be considered. Twenty particles per cell is chosen to be the desired level of resolution. Fallavollita *et al*. determined that increasing the number of particles and decreasing the number of sampling steps yields more accurate results [3]. The results presented here can be scaled to any desired particle threshold by decreasing the overall particle weight employed in the simulation.

Simulation of the plume begins at the exit of the nozzle and extends above and beyond the surface. The plume is generated by a heated nozzle flow which was investigated in a previous study [9]. Inflow profiles at the nozzle exit plane are taken from a DSMC simulation of this nozzle. At the centerline of the nozzle exit plane the flow conditions are a velocity of 1100 m/s, number density of 4.8×10^{21} m^{-3}, and temperature of 98 K. The same inflow

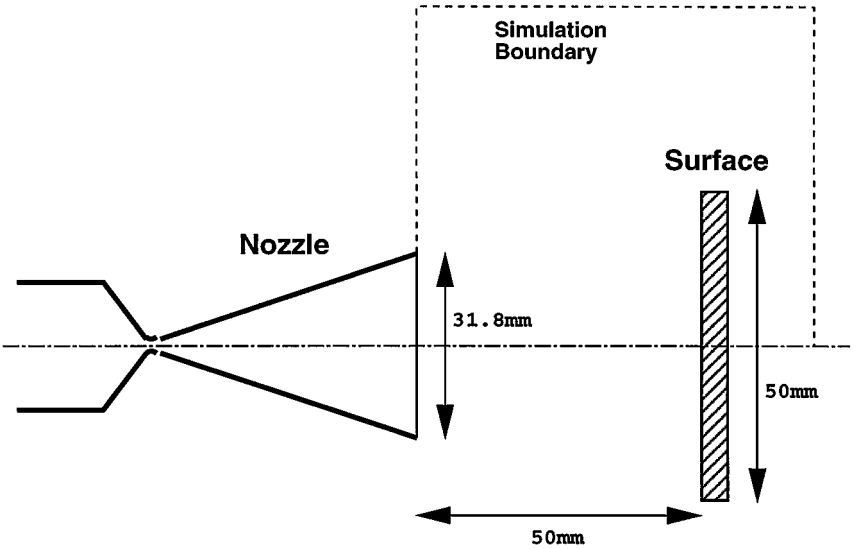

FIG. 1. Schematic of demonstration impingement case.

conditions will be used for both axisymmetric and two-dimensional simulations. The test gas is molecular nitrogen and is assumed to be vibrationally and chemically frozen.

The DSMC computations are performed using a scalar optimized, parallel algorithm called MONACO [10]. This code handles hybrid computational grids and provides the ability to use a unique particle weight and time step in each cell.

Two-Dimensional Case

The simplest, interesting case to consider is two-dimensional flow. The problem then involves the plume from a plane nozzle impinging on an infinitely wide plate. Although this configuration is not representative of a real problem, a two-dimensional simulation most clearly shows the effects of density scaling and demonstrates the appropriate strategies.

As a base case, the flow is calculated using a fully unstructured (triangulated) grid. Cell sizes are twice the local mean free path as determined by a coarse, preliminary calculation. Two mean free paths are used in order to reduce the computational cost. For purposes of this study this level of spatial resolution is sufficient. Figure 2 shows the grid for this calculation.

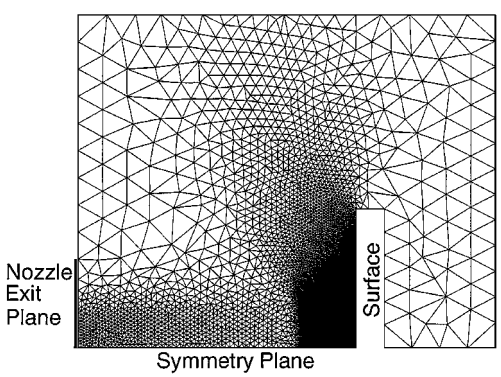

FIG. 2. Two-dimensional unstructured grid for demonstration impingement case.

