

# A Novel Cartesian Implementation of the Direct Simulation Monte Carlo Method

Jonathan M. Burt<sup>1</sup> and Eswar Josyula<sup>2</sup>  
*U.S. Air Force Research Laboratory  
Wright-Patterson Air Force Base, OH 45433*

Iain D. Boyd<sup>3</sup>  
*Department of Aerospace Engineering  
University of Michigan, Ann Arbor, MI 48109*

**A new Cartesian implementation of the direct simulation Monte Carlo (DSMC) method, named the hypersonic aerothermodynamics particle (HAP) code, is presented. This code is intended for rapid setup and simulation of rarefied flow problems, and as a framework for evaluating new physical models and numerical techniques. Unique features include the use of nonuniform Cartesian adaptive subcells, a collision probability modification to reduce errors associated with spatial averaging in collision probabilities, and automatic planar element approximation of analytically defined two or three-dimensional surface geometries. In this work, simulations are performed using both HAP and an established DSMC code for a rarefied hypersonic flow over a flat plate, and excellent overall agreement is found. Additional simulations are employed to demonstrate reduced dependence on cell size through a proposed collision probability modification. Results are also presented for a three-dimensional HAP simulation of hypersonic flow over a blunted cone, and reasonably good agreement with experimental data is observed.**

## I. Introduction

The direct simulation Monte Carlo (DSMC) method, introduced by Bird nearly 50 years ago,<sup>1,2</sup> has over the past several decades become the dominant technique for numerical simulation of rarefied gas flows. This method involves Lagrangian tracking and binary collision operations among a collection of simulated particles, each representing a large number of atoms or molecules, in a manner which reproduces the underlying physics of the governing Boltzmann equation.

DSMC is often used in a wide variety of applications related to rarefied gas flow simulation, including aerothermodynamic design and analysis of atmospheric entry vehicles and hypersonic cruise vehicles, simulation of nanoscale and microscale subsonic flows, simulation of exhaust plumes from spacecraft or high altitude rockets, chemical vapor deposition, spacecraft drag calculations, and analysis of basic fluid flow phenomena and instability problems. The popularity of DSMC in this broad range of applications can be attributed to a combination of accuracy, simplicity, computational efficiency, and relatively easy application to complex geometries. In rarefied gas dynamics problems such as those mentioned above, the equilibrating effects of intermolecular collisions may be insufficient to retain a near-equilibrium gas velocity distribution, and the resulting nonequilibrium velocity distributions tend to invalidate underlying assumptions in the Navier-Stokes equations.<sup>2</sup> As a result, standard computational fluid dynamics (CFD) techniques generally provide results of poor or questionable accuracy, and alternate methods<sup>3-5</sup> which reproduce the nonequilibrium characteristics of the Boltzmann equation are required. Among such alternate methods, DSMC is often the most efficient technique which maintains the accuracy of the full Boltzmann equation.

In this paper, we present a new simplified implementation of DSMC named the hypersonic aerothermodynamics particle (HAP) code. This code is intended to combine the simplicity of problem-specific DSMC implementations with many capabilities of larger generalized DSMC codes,<sup>6-11</sup> and development of the HAP code is motivated in part

---

<sup>1</sup> Research scientist, Universal Technology Corporation, AIAA member.

<sup>2</sup> Senior research aerospace engineer, AIAA associate fellow.

<sup>3</sup> Professor, AIAA associate fellow.

by a desire to ease implementation and testing of new DSMC models and algorithms. The code is made considerably simpler through the use of a uniform Cartesian grid, and newly developed collision procedure modifications are utilized to reduce spatial discretization errors. As an additional motivation for development of the HAP code, a very simple, intuitive and configurable user interface is desired for rapid simulation setup by inexperienced DSMC users.

The HAP code is described in the following sections, and various features of the code are outlined. Next, errors associated with a lack of grid adaptation to the local mean free path are discussed, and set of new techniques are proposed to reduce these errors. A rarefied hypersonic flow over a flat plate with a sharp leading edge is used as a test case to evaluate the new code, and HAP simulation output quantities are compared with results from established DSMC codes and with available experimental data. A grid refinement study is employed to quantify spatial discretization errors for the flat plate flow, and to demonstrate error reduction through a proposed collision probability modification. HAP simulation results are also compared with experimental data for a three-dimensional test case involving a hypersonic flow over a blunted cone. Finally, conclusions are presented, and planned areas of future work are discussed.

## II. Description of DSMC Implementation

A uniform Cartesian grid is employed in HAP for one, two or three-dimensional simulations, and zero-dimensional relaxation calculations may be performed as an additional option. The code can be run in either steady state mode, with time-averaged sampling between a user-designated transient startup time and simulation termination, or in unsteady mode, with multiple short sampling periods. For either mode, results are automatically output during the simulation in a commonly used visualization software format, and multiple zones are employed in unsteady mode for use in generating animations. Output quantities include various macroscopic flowfield variables, surface properties, and surface-integrated forces, moments and heat transfer. In zero-dimensional simulations of homogeneous relaxation problems, nonequilibrium initial conditions may be specified as the superposition of multiple Maxwellian velocity distributions, and the resulting velocity distribution at various elapsed times can be viewed in one, two or three dimensions. Multiple inflow and wall boundary conditions can be used, and immersed bodies may be included within the rectangular (in two dimensions) or cuboidal (in three dimensions) simulation domain. One-dimensional simulations can be performed using Bird's approximate stagnation streamline modeling technique.<sup>2</sup> Full restart capabilities allow calculations to be stopped, started and modified during a simulation with no undesired impact on simulation results.

### A. Physical Models and Algorithms

Cartesian ray tracing routines are used to move particles through the grid during each time step. These routines involve tracing a particle's trajectory to the nearest cell face, then moving the particle to the trajectory-face intersection point if the associated distance is less than the product of the time step interval and the particle speed. This type of procedure may be somewhat less efficient than simply moving each particle to some final position at the end of the time step before re-indexing the particle to a new cell.<sup>10</sup> However, the ray tracing technique greatly simplifies calculations involving immersed bodies, and should ease future implementation of parallel domain decomposition and time step adaptation. As in other DSMC codes,<sup>6</sup> ray tracing is performed for a given particle only if that particle is found to exit the assigned cell or impact a solid boundary during the current time step; otherwise more efficient procedures are utilized. If cell residence times are much larger than the time step interval, as is generally the case, then any efficiency reduction due to ray tracing should be small.

Physics models implemented in HAP include the no time counter (NTC) scheme of Bird for collision pair selection, the variable hard sphere (VHS) model for collision dynamics, the Maxwell model for gas-surface interaction, and the harmonic oscillator model for discretized vibrational energy distributions in a vibrationally excited diatomic gas.<sup>2</sup> Simulations can be run for a gas mixture involving an arbitrary number of species. The Larsen-Borgnakke<sup>13</sup> scheme has been implemented for continuous rotational-translational and quantized vibrational-translational energy exchange. As optional alternatives to the NTC collision model, the majorant frequency scheme (MFS) of Ivanov et al.<sup>9</sup> and the scheme of Baganoff and McDonald<sup>14</sup> have also been implemented.

In order to efficiently organize particle information while avoiding excessive memory requirements, HAP employs both cell-based and global dynamic arrays of pointers to particle data. When a new particle either enters the simulation domain or enters a new cell, the memory address for this particle is added to the next unused element in the associated pointer array. Likewise, when a particle exits the cell or the simulation domain, the last used element in the associated pointer array is moved to the position previously occupied by a pointer to this particle, and a variable indicating the number of used elements in the array is decremented by one. The size of both cell-based and global particle pointer arrays is incremented, as necessary, in large blocks; the default block size is 100 for cell-

based arrays and 100,000 for the global array. This use of memory blocks balances a desire to avoid unnecessary memory use with a competing desire to minimize the frequency of computationally expensive memory reallocation for large dynamic arrays. To further improve computational efficiency by reducing memory allocation, the memory location for any particle which exits the simulation domain is added to a linked list of unused particles. The same memory is then assigned, when needed, to a newly generated particle at an inflow boundary.