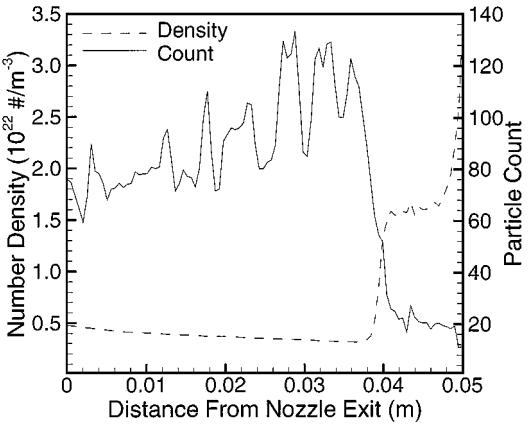

FIG. 3. Density and particle count along stagnation line for base two-dimensional case.

Compression of the gas at the surface results in a high density of grid cells while expansion away from the body causes the cells to rapidly grow in size.

Data at the symmetry plane will be examined to consider the effects of particle count scaling. The symmetry plane is also a stagnation plane for the flow so that the maximum compression at the surface occurs here. The wide range of densities occurring on this plane (or line in the simulation space) makes this the most interesting portion of the domain to consider for this study.

Figure 3 shows results taken along the stagnation line. Number density and particle count are plotted against distance from the nozzle exit. As expected, the number of particles per cell increases through the expansion region and decreases as the flow compresses in front of the impingement surface. Equation (4) indicates that the particle count will be inversely proportional to the density. This relationship is demonstrated in Fig. 4 which plots the product of density and particle count along the stagnation line, normalized by the values at the exit plane. The product is constant on average with fluctuations of 20%. Thus the particle count is inversely proportional to the density to within this variation.

The deviation from inverse proportionality is a result of the variation of cell sizes occurring in the grid. Figure 5 plots cell volumes and the square of the local mean free path along

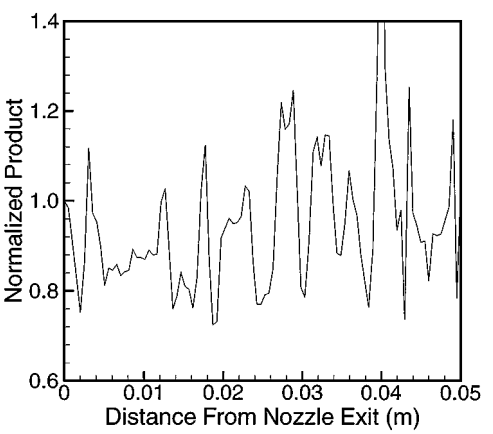

FIG. 4. Normalized product of density and particle count along stagnation line for base two-dimensional case.

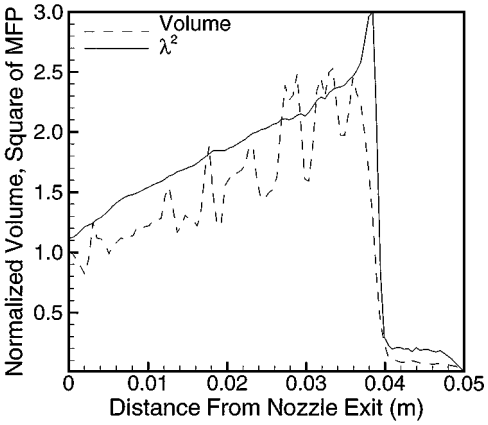

FIG. 5. Normalized cell volume and square of mean free path along stagnation line for base two-dimensional case.

the stagnation line. Both profiles are normalized using values at the inflow plane. In an ideal grid, cell dimensions would be exactly proportional to the mean free path leading to volumes which are proportional to the square of the mean free path in a two-dimensional flow. In a real grid, some variation in cell dimensions is necessary to completely fill the domain. This variation in cell size and volume causes the particle count to deviate from the inverse linear relationship specified by Eq. (4).

The calculation uses a total of 750,000 particles. If 20 particles per cell is assumed to be necessary to accurately resolve the flow then the simulation size cannot be made smaller for this problem without improving the particle distribution. Cells immediately at the surface have somewhat less than 20 particles so that a smaller particle weight and larger total number of particles would be needed to satisfy the 20 particle limit throughout.

Embedded Surface Grid

Compression of the gas at the surface causes the particle count to drop rapidly in the vicinity of the surface. The use of an embedded quadrilateral grid with cell stretching parallel to the surface can improve resolution. Figure 6 shows a portion of the resulting hybrid grid when such an embedded grid is included. At the surface, cells are ten mean free paths in height, perpendicular to the surface gradients, and two in width. At the outer extent of the embedded grid the aspect ratio of the cells is much closer to unity; this is necessary

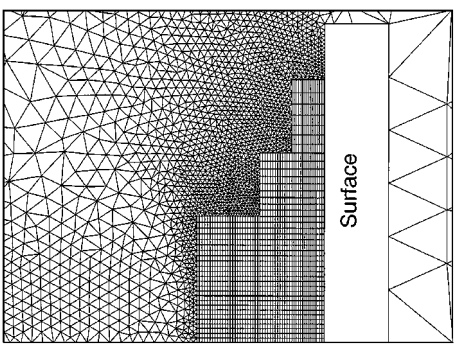

FIG. 6. Hybrid grid with embedded quadrilateral grid at surface.

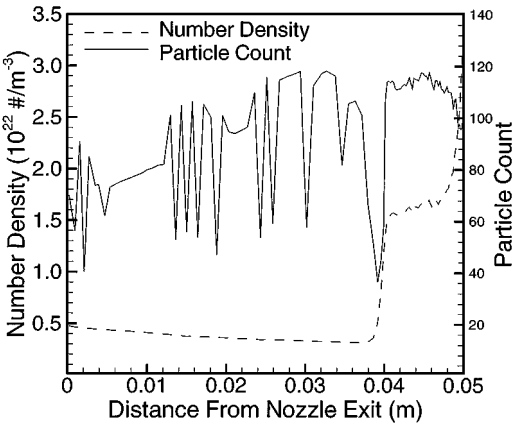

FIG. 7. Density and particle count along stagnation line for hybrid grid, two-dimensional case.

to have a smooth transition to the unstructured portion of the grid. The embedded grid extends to the edge of the compression region in front of the body (approximately 10 mm). It extends over only 80% of the height of the body. Above this point gradients in the vertical direction start to become important making grid stretching less acceptable. Additionally, the density has fallen off sufficiently at this point so that the unstructured grid can provide adequate resolution.

The effect of the embedded grid is to improve particle resolution at the surface without sacrificing important spatial resolution. Figure 7 shows the variation of density and particle count along the stagnation line. The number of particles per cell in the compression region is increased by a factor of five, the same factor used to stretch the surface quadrilateral cells. Density and temperature profiles from the base and embedded grid cases are compared in Fig. 8. Both properties are unaffected by the change in grid structure. Stretching of the surface grid makes it possible to fully resolve surface conditions using fewer overall particles.

The use of an embedded grid has no effect on the total number of particles in the simulation. Since the same particle weight is employed as in the base two-dimensional case,

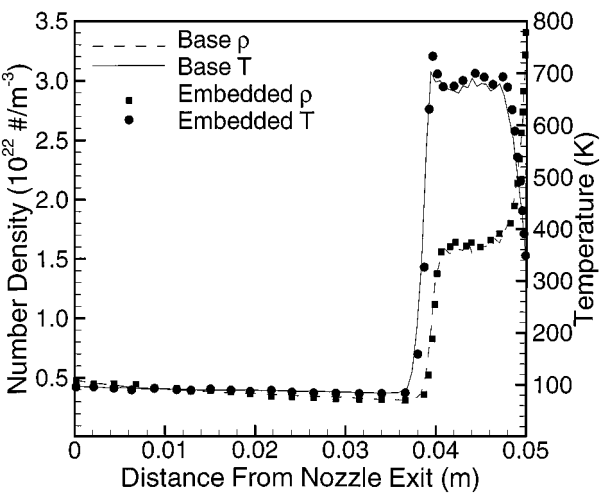

FIG. 8. Comparison of density and temperature profiles for base two-dimensional and hybrid grid case.

the size of the simulation remains the same (750,000 particles). However, Fig. 7 indicates that the total number of particles could be reduced by a factor of two (by increasing the particle weight) and the nominal 20 particle limit would be satisfied everywhere on the stagnation plane. Since the stagnation plane has the highest densities and is thus the most difficult portion of the flow to resolve, this conclusion can be considered to hold throughout the flow field. Thus, using stretched quadrilateral cells at the surface would allow the problem to be accurately calculated using only 375,000 particles and a 50% savings in CPU time.