## B. Collision Selection

Although HAP is currently limited to a uniform user-defined cell size, transient adaptive subcell and nearest neighbor procedures are used to select collision pairs, if possible, with sufficiently small separation distances to meet standard DSMC guidelines<sup>2</sup> or some criterion based on the ratio of the mean collision separation (MCS) to the mean free path.<sup>2</sup> If possible, efficient  $O(N)$  procedures are utilized to find collision partners from within the same subcell; otherwise a more expensive nearest neighbor search is performed.

As a first step in collision operations during a given time step, NTC or an alternative scheme is used to determine the number of potential collision pairs in a cell. Subcell dimensions within this cell are then computed from the local hard sphere mean free path and a user-designated ratio of subcell size to mean free path. Subcell dimensions may vary among different areas of a cell due to intra-cell variation in the hard sphere mean free path, as described below in Section III.B. Next, all particles are sorted into subcells, using a rapid sorting routine also described below. A prescribed number of particles – equal to the number of potential collision pairs – are then randomly selected, and are paired if possible with previously unpaired particles located within the same subcell. If no unpaired particles are available, or if the only remaining unpaired particle in the same subcell was the last collision partner for the first particle in the pair, then a nearest neighbor from outside this subcell is selected as the second particle in the pair. Finally, once the required number of potential collision pairs has been selected, probabilistic procedures (as outlined below for the NTC scheme) are used to select which of these pairs will collide. Post-collision resampling routines are then performed for all selected pairs, in a manner which is consistent with the VHS model and which enforces momentum and energy conservation.

For nearest neighbor selection as for subcell-based pairing, no two particles are paired together if each of these particles most recently collided with the other. As described by Bird,<sup>8</sup> prevention of repeated collisions between the same two particles is expected to improve overall simulation accuracy, and is particularly important for error reduction when either subcells or nearest neighbor collisions are employed.

As proposed by Macrossan<sup>15</sup> for improved efficiency, the nearest neighbor selection procedure is performed using a randomly ordered array of unselected particles within the cell, and the first particle in this array for which the separation distance is smaller than some tolerance is chosen as the collision partner. For consistency with subcell procedures, the tolerance distance is set in HAP to equal the subcell size. If no particles in this array meet the separation distance criterion, then the nearest neighbor is selected.

In a further modification to nearest neighbor collision selection, a maximum of 30 randomly selected particles are considered as possible collision partners for each particle which requires nearest neighbor identification. This modification was proposed by Bird and by Macrossan<sup>15</sup> to avoid costly  $O(N^2)$  selection operations in cells which contain a large number of particles, and has been employed in a recent study by Gallis and Torczynski.<sup>16</sup> As an additional option available in HAP, such limited-search nearest neighbor procedures may be performed, in place of subcell based selection routines, to select every collision partner. Note that the number 30 is a rough transition point, identified by Bird et al.,<sup>17</sup> above which nearest neighbor procedures become more expensive than subcell-based collision pairing. Also note that limited searches for collision partner selection, due to either some acceptance distance tolerance or random exclusion of particles within the cell, have previously been called “near neighbor” selection, but this term is not used here in order to avoid confusion.

## C. Cut Cell Implementation

Two different options, analytical shapes and externally defined triangulated surfaces, are available in HAP for inclusion of immersed solid bodies within the rectangular or cuboidal simulation domain. For either option, cut cells are employed, with each cut cell containing one or more planar surface elements.

Analytically defined shapes can be used for either two or three-dimensional simulations, and – in contrast to similar capabilities in other DSMC codes<sup>2,9</sup> – these shapes are approximated by a series of planar faces, with one such face per cut cell. This allows relatively simple treatment of complex shapes, and greatly simplifies the process of modifying the code for use with shapes that are not currently available. Monte Carlo procedures for determining cut face properties are performed at simulation startup. For a two-dimensional simulation, these procedures are as follows: First, a large number of random points are generated in each cell. If some, but not all, of these points are within the body, then the cell is designated as a cut cell. Dimensions are determined for the smallest possible

rectangle which includes all random points which are outside the body, and corresponding dimensions are determined for a rectangle including all points inside the body. The  $x$  and  $y$  coordinates of the surface normal vector in this cell are then proportional to the  $y$  and  $x$  coordinate dimensions, respectively, of a third rectangle defined by the intersection region of the first two rectangles. The sign of each normal vector coordinate is determined by checking whether points along axis-aligned lines through the center of the intersection rectangle are inside or outside the body.

For a three-dimensional simulation, a somewhat more complicated procedure is required. In this case, two cuboids containing all random points inside and outside the body are found, and similar procedures as used for a two-dimensional simulation are employed in the central plane of the resulting intersection cuboid. In a three-dimensional simulation involving an analytically defined curved surface, small gaps may exist between planar cut faces in neighboring cells; for simplicity these gaps are treated as grid-aligned surface boundary faces.

While routines used to evaluate planar cut face properties are relatively complex and computationally expensive, these routines are performed only once at simulation startup, and require only that any point in the simulation domain can be established as either inside or outside the solid body. This requirement contrasts with analytical shape capabilities in other DSMC codes, for which expressions defining the surface (as opposed to the volume) of a solid body are required, and for which implementation of new shapes may be complicated and time consuming.<sup>2</sup> As implemented in HAP, the user may choose from a variety of two and three-dimensional shapes including cylinders, spheres, ellipsoids, cuboids, spherically blunted cones, and generic lifting body geometries.

For simulation of three-dimensional flows over arbitrary triangulated surfaces, an optional input file in the ASCII stereo-lithography (STL) format may be used in HAP. STL is a standard format compatible with most commercial CAD packages, and allows curved surfaces to be approximated by a set of contiguous triangular facets. At startup of a HAP simulation, each STL facet is linked to one or more cut cells by generating a large number of random points along the facet and identifying the cell in which each point is located. During particle movement routines, any particle located in a cut cell is considered for collisions with all facets linked to that cell.

#### **D. User Interface**

One main attribute of HAP, in contrast to some other DSMC codes intended for general rarefied flow simulation, is an extremely simple, user-configurable and user-friendly interface. The code employs only a single input file, which includes input parameters for global simulation quantities, flow properties, gas species properties and flowfield geometry. These parameters can be rearranged within the input file as desired, and most parameters may be excluded from the file for use with default parameter values.

Uniform numerical weight and time step interval values are utilized, but do not need to be specified directly in the input file. (Implementation and testing of time step adaptation capabilities is currently in progress.) The time step interval is calculated from a user-provided ratio of the time step to the mean collision time at inflow or ambient conditions, with automatic adjustments based on the mean particle transit time across each cell. In contrast to the time step interval, the range of appropriate values for the nondimensional time step input parameter should be relatively problem independent. If the value of this input parameter results in an overly large time step anywhere in the simulated flowfield, then the user is alerted by means of the number of “bad collisions” which is periodically output to the command window and appended to a log file. Here bad collisions are defined as those with a collision probability greater than 1, after adjusting for a maximum of one collision per particle per time step. For additional confirmation that temporal discretization errors are acceptably small, the ratio of the time step interval to the local mean collision time is also included in a field quantity output file. Similarly, the numerical weight (i.e. the number of atoms represented by each simulated particle) is calculated automatically from a user-designated number of particles per subcell based on inflow conditions. This number should generally range between around 2 to 20, and an overly small number can be detected via the ratio of mean collision separation (MCS) to mean free path, as found in the field quantity output file.

An additional numerical parameter – the number of time steps to reach steady state – must still be specified directly by the user for steady state flow simulation. However, the user can monitor transient characteristics in the flow by observing changes in the total numbers of particles, face crossings and collisions per time step; all three quantities are included in a log file. If the flow has not yet reached steady state at initiation of sampling routines, then the user can alter the number of startup time steps and restart the simulation with no loss in accuracy or efficiency.

In addition to a simple user interface and other features described above, one further attribute of HAP is a compact source file. The source code is contained within a single file, which consists of approximately 6500 lines including extensive comments. Several different functions are used within this file for various DSMC operations (e.g. flowfield initialization, particle movement, collisions, sampling) and output file generation, with about 20% of

the source code devoted to the second category. An additional 20% is devoted to automatic generation and initialization of cut cell cells for analytically defined two and three-dimensional surfaces, and to pre-processing for triangulated surface geometries defined in an optional STL input file.

### III. Uniform Cartesian Grid Methodology

As mentioned above, HAP employs a uniform Cartesian grid without capabilities for grid adaptation. While this approach has significant advantages associated with reduced code complexity, grid adaptation is required in traditional DSMC algorithms for accurate and efficient simulation of flows involving a wide density range.<sup>2</sup> The most important historical reason for the use of grid adaptation is to limit the separation distance between colliding particles. Accurate simulation of the Boltzmann equation requires that the mean collision separation (MCS) be considerably smaller than the local mean free path,<sup>8</sup> and this condition is violated if collision partners are frequently selected from opposite sides of a cell which is large in comparison to the mean free path. However, the MCS argument for requiring grid adaptation no longer applies when either transient adaptive subcells<sup>8</sup> or nearest neighbor collisions<sup>18</sup> (with a sufficient particle population and appropriate means of avoiding repeated collisions) are employed, as these techniques allow MCS values which are nearly independent of cell size.

There are additional reasons for use of grid adaptation which deserve some discussion: First, higher resolution of simulation results may be desired in regions of relatively high density, and output of field quantities with comparatively low scatter in low density regions can require spatial averaging over larger cell volumes. Although no apparent alternatives to grid adaptation exist for adjustment of output resolution in field quantities, it should be noted that if surface fluxes are the main quantities of interest, then cell size independence can be accomplished by decoupling surface geometry from the cell data structure. In this approach, as used in HAP for three-dimensional triangulated surfaces defined in an optional STL format input file, several surface facets can be located within a single cell, or one facet can extend over several cells.

As an additional argument for grid adaptation in DSMC, the calculation of DSMC collision probability is based on assumptions of uniform density in each cell, and large density variation within a cell can have a significant impact on the calculated collision frequency. Errors associated with uniform density assumptions in DSMC collision probability are discussed in detail below.

As a final reason for grid adaptation, the local mean free path can vary across comparatively large cells in an unadapted grid, and the required size of DSMC subcells to meet a MCS based criterion<sup>8</sup> for simulation accuracy may therefore vary within each cell. Although nearest neighbor pairing can be utilized as an alternative to adaptive subcells, the use of nearest neighbor collision selection may be prohibitively expensive in cells which are significantly larger than the local mean free path and which contain a large number of particles. Limited search “near neighbor” algorithms, as recently employed by Gallis and Torczynski<sup>16</sup> and by Macrossan<sup>15</sup> to improve efficiency relative to standard nearest neighbor procedures, can potentially provide a similar balance of efficiency and accuracy as subcell procedures in such large cells. However, as demonstrated by LeBeau et al,<sup>18</sup> the MCS value depends on the number of particles considered in each search. If a large number of particles per search are required to meet some MCS criterion, then subcell based collision pairing may be more appropriate.

While Cartesian transient adaptive subcell procedures are an efficient means of achieving sufficiently small MCS in large cells which contain a large number of particles,<sup>8</sup> the desired subcell size may differ considerably across a cell due to variation in flowfield quantities over the cell volume. Some error or efficiency reduction is therefore associated with the use of a uniform subcell size within each cell, as is employed in existing Cartesian implementations of the transient adaptive subcell technique.

Although some of the above arguments for using grid adaptation can be resolved for the simpler uniform grid approach in HAP by means of preferential pairing of nearby particles, and by decoupling cell and surface data structures, the last two arguments – errors in collision probability, and variation in appropriate subcell size across a cell – require new modifications to DSMC procedures. These modifications are presented as follows.

#### A. Gradient-Based Modification to Collision Probability

In order to demonstrate the error in DSMC binary collision probabilities due to internal density variation within a cell, we consider the collision frequency in a simple gas as calculated using the no time counter (NTC) scheme of Bird.<sup>2</sup> The number of potential collision pairs  $N_{coll}$  is computed as

$$N_{coll} = \left[ \frac{1}{2} \frac{WN(N-1)}{V} (\sigma g)_{\max} \Delta t + R \right] \quad (1)$$

where the operator  $\lfloor \rfloor$  rounds to the nearest smaller integer,  $R$  is a random number in  $[0,1]$ ,  $W$  is the numerical weight (i.e. the number of atoms or molecules represented by each particle),  $V$  is the cell volume,  $\Delta t$  is the time step interval,  $N$  is the current number of particles in the cell, and  $(\sigma g)_{\max}$  is the maximum product of collision cross section and incident relative speed for all collision pairs in the cell over a large number of time steps. Each of the  $N_{coll}$  potential collision pairs is then selected to experience a collision with probability

$$P_{jk} = \frac{\sigma_{jk} g_{jk}}{(\sigma g)_{\max}} \quad (2)$$

where  $j$  and  $k$  are the particle indices, and  $\sigma_{jk}$  and  $g_{jk}$  are the collision cross section and relative speed respectively. As  $N$  follows a Poisson distribution, the expected mean value of  $N(N-1)$  in Eq. (1) should equal  $\bar{N}^2$  where  $\bar{N}$  is the value of  $N$  averaged over a large number of time steps.<sup>8</sup> The mean collision frequency  $F(\mathbf{x})$  at a given location  $\mathbf{x}$  in this cell is therefore

$$F(\mathbf{x}) = \frac{2}{\bar{N}\Delta t} \langle N_{coll} P_{jk} \rangle = \frac{W\bar{N}}{V} \langle \sigma_{jk} g_{jk} \rangle \quad (3)$$

where the operators  $\langle \rangle$  indicate an average over all velocity space, weighted by the velocity distribution  $f$  and the probability density  $p$  in physical space, for all potential collision partners  $j$  and  $k$ . Thus,

$$\langle \sigma_{jk} g_{jk} \rangle = \iiint \sigma(\mathbf{c}_j, \mathbf{c}_k) |\mathbf{c}_j - \mathbf{c}_k| f(\mathbf{c}_j, \mathbf{x}_j) f(\mathbf{c}_k, \mathbf{x}_k) p(\mathbf{x}_j, \mathbf{x}) p(\mathbf{x}_k, \mathbf{x}) d\mathbf{x}_j d\mathbf{x}_k d\mathbf{c}_j d\mathbf{c}_k \quad (4)$$

where  $\mathbf{c}_j$  and  $\mathbf{c}_k$  represent particle velocities,  $\mathbf{x}_j$  and  $\mathbf{x}_k$  represent particle positions, and the integrals in Eq. (4) are carried out over all phase space for both  $j$  and  $k$ .