Variable Time Steps

The scaling of particle count with density can be effectively overcome in two-dimensional flows by utilizing a time step in each cell which is inversely proportional to the density. The fully unstructured grid is employed along with time step variation according to this rule. The local density is estimated using the size of cells, measured by the altitude of the triangle, since cells are sized according to the mean free path.

The use of time scaling according to an inverse density rule should eliminate the variation of particle count with density. Figure 9 shows the resulting densities and particle counts along the stagnation line. The particle count shows no correlation with density and is constant on average. Fluctuations can again be attributed to deviation in cell size from the local mean free path. The effect is compounded by the use of cell size in the determination of time scale. A more accurate scheme would directly utilize the density field which was used for grid generation.

Density and temperature profiles on the stagnation line for the base and variable time step cases are compared in Fig. 10. The scaling of time steps has no significant effect on the resulting flow field properties.

The results of this calculation indicate that for this problem variable time steps are the most effective way of obtaining a uniform particle count distribution. The distribution is approximately independent of density throughout the flow field. This provides the most efficient use of computational power.

The use of variable time steps directly affects the number of particles in the simulation through the effective particle weight. As a result the total number of particles used is

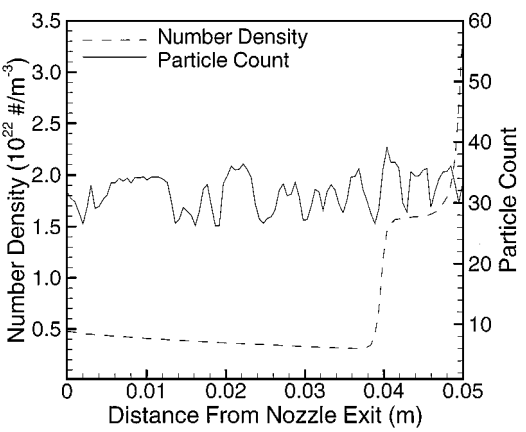

FIG. 9. Density and particle count along stagnation line for two-dimensional case with variable time steps.

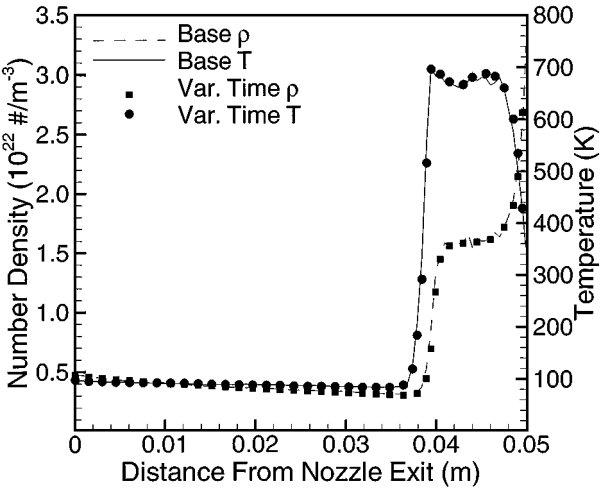

FIG. 10. Comparison of density and temperature profiles for base two-dimensional and variable time step case.

only 540,000 with the same overall particle weight. Further, Fig. 9 indicates that the total number could be reduced by 40% and still satisfy the requirement of 20 particles per cell. Additionally, the flow converges to a steady state in many fewer iteration using variable time steps. The length of the transient is reduced by a factor of four in this case. The total CPU time savings which can be realized for this problem using variable time steps is on the order of 80%.

Axisymmetric Case

The same impingement flow is used to demonstrate the effects of density scaling in axisymmetric flows. The configuration is now a conical nozzle generating a plume which impacts on a disk. Qualitatively the flow field is the same as in two dimensions. However, the shock is closer to the body and the resulting compression region is smaller in the axisymmetric case.