The ratio  $W\bar{N}/V$  in Eq. (3) is the gas number density averaged over the cell volume. If the cell is large in comparison to the local mean free path, then both the number density and the velocity distribution may be strong functions of position  $\mathbf{x}$  in the cell. If we further assume that the MCS is much smaller than the mean free path, then  $f(\mathbf{c}_j, \mathbf{x}_j)p(\mathbf{x}_j, \mathbf{x}) \approx f(\mathbf{c}_j, \mathbf{x})\delta(\mathbf{x}_j - \mathbf{x})$  and  $f(\mathbf{c}_k, \mathbf{x}_k)p(\mathbf{x}_k, \mathbf{x}) \approx f(\mathbf{c}_k, \mathbf{x})\delta(\mathbf{x}_k - \mathbf{x})$  in Eq. (4), where  $\delta$  is the Dirac delta. It follows that

$$\langle \sigma_{jk} c_{jk} \rangle \approx \iint \sigma(\mathbf{c}_j, \mathbf{c}_k) |\mathbf{c}_j - \mathbf{c}_k| f(\mathbf{c}_j, \mathbf{x}) f(\mathbf{c}_k, \mathbf{x}) d\mathbf{c}_j d\mathbf{c}_k \quad (5)$$

and  $\langle \sigma_{jk} g_{jk} \rangle$  should vary in a physically appropriate way as a function of  $\mathbf{x}$ . However, Eq. (3) contains no adjustment to the local number density  $n(\mathbf{x})$  if  $n(\mathbf{x}) \neq W\bar{N}/V$ . Some error in  $F$  is therefore expected for cells which are large relative to the mean free path and located in high density gradient regions. In particular, if large portions of a cell are consistently unpopulated with particles, then  $n(\mathbf{x}) > W\bar{N}/V$  in populated regions, and  $F(\mathbf{x})$  will be unphysically low. More generally, if  $\nabla n \neq 0$  then  $F(\mathbf{x})$  should decrease with increasing cell size, and unphysical preference in collision selection is given to particles in portions of the cell for which  $n(\mathbf{x}) < W\bar{N}/V$ . Note that, although the NTC scheme is used in the above discussion, the same error due to density gradients is expected for other DSMC collision schemes including the majorant frequency scheme<sup>9</sup> (MFS) and the scheme of Baganoff and McDonald.<sup>14</sup>

As a means of correcting for this error, we first assume that  $\nabla n$  is uniform over the cell volume. Thus,

$$n(\mathbf{x}) = n_c + \nabla n \cdot (\mathbf{x} - \mathbf{x}_c) \quad (6)$$

where  $n_c = W\bar{N}/V$  is the cell-averaged number density and  $\mathbf{x}_c$  is the geometric center of the cell. The time-averaged center of mass  $\bar{\mathbf{x}}_{cm}$  of all particles in the cell is then

$$\bar{\mathbf{x}}_{cm} = \mathbf{x}_c + \frac{1}{n_c V} \int n(\mathbf{x})(\mathbf{x} - \mathbf{x}_c) d\mathbf{x} \quad (7)$$

By substituting Eq. (6) into Eq. (7), and assuming a Cartesian cell of equal length  $\Delta x$  in each coordinate direction, we can express  $\nabla n$  as a simple function of the vector difference ( $\bar{\mathbf{x}}_{cm} - \mathbf{x}_c$ ).

$$\nabla n = \frac{12n_c}{(\Delta x)^2} (\bar{\mathbf{x}}_{cm} - \mathbf{x}_c) \quad (8)$$

To correct the binary collision probability for effects of spatial density variation, so that the collision frequency  $F(\mathbf{x})$  in Eq. (3) scales with  $n(\mathbf{x})$ , we multiply the probability  $P_{jk}$  in Eq. (2) by the ratio  $n(\mathbf{x}_{jk})/n_c$ , where  $\mathbf{x}_{jk}$  is the center of mass of the collision pair. From Eqs. (2), (6) and (8), we find

$$P_{jk} = \frac{\sigma_{jk} g_{jk}}{(\sigma g)_{\max}} \left( 1 + \frac{12}{(\Delta x)^2} (\mathbf{x}_{jk} - \mathbf{x}_c) \cdot (\bar{\mathbf{x}}_{cm} - \mathbf{x}_c) \right) \quad (9)$$

As implemented in HAP, this correction is applied to all binary collision probabilities, in order to reduce the error associated with density variation in cells which are large compared to the local mean free path. For simple application of Eq. (9) to a gas mixture, effects of species-based properties (e.g. collision cross section) are neglected and  $\bar{\mathbf{x}}_{cm}$  is evaluated without biasing to species mass.

In Eq. (9), the time-averaged center of particle mass in a cell  $\bar{\mathbf{x}}_{cm}$  is calculated by first evaluating the instantaneous center of mass  $\mathbf{x}_{cm}$  for all  $N$  particles currently in the cell, then performing an exponential moving average operation

$$(\bar{\mathbf{x}}_{cm})_t = \theta (\mathbf{x}_{cm})_t + (1 - \theta) (\bar{\mathbf{x}}_{cm})_{t - \Delta t} \quad (10)$$

where  $\theta$  is a constant much less than one ( $\theta = 0.01$  is used in this work) and  $t$  indicates the elapsed simulation time. More complicated and potentially more accurate time-averaging procedures, such as the subrelaxation technique of Sun and Boyd,<sup>19</sup> may be used in place of Eq. (10). Note that, for accurate simulation of an unsteady flow,  $\theta$  should be much larger than the ratio of  $\Delta t$  to the minimum characteristic transient time scale.

## B. Nonuniform Transient Adaptive Subcells

As discussed above, transient adaptive subcells are an efficient means of physically appropriate collision pairing in cells which are large relative to the mean free path. In such cells, a large number of particles may be required for sufficiently small MCS, and subcell indexing is generally much more efficient in these cells than the  $O(N^2)$  operations associated with nearest neighbor collision selection. However, in the presence of a strong density gradient, the mean free path may vary considerably over the volume of a cell, and division of the cell into subcells based on the cell-averaged mean free path may lead to some reduction in accuracy. Alternatively, the use of overly small subcells – based on estimates of the minimum mean free path anywhere in a cell – can reduce simulation efficiency by creating more subcells than necessary in regions of relatively large mean free path. An excessive number of subcells tends to increase the frequency of expensive searches for collision partners outside the same subcell, using either complicated subcell-based search logic<sup>8</sup> or (as employed in HAP) nearest neighbor routines. Thus, a somewhat improved balance between efficiency and accuracy may be possible by clustering subcells in portions of a cell where the local mean free path is smaller than the cell-averaged value. One procedure for such nonuniform subcell indexing is described as follows.

First, we desire for the subcell length to be some fraction  $C$  of the local hard sphere mean free path, which is inversely proportional to the local number density  $n(\mathbf{x})$ . For simplicity, we assume a simple gas; effects of concentration gradients in a gas mixture are neglected. We next assume, as above, that the density gradient  $\nabla n$  is uniform over the cell volume. The subcell index  $i$  in the  $x$  coordinate direction can then be based on uniform subcell spacing in some alternate coordinate system for which the location  $u(x)$  is related to the global coordinate  $x$  by  $du/dx = n(x, y_c, z_c)$  and  $u(x_c - 1/2\Delta x) = 0$ . Here  $(x_c, y_c, z_c) = \mathbf{x}_c$  is the cell center location and  $\Delta x$  is the cell length. Hence  $du = n(x, y_c, z_c) dx$ , and from Eqs. (6) and (8), we find

$$u(x) = \int_{x_c - \frac{1}{2}\Delta x}^x \left( n_c + \frac{12n_c}{(\Delta x)^2} (\bar{x}_{cm} - x_c)(x' - x_c) \right) dx' \quad (11)$$

where  $\bar{x}_{cm}$  is the  $x$  coordinate of the time-averaged center of mass of particles in the cell  $\bar{x}_{cm}$ . The corresponding subcell index value is then

$$i(x) = \left\lfloor (i_{max} + 1) \frac{u(x)}{u\left(x_c + \frac{1}{2}\Delta x\right)} \right\rfloor \quad (12)$$

After integration of Eq. (11) and substitution into Eq. (12), we can express the index  $i \in [0, i_{max}]$  in the  $x$  direction for a given particle location  $\mathbf{x} = (x, y, z)$  as

$$i(x) = \left\lfloor (i_{max} + 1) \xi(x) \left( 1 + \frac{6}{\Delta x} (\bar{x}_{cm} - x_c) (\xi(x) - 1) \right) \right\rfloor \quad (13)$$

where

$$\xi(x) \equiv \frac{1}{\Delta x} (x - x_c) + \frac{1}{2} \quad (14)$$

The maximum index value  $i_{max}$  in Eq. (13) is given by

$$i_{max} = \left\lfloor \frac{\Delta x}{C \langle \lambda \rangle_x} \right\rfloor \quad (15)$$

where  $\langle \lambda \rangle_x$  is the hard sphere mean free path averaged over a line segment through the cell center for which  $x \in [x_c - \frac{1}{2}\Delta x, x_c + \frac{1}{2}\Delta x]$ . The value of  $\langle \lambda \rangle_x$  is calculated through the approximate expression

$$\langle \lambda \rangle_x \approx \frac{1}{\sqrt{2}\sigma n_c} \left( 1 + 12 \left( \frac{\bar{x}_{cm} - x_c}{\Delta x} \right)^2 \right) \quad (16)$$

where  $\sigma$  is a reference collision cross section. A derivation of Eq. (16) is included in the Appendix.

To further improve code efficiency by avoiding insufficient particle populations within each subcell, the maximum subcell index  $i_{max}$  can be limited by

$$i_{max} \leq \max \left\{ \left\lfloor \left( \frac{N}{N_{min,sub}} \right)^{1/D} \right\rfloor - 1, 0 \right\} \quad (17)$$

where  $D$  is the number of simulation dimensions and  $N_{min,sub}$  is the minimum desired average number of particles per subcell. ( $N_{min,sub} = 2$  is used in this work.) If  $C \approx 1$  in Eq. (15), then the limiting condition in Eq. (17) is likely imposed only in cells where too few particles exist to satisfy standard DSMC guidelines for the MCS, and where some reduction in numerical weight  $W$  may be needed for accurate simulation. However, the condition in Eq. (17) can be particularly useful if  $C \ll 1$ , as this allows MCS  $\ll \langle \lambda \rangle_x$  in cells which contain a sufficiently large number of particles, but avoids excessive use of inefficient inter-subcell selection procedures in cells which contain too few particles for the desired MCS value.

In multidimensional simulations, additional subcell indices are required in other coordinate directions. For example, in a three-dimensional simulation, a unique subcell identification number for the particle location



$\mathbf{x} = (x, y, z)$  can be given as  $i(x) + (i_{max}+1)j(y) + (i_{max}+1)(j_{max}+1)k(z)$ , where  $j(y) \in [0, j_{max}]$  and  $k(z) \in [0, k_{max}]$  are the subcell indices in the  $y$  and  $z$  directions respectively.

By utilizing the proposed subcell indexing procedure, we allow for smaller subcells in regions of increased density and reduced mean free path. For relatively large cells in the presence of a density gradient, subcell lengths may vary significantly along different coordinate directions. However, if  $\nabla n = 0$ , then Eq. (13) reduces to

$$i(x) = \left\lfloor (i_{max} + 1) \left( \frac{1}{\Delta x} (x - x_c) + \frac{1}{2} \right) \right\rfloor \quad (18)$$

as is used for particle indexing to Cartesian subcells of uniform length.

Both Eqs. (9) and (13), which are employed in HAP for collision selection and subcell indexing respectively, assume a Cartesian cell with uniform length  $\Delta x$  in each coordinate direction. In a cut cell, for which part of the cell is inside a solid body, approximate corrections for the density gradient are enabled by using as  $x_c$  the geometric center of the portion of the cell which is outside the body. This location may be found for each cut cell at simulation startup through a simple Monte Carlo sampling technique, which involves generating a large number of random points throughout the cell volume and discarding any points within the body.

Note that the proposed nonuniform subcell technique follows from an initial assumption that the subcell size should be based on the local mean free path, not solely on the desired number of particles per subcell as in other implementations of the transient adaptive subcell scheme.<sup>8,16</sup> In adapting subcells to the mean free path, we attempt to reduce the MCS only as much as necessary to meet a desired maximum ratio of the MCS to the mean free path, while minimizing the frequency of expensive searches for neighboring particles outside the same subcell. For example, if subcells are scaled in proportion to the mean free path, and if a cell contains enough particles to satisfy some MCS criterion while populating each subcell with a large number of particles, then nearly every collision pair can be efficiently selected from within the same subcell. Under these conditions, computational expense for collision selection scales approximately linearly with the number of particles per cell, and overall simulation efficiency should vary little if the cell size is further increased. It follows that subcell adaptation to the local mean free path is ideally suited to a uniform grid approach, as relatively large cells can be used without significantly impacting either MCS values or simulation efficiency. Similar arguments are used to motivate the use of subcell sizes based on the mean free path in the MONACO DSMC code.<sup>6</sup>

One additional application of the proposed nonuniform subcell technique deserves some mention: As a first step in collision procedures, Eqs. (13) through (17) can be used to efficiently arrange particles into subcells for a cell which is much larger than permitted by standard DSMC guidelines. NTC, MFS or some other scheme can then be independently applied within each subcell, to select collision pairs from among the particles assigned to the same subcell. Eq. (1) is evaluated separately for each subcell, where  $N$  is replaced by the number of particles per subcell,  $V$  is replaced by the subcell volume, and the quantity  $(\sigma_r)_{max}$  is stored as a single value for the entire cell. (Nearest neighbor routines may also be employed to further reduce the MCS, where only particles within the same subcell are considered as potential nearest neighbors.) This approach should eliminate the need for the collision probability modification given as Eq. (9), and should avoid any accuracy loss associated with the uniform gradient assumption in that equation. There are, however, significant drawbacks of this approach relative to the nonuniform subcell approach described above. In particular, additional complexity is required to evaluate the volume of subcells which intersect surface elements for a solid body, and collision partners cannot be selected from among different subcells.

#### IV. Initial Code Evaluation

For initial evaluation, HAP is applied to a two-dimensional hypersonic flow of  $N_2$  over a flat plate with a sharp leading edge. The plate is aligned with the freestream direction, the freestream Mach number is 20.2, and the Knudsen number based on the total length of the plate is 0.015. The plate surface is diffusely reflecting at a temperature of 290 K, and the freestream temperature is 13.32 K. Using the VHS collision model with parameters recommended for  $N_2$  by Bird,<sup>2</sup> we calculate a freestream mean free path is 1.688 mm. This case is based on tests by Lengrand et al.<sup>20</sup> and Heffner et al.<sup>21</sup> in the SR3 low density wind tunnel at CNRS in Meudon, France.

A corresponding simulation is performed with the DSMC code MONACO,<sup>6</sup> using the same grid, boundary conditions, physical models and numerical parameters. Results are compared between the two simulations in order to highlight any discrepancies which may indicate potential errors in HAP. Published DSMC results of Heffner et al.,<sup>21</sup> as well as experimental data of Heffner et al. and Lengrand et al.,<sup>20</sup> are also used for comparison.