As a base case the flow is calculated using an unstructured grid using variable time steps. The grid used differs from that used in two dimensions because the density field is different. Time steps are again scaled using cell size. In an axisymmetric flow, the particle count should scale like a two-dimensional flow away from the axis and like a three-dimensional flow close to the axis. Using variable time steps the particle count should then be approximately constant off axis and inversely proportional to density at the axis.

Figure 11 shows density and particle count taken at a radius equal to half the nozzle radius. This particle count is approximately constant from nozzle to surface which is expected at this radius. Fluctuations are again a result of variations in cell size. Figure 12 shows a similar profile taken at the axis. As expected the particle count varies inversely with the density. The inverse linear relationship is demonstrated in Fig. 13 which plots the normalized product. The magnitude of fluctuations is somewhat larger than seen previously due to the extremely low particle count.

This calculation uses a total of 430,000 particles. The extremely low number of particles at the axis, around 0.25 per cell at the wall, indicates that approximately 80 times more

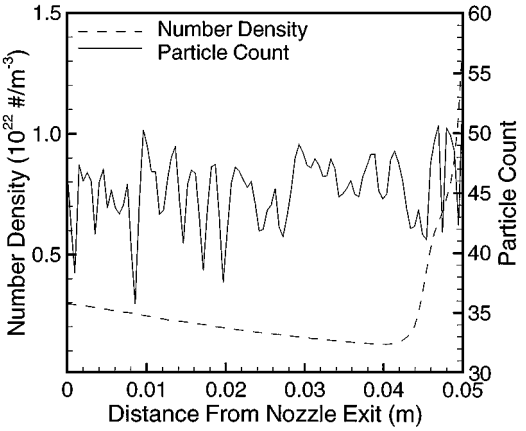

FIG. 11. Density and particle count at radius equal to half the nozzle radius for axisymmetric case.

particles would be needed to meet the nominal limit of 20 particles per cell. Although this limit can be relaxed somewhat for cells in the vicinity of the axis, the size of the simulation would need to be increased by at least an order of magnitude to achieve reasonable accuracy.

Embedded Axial Grid

The major resolution difficulty with axisymmetric flows is in the vicinity of the axis. The scaling of particle count with radius means that there will be a small number of particles near the axis. This is especially evident in high density regions at the axis due to the additional scaling with density. In order to properly compute any axisymmetric flow this problem must be overcome.

There cannot be gradients in the radial direction at the axis of symmetry. It is thus appropriate to stretch cells in the radial direction in the vicinity of the axis. The use of stretched quadrilateral cells at the axis can have a dramatic effect on cell volumes and the

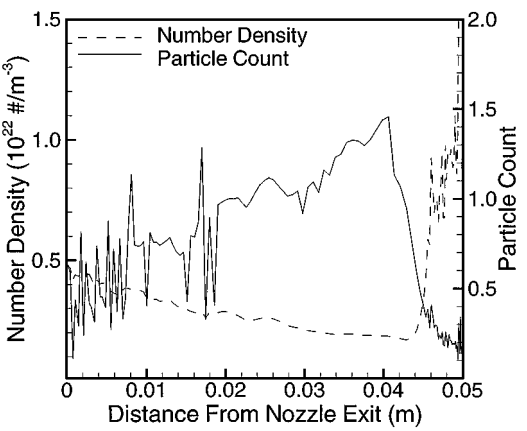

FIG. 12. Density and particle count along axis for axisymmetric case.