HAP and MONACO simulations employ a uniform grid with 128 by 48 square cells of length 1 mm, and with a uniform inflow boundary located 16 cells upstream of the plate leading edge. The time step interval  $\Delta t$  is set to  $3.102 \times 10^{-7}$  s, which is approximately 0.06 times the minimum mean collision time at steady state. The numerical weight  $W$  is set to  $2.3225 \times 10^{12}$  for an average of 160 particles per cell in the freestream. Time averaged sampling is performed over 18000 time steps, following a transient startup period of 2000 steps. For reference, various simulation parameters are shown in Table 1. The HAP simulation requires about 5.3 hours of CPU time on a single 1.9 GHz processor, and includes  $1.06 \times 10^6$  particles at steady state.

Figures 1 through 4 show isocontour lines for various field quantities from both HAP and MONACO simulations. Coordinates are normalized by the freestream mean free path, and the coordinate system origin is located at the plate leading edge. HAP simulation results are displayed as multi-colored solid lines, while results from the MONACO simulation are shown as black dashed lines. Figures 1, 2, 3 and 4 show respectively contours of Mach number, number density normalized by the freestream value, translational temperature, and rotational temperature. Relatively complex characteristics are observed in the figures, including a highly diffuse oblique shock which forms within a smooth compression region near the leading edge, a low density expansion region within the post-shock area near the surface, and significant rotational temperature lag behind the shock. In all four figures, virtually no noticeable differences are found between results from the two codes; the one exception is a small discrepancy in rotational temperature near the plate surface around  $x/\lambda_\infty = 45$ . Although no clear explanation for this discrepancy is apparent, the local gradient in rotational temperature is small, and the relative difference of well under 1% is within the likely range of statistical scatter.

In Fig. 5, normalized density values  $n/n_\infty$  from the HAP and MONACO simulations are extracted along a transverse plane at  $x/\lambda_\infty = 44.55$ . Experimental data from electron beam fluorescence measurements of Lengrand et al.<sup>20</sup> are also displayed for comparison. As in Fig. 2, excellent agreement is found between density values from the two simulations, and any noticeable differences are within statistical scatter. In contrast, large discrepancies are observed between experimental and simulation data points, with the experiment showing a roughly 30% increase in distance between the plate and the shock center location. Note however that reasonably good agreement is found in the density along the plate and in the maximum density within the shock. Differences between experimental and DSMC density values can be mainly attributed to uncertainty and modeling approximations in gas-surface interaction, high sensitivity of experimental results to the plate angle of attack, and nonuniform freestream conditions in the SR3 experimental facility.<sup>22</sup>

Figures 6 and 7 show profiles of bulk velocity and temperature, respectively, extracted from HAP and MONACO results along the same transverse plane as in Fig. 5. Transverse and longitudinal velocity components in Fig. 6 are normalized by the inflow bulk velocity, and in Fig. 7 translational and rotational temperatures are normalized by the inflow temperature. As in Figs. 1 through 4, no statistically significant disagreement is found between HAP and MONACO results in Figs. 6 and 7.

In Fig. 8, the pressure coefficient  $C_p$  is plotted along the plate surface from HAP and MONACO simulation results, with additional published DSMC results of Heffner et al.<sup>21</sup> and experimental data of Lengrand et al.<sup>20</sup> The pressure coefficient is defined as

$$C_p = \frac{P - P_\infty}{\frac{1}{2} \rho_\infty V_\infty^2} \quad (19)$$

where  $P$  is the surface-normal momentum flux, and  $P_\infty$ ,  $\rho_\infty$  and  $V_\infty$  are respectively the freestream pressure, mass density and bulk velocity magnitude. As in Figs. 1 through 7, excellent agreement is observed in Fig. 8 between MONACO and HAP results. However, relative to other DSMC results in Fig. 8, some small but noticeable underprediction is found in the Heffner et al. results, likely due in part to the use of a different set of parameter values by Heffner et al. for the VHS collision model. [1] Considerable disagreement is displayed in Fig. 12 between the experimental data points and all DSMC results, with consistently lower experimental  $C_p$  values for  $x/\lambda_\infty > 15$ . The dominant source of the discrepancy is unclear, but likely contributors include the potential error sources mentioned above in the discussion of Fig. 5. Note the sharp drop-off in  $C_p$  observed in both HAP and MONACO results near the downstream edge of the simulation domain. This trend is likely unphysical, and results from the suction effect of the DSMC vacuum outflow boundary within a small subsonic boundary layer region.

Figure 9 shows the shear stress coefficient, defined as the ratio of the surface tangential momentum flux to the freestream dynamic pressure  $\frac{1}{2} \rho_\infty V_\infty^2$ , from HAP and MONACO simulation results. As expected, no discrepancies are observed outside the range of statistical scatter.

Variation in the heat transfer coefficient  $C_h$  along the plate surface is shown in Fig. 10. This coefficient is defined as

$$C_h = \frac{q}{\frac{1}{2}\rho_\infty V_\infty^3} \quad (20)$$

where  $q$  is net surface energy flux. Results are displayed from HAP and MONACO simulations described above, along with the published DSMC results of Heffner et al.<sup>21</sup> and the experimental data of Heffner et al. and Lengrand et al.<sup>20</sup> For comparison, results are also plotted in Fig. 10 from an additional HAP simulation for which the thermal accommodation coefficient (TAC) is reduced from 1 to a potentially more realistic value of 0.8. All three sets of DSMC results for TAC = 1 show very good agreement, with no differences between HAP and MONACO values outside the level of statistical scatter. Significantly lower  $C_h$  values are found in the experimental results of both Heffner et al. and Lengrand et al., and considerable disagreement is also observed between the two experimental data sets. No dominant source of the discrepancies is apparent, but, gas-surface interaction is a potential contributing factor, as suggested by Heffner et al. and as indicated by the relatively good agreement between the experimental data of Heffner et al. and results from the HAP simulation with TAC = 0.8. Sensitivity to and uncertainty in the plate angle of attack is another likely contributor, and may partly account for the disagreement between the two sets of experimental data. Nonuniform freestream conditions<sup>22</sup> should also reduce the level of agreement between DSMC and experimental results, and may warrant future DSMC simulations which include much of the upstream expansion region in the SR3 facility.

## V. Evaluation of Error Reduction Due to Collision Probability Modification

In both HAP and MONACO simulations described above, the cell size is 0.6 times the freestream mean free path and is uniformly less than the local mean free path throughout the simulated flowfield. It is therefore expected that approximations associated with spatial averaging in density over the cell volume, as used to compute collision probabilities in the standard NTC scheme,<sup>2</sup> account for negligibly small errors in the output quantities of interest. However, from the logic outlined in Section III.A, the spatial discretization error in collision probabilities should increase in regions of high density gradient as the cell size is increased. Likewise, the proposed modification to the collision probability, as implemented in HAP, should have a progressively greater impact on simulation results when larger cells are used. In order to quantify the error due to density averaging in collision probabilities, and to demonstrate the effectiveness of the proposed modification in reducing this error, a series of additional HAP simulations are run for the same hypersonic flat plate flow presented above in Section IV. The ratio  $\Delta x/\lambda_\infty$  is varied by factors of two between 0.6 and 4.8, and simulations are performed both with and without the collision probability modification. The same values are used in all simulations for the numerical weight and time step interval, and transient adaptive subcell procedures should eliminate or greatly reduce any effects of cell size on the mean collision separation. It follows that the discretization error described in Section III.A should be isolated from other DSMC spatial and temporal discretization errors, and only this first error is varied between the different simulations.

Figures 11, 12 and 13 show respectively the variation in pressure, shear stress and heat transfer coefficients along the plate surface, from eight simulations employing different  $\Delta x/\lambda_\infty$  ratios and using either standard or modified collision probabilities. As shown in the three figures, only very small differences appear in the surface fluxes between results from all simulations employing the modified probabilities, while the standard DSMC simulations display considerably greater disagreement. Errors among the standard DSMC results are particularly noticeable for simulations with normalized cell sizes  $\Delta x/\lambda_\infty$  of 2.4 and 4.8, which significantly underestimate flux values over much of the plate surface relative to the  $\Delta x/\lambda_\infty = 0.6$  simulations. In addition, significant overestimates in both shear stress and heat flux are found at  $x/\lambda_\infty > 60$  for the  $\Delta x/\lambda_\infty = 4.8$  simulation utilizing standard DSMC collision probabilities. In comparing results between the  $\Delta x/\lambda_\infty = 4.8$  simulations with and without the collision probability modification, we find large errors in the standard DSMC simulation, and a considerable reduction in these errors when the proposed modification is used. Note that surface fluxes are typically among the output quantities most sensitive to errors in DSMC calculations, and that only very small errors (less than 3%) were observed in a comparison of field quantities between these simulations.

To further quantify the errors associated with spatial density averaging in collision probabilities, and to quantify the error reduction due to the proposed modification, the  $L_\infty$ -norm of the heat transfer coefficient is plotted as a function of normalized cell size  $x/\lambda_\infty$  in Fig. 14. This norm is calculated as the maximum difference between local  $C_h$  values and those for the  $\Delta x/\lambda_\infty = 0.6$  standard DSMC simulation, normalized by the local  $C_h$  value from the same

$\Delta x/\lambda_\infty = 0.6$  simulation. By assuming that differences in  $C_h$  values between modified and standard  $\Delta x/\lambda_\infty = 0.6$  simulations are primarily due to statistical scatter, we can designate the  $L_\infty$ -norm value for the modified  $\Delta x/\lambda_\infty = 0.6$  simulation as the minimum  $L_\infty$ -norm level above which statistically significant errors in  $C_h$  are detected. This minimum level is displayed as a dashed horizontal line in Fig. 14. Linear trend lines, based on least-squares fits, are also shown in Fig. 14 to highlight general trends among the two sets of data points.

In comparing results with and without the collision probability modification, we find a considerable increase in error with increasing  $\Delta x/\lambda_\infty$  for the standard DSMC simulations, and a much weaker dependence on cell size when modified collision probabilities are employed. In particular, the maximum local error in  $C_h$  is significantly less for the  $\Delta x/\lambda_\infty = 4.8$  simulation with modified probabilities than for the  $\Delta x/\lambda_\infty = 1.2$  standard DSMC simulation.

In Fig. 15, the L2-norm in  $C_h$  is plotted as a function of  $\Delta x/\lambda_\infty$  for simulations with and without modified collision probabilities. This norm is a measure of the root-mean-square error in local heat flux, relative to the standard DSMC simulation with  $\Delta x/\lambda_\infty = 0.6$ . The same general trends are observed in Fig. 15 as in Fig. 14, although a somewhat greater dependence on cell size is found for the L2-norm than for the  $L_\infty$ -norm in results from the modified probability simulations.

In both Figs. 14 and 15, a comparatively small – but still significant – increase in error for larger  $\Delta x/\lambda_\infty$  is found for the modified probability simulations. This increase can be primarily attributed to breakdown in the assumption of a uniform density gradient within each cell. Curvature of the density profile is clearly shown in Fig. 5, due in part to competing effects of compression within the shock, expanding flow in the post-shock region, and relatively stagnant flow along the wall. In Fig. 5, we find that approximations of a locally uniform density gradient should be particularly inaccurate for cells along the plate surface or near the location of maximum density within the shock. For accurate simulation using the proposed modification, it follows that the cell size may need to be limited by length scales, such as  $|\nabla n/\nabla \cdot \nabla n|$ , which are associated with the curvature of the density profile. Additional reductions in errors related to density averaging may also be possible by repeating the analysis in Section III.A under assumptions that  $\nabla n$  is a linear function of position within each cell, or by applying independent collision selection procedures in each subcell as suggested in Section III.B.

## VI. Comparison with Experiment for Hypersonic Flow Over a Blunted Cone

As an additional test case, we consider a three-dimensional hypersonic flow of  $N_2$  over a spherically blunted cone at a  $10^\circ$  angle of attack. This case is based on a series of experiments by Allegre et al. in the SR3 facility at CNRS,<sup>22,23</sup> and involves a subscale model of the Mars Pathfinder aeroshell. Geometry parameters include a cone angle of  $70^\circ$ , a base radius of 2.5 cm, and curvature radii of 1.25 cm and 1.25 mm at the nose and shoulder respectively. All of these parameters are specified in the HAP input file, for automatic generation of planar surface elements which comprise the aeroshell shape. The freestream Mach number is 20.2, the global Knudsen number (based on cone diameter) is 0.032, and the Reynolds number is 1420. Additional simulation parameters include a freestream temperature of 13.3 K, a wall temperature of 300 K, a wall thermal accommodation coefficient of 1, and collision numbers of 5 and 50 for rotational and vibrational energy respectively. The simulation is performed on a uniform Cartesian grid with  $3 \times 10^5$  cells and approximately  $2 \times 10^6$  particles at steady state. Time-averaged sampling is carried out over 22,000 time steps, following a transient startup period of 2000 steps. A CPU time of about 19 hours is required.

Figure 16 shows contours of surface heat flux and streamlines along the symmetry plane. (For reduced simulation expense, calculations were performed only on one side of the symmetry plane, and Fig. 16 includes a mirror image of the direct simulation results.) Some asymmetry in streamlines is observed along the pitching plane, with a stagnation point slightly below the center of the nose region. As expected, the heat flux is highest around the nose, and higher heat flux values are found on the windward side of the forebody than on the lee side. A local maximum in heat flux occurs around the windward shoulder, and a rapid drop-off in heat flux is observed in the strong expansion region along the downstream portion of the shoulder.

In Fig. 17, isocontour lines for normalized density  $n/n_\infty$  are shown from HAP simulation results, and from electron beam fluorescence measurements of Allegre et al.<sup>22</sup> Relatively good overall agreement is found, particularly within the forebody shock layer and in the wake expansion regions. Both simulation and experimental results show considerable asymmetry due to the nonzero angle of attack. However, significant discrepancies are observed between the two sets of results, with particularly noticeable differences within near-wake and off-axis shock layer regions. Other differences in Fig. 17 are due to freestream scatter in HAP contour lines at  $n/n_\infty = 1$ ; this results from the use of contour values at the designated inflow condition, and should not be considered either as an indication of excessive statistical scatter or as an example of disagreement with experimental data.

While several possible error sources contribute to the observed differences between HAP and experimental results in Fig. 17, one main probable contributor is the highly nonuniform freestream condition in the experiment. The freestream density distribution was characterized by Allegre et al. through density measurements in an empty test section, and showed considerable differences in both axial and transverse directions.<sup>22</sup> Unfortunately, a HAP simulation involving similarly nonuniform freestream conditions would require additional measurements for freestream velocity and temperature. Effects of freestream translational nonequilibrium or rotational freezing could also contribute to the differences shown in Fig. 17, and accurate simulation of inflow conditions at the test section may require that the simulation domain extend far upstream into the divergent nozzle region of the SR3 facility. For somewhat greater consistency, experimental density values in Fig. 17 are normalized by the local freestream density as measured in the empty test section. Another likely contributor to discrepancies within near-wake region is the presence of the sting in the experiment; exclusion of the sting in the simulation probably reduced the calculated density in this region and contributed to the disagreement observed in Fig. 17.