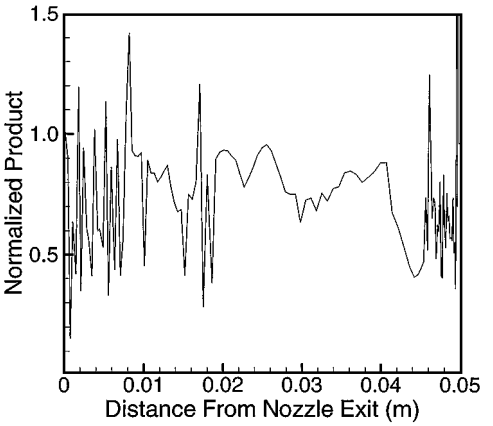

FIG. 13. Product of density and particle count along axis for axisymmetric case.

resulting particle counts. Consider a cell which is an equilateral triangle of side length l located at the axis. The volume of this cell is given by

$$\begin{aligned} \mathcal{V} &= 2\pi R_C A_P \\ &= \frac{1}{2}\pi l^3. \end{aligned} \tag{16}$$

A quadrilateral cell whose radial extent is stretched by a factor of 3 to $3l$ has the volume

$$\begin{aligned} \mathcal{V} &= 2\pi R_C A_P \\ &= 9\pi l^3. \end{aligned} \tag{17}$$

A stretched quadrilateral with a three to one aspect ratio therefore results in an 18-fold increase in cell volume.

An embedded quadrilateral grid is used to resolve the axis of the impingement flow. Four rows of cells are used. Radial stretching by a factor of three is used for the first row, closest to the axis. A factor of two is used in the second row and the third and fourth rows are

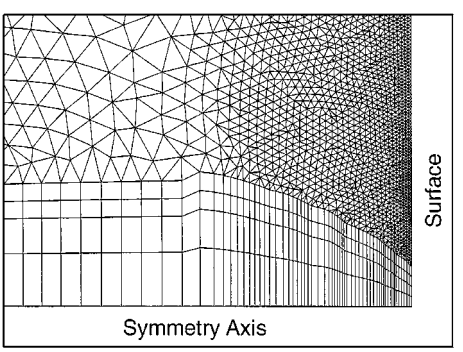

FIG. 14. Hybrid grid at axis for axisymmetric case.

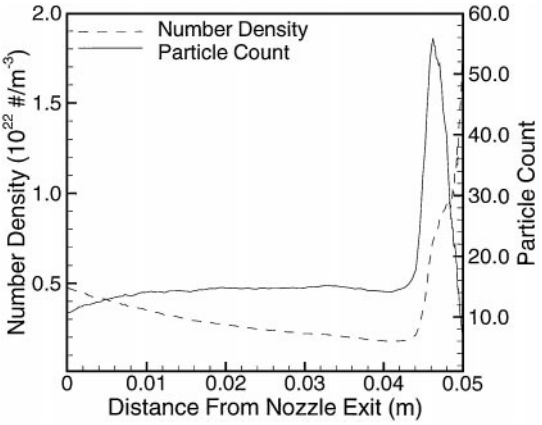

FIG. 15. Density and particle count along axis for axisymmetric case using hybrid grid.

unstretched. Within the compression region at the surface, stretching by a factor of five is used for all rows to compensate for the rise in density. Figure 14 shows a portion of the embedded grid near the front of the impingement surface.

Figure 15 shows density and particle count along the axis using the hybrid grid. The number of particles per cell is at an acceptable level across the range and is fairly uniform. The peak occurring at the beginning of the compression region indicates that the stretching is somewhat excessive at this point. This results from the necessity to have smoothly varying cell sizes in the structured grid at the axis.

Density and temperature profiles along the axis for the unstructured and embedded grids are compared in Fig. 16. Both grids show essentially the same profiles.

The total size of the simulation is unaffected by the change in grid structure and remains approximately 430,000 particles. Although the nominal minimum of 20 particles per cell is not reached in most cells, grid stretching results in an order of magnitude increase in

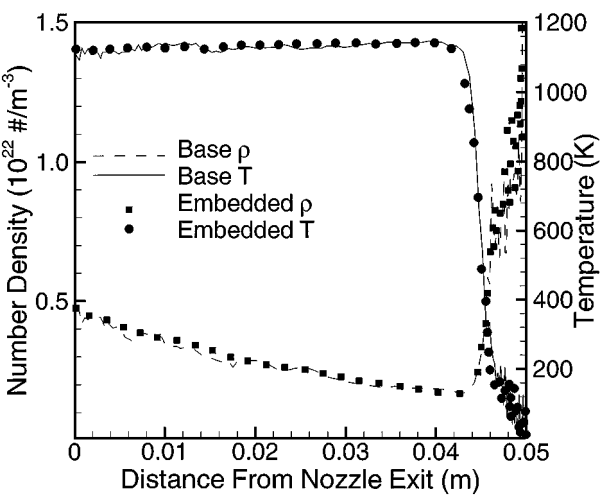

FIG. 16. Comparison of density and temperature profiles for base axisymmetric and hybrid grid case.

particle count and resolution. The cost of accurately computing this flow would then be reduced by an order of magnitude.

CONCLUSIONS

The number of particles employed in a DSMC simulation has a first order effect on the overall cost of the calculation. A uniform number of particles per computational cell throughout the flow domain ensures that computational effort is used efficiently. This optimal situation is not obtained under most circumstances because particle count scales with flow density. The scaling is such that the number of particles decreases with increasing density, resulting in over resolution of low density regions and wasted computational effort. This effect is particularly important in three-dimensional flows where the scaling with density is quadratic.

Several strategies have been presented to improve the distribution of particles. Direct manipulation of local particle weights is the least desirable method due to the detrimental effects of cloning of particles on statistical accuracy. Variation of time steps can reduce the scaling of particle count with density by one order of magnitude without side effects while also improving convergence. Manipulation of grid structure and shape can result in significant increases in resolution while sacrificing some spatial resolution.

A model flow problem was considered which included both expansion and compression. For a two-dimensional simulation, the use of grid cell stretching was found to reduce the overall cost of the simulation by 50%. The use of variable time steps reduced the cost by 80%. Under axisymmetric conditions, variable time steps are required to perform the simulation in reasonable time. The use of grid cell stretching along with variable time steps reduces the computational cost by an order of magnitude when compared with a simulation using only variable time steps.

The results from this model problem should be directly applicable to a wide range of problems when simulated using DSMC. The strategies which lead to a significant improvement in particle distribution and computational efficiency are not limited to this model problem. Complexities in the physical model such as vibrational relaxation or chemical reactions should not affect the applicability since the strategies are related to the kinematics of the gas. The methods extend directly to three-dimensional flows as well where efforts to improve performance are extremely important.

REFERENCES

1. G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows* (Clarendon, Oxford, 1994).
2. G. Chen and I. D. Boyd, Statistical error analysis for the direct simulation monte carlo method, *J. Comput. Phys*. **126**, 434 (1996).
3. M. A. Fallavollita, D. Baganoff, and J. D. McDonald, Reduction of simulation cost and error for particle simulations of rarefied flows, *J. Comput. Phys*. **109**, 30 (1993).
4. G. Lapenta and J. Brackbill, Dynamic and selective control of the number of particles in kinetic plasma simulations, *J. Comput. Phys*. **115**, 213 (1994).
5. H. F. Deng, J. P. Stark, and H. Lidell, An investigation into the effect of weighting factor on computational performance in 2d axi-symmetric flow past a flat-nosed cylinder, in *Rarefied Gas Dynamics 20*, edited by C. Shen (Peking Univ. Press, Beijing, 1997), p. 155.

6. K. C. Kannenberg, *Computational Methods for the Direct Simulation Monte Carlo Technique with Application to Plume Impingement*, Ph.D. thesis, Cornell University, Ithaca, New York, 1998.
7. D. P. Karipies, I. D. Boyd, and G. E. Caledonia, Development of a Monte Carlo overlay method with application to spacecraft glow, *J. Thermophys. Heat Transfer* **12**, 30 (1998).
8. I. D. Boyd, P. F. Penko, D. L. Meissner, and K. J. DeWitt, Experimental and numerical investigations of low-density nozzle and plume flows of nitrogen, *AIAA J*. **30**, 2453 (1992).
9. K. C. Kannenberg, I. D. Boyd, and S. Dietrich, *Development of an Object-Oriented Parallel DSMC Code for Plume Impingement Studies*, AIAA Paper 95-2052, June 1995.
10. S. Dietrich and I. D. Boyd, Scalar and parallel optimized implementation of the direct simulation Monte Carlo method, *J. Comput. Phys*. **126**, 328 (1996).