Variation in surface heat flux along the symmetry plane is shown from both HAP and experimental results in Fig. 18. In observing HAP results, we find significant asymmetry over the forebody surface; the maximum heat flux is located slightly off center, and a local maximum is displayed along the windward forebody shoulder region. Near-zero heat flux is found over much of the back side, with negative flux values in a small region of cold rapidly expanding flow around the downstream portion of the shoulder. Good qualitative agreement is found between HAP and experimental data points, although local differences of over 10% are shown along the forebody. Discrepancies are partly attributable to uncertainty in freestream conditions and gas-surface interaction modeling, among other factors.

## VII. Conclusions

A new Cartesian DSMC code, HAP, has been introduced. HAP is intended for fast setup and simulation of relatively small rarefied flow problems, and as a foundation for development and testing of new DSMC models and algorithms. The code is greatly simplified by avoiding complex routines for dynamic grid adaptation, and a series of alternate procedures are employed to compensate for the expected reduction in simulation accuracy when computational cells are not refined to the local mean free path. These procedures include the use of nonuniform transient adaptive subcells refined approximately to the local mean free path, and a new collision probability modification to correct for errors associated with density averaging over the cell volume. In addition to nonuniform subcells and modified collision probabilities, another unique feature of HAP is the ability to automatically generate two and three-dimensional shapes comprising a series of planar surface elements. This avoids a typical requirement to import externally defined boundary geometries for a wide variety of DSMC simulations, and considerably reduces setup time.

For initial code evaluation, a detailed comparison has been performed between results from HAP, other DSMC codes, and two sets of experiments for a rarefied hypersonic flow over a flat plate. Excellent agreement was found between results from the HAP simulation and those from the established DSMC code MONACO, with no observed differences outside expected levels of statistical scatter. Good qualitative agreement was also found with experimental density and surface flux data, and likely explanations were offered for observed discrepancies. As an additional test case, HAP was used to simulate a three-dimensional hypersonic flow over a spherically blunted cone. Both gas density and surface heat flux were compared to published experimental data, with generally encouraging results.

In another study involving the flat plate flow problem, several HAP simulations were performed using different grid refinement levels, both with and without the proposed collision probability modification. Spatial discretization errors associated with collision probability were quantified, and significant error reduction was demonstrated through the proposed modification. To the authors' knowledge, no previous investigation of these errors exist in the literature, and it is hoped that the present work encourages more detailed analysis of this problem. Previous work on spatial discretization errors in DSMC has focused on the effects of the mean collision separation (MCS), and modern DSMC collision selection procedures (e.g. transient adaptive subcells) tend to reduce or eliminate the correlation between MCS and cell size. Even with arbitrarily small MCS, however, spatial discretization errors may persist through a strong cell size dependence in collision probabilities. It should be emphasized that techniques for reducing the MCS, such as nearest neighbor collisions and adaptive subcells, do not reduce errors due to spatially averaged collision probabilities in the presence of a density gradient. Both the collision probability correction and the nonuniform subcell technique presented here are therefore suggested as ways to reduce the influence of cell size on simulation accuracy.

Planned areas of future work include implementation of procedures for dynamic time step adaption, parallelization with dynamic load balancing and utilization of shared memory for improved efficiency, and implementation of chemistry routines for simulation of high enthalpy reacting flows. Expected applications include aerothermal analysis of hypersonic technology demonstration vehicles and lifting reentry vehicles at high altitude.

### Appendix

A derivation is provided for Eq. (16), which gives an approximate expression for the hard sphere mean free path  $\langle \lambda \rangle_x$  averaged in the  $x$  coordinate direction over a line segment through the cell center  $(x_c, y_c, z_c)$  with  $x \in [x_c - 1/2\Delta x, x_c + 1/2\Delta x]$ . From Eq. (6), the number density along the line segment can be expressed as  $n(x) = n_c + (\partial n / \partial x)(x - x_c)$  where  $n_c$  is the cell-averaged number density and  $\partial n / \partial x$  is assumed constant. The local hard sphere mean free path is therefore

$$\lambda(x) = \frac{1}{\sqrt{2}\sigma \left( n_c + \frac{\partial n}{\partial x}(x - x_c) \right)} \quad (\text{A.1})$$

where  $\sigma$  is the hard sphere collision cross section. Integration of Eq. (A.1) over the interval  $[x_c - 1/2\Delta x, x_c + 1/2\Delta x]$  gives

$$\langle \lambda \rangle_x = \frac{1}{\Delta x} \int_{x_c - 1/2\Delta x}^{x_c + 1/2\Delta x} \lambda(x) dx = \frac{1}{\sqrt{2}\sigma n_c \Delta x} \ln \left( \frac{1 + \frac{\Delta x}{2n_c} \frac{\partial n}{\partial x}}{1 - \frac{\Delta x}{2n_c} \frac{\partial n}{\partial x}} \right) \quad (\text{A.2})$$

The right side of Eq. (A.2) is undefined when  $\partial n / \partial x = 0$ , and is potentially subject to large truncation error when the term in the parenthesis is close to 1, so an approximate expression for  $\langle \lambda \rangle_x$  is desired. The Taylor series approximation

$$\ln \left( \frac{1+q}{1-q} \right) \approx 2q + \frac{2}{3}q^3$$

for an arbitrary variable  $q \in (-1,1)$  is used to find

$$\langle \lambda \rangle_x \approx \frac{1}{\sqrt{2}\sigma n_c} \left( 1 + \frac{1}{12} \left( \frac{\Delta x}{n_c} \frac{\partial n}{\partial x} \right)^2 \right) \quad (\text{A.3})$$

Finally, we employ Eq. (8) to express  $\partial n / \partial x$  as a function of the  $x$  coordinate of the time-averaged particle center of mass in the cell,  $\bar{x}_{cm}$ , and substitute into Eq. (A.3).

$$\langle \lambda \rangle_x \approx \frac{1}{\sqrt{2}\sigma n_c} \left( 1 + 12 \left( \frac{\bar{x}_{cm} - x_c}{\Delta x} \right)^2 \right) \quad (\text{A.4})$$

### References

- <sup>1</sup>Bird, G. A., "Approach to Translational Equilibrium in a Rigid Sphere Gas," *Physics of Fluids*, Vol. 6, 1963, pp. 1518-1519.
- <sup>2</sup>Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, 1994.
- <sup>3</sup>Alder, B. J., and Wainwright, T. E., "Studies in Molecular Dynamics," *Journal of Chemical Physics*, Vol. 27, 1957, pp. 1208-1209.
- <sup>4</sup>Kolobov, V. I., Arslanbekov, R. R., Aristov, V. V., Frolova, A. A., and Zabelok, S. A., "Unified Solver for Rarefied and Continuum Flows with Adaptive Mesh and Algorithm Refinement," *Journal of Computational Physics*, Vol. 223, 2007, pp. 589-608.

<sup>5</sup>Burt, J. M., and Boyd, I. D., “Evaluation of a Particle Method for the Ellipsoidal Statistical Bhatnagar-Gross-Krook Equation,” AIAA paper 2006-989.

<sup>6</sup>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.

<sup>7</sup>LeBeau, G. J., “A Parallel Implementation of the Direct Simulation Monte Carlo Method,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 174, 1999, pp. 319-337.

<sup>8</sup>Bird, G. A., “The DS2V/3V Program Suite for DSMC Calculations,” *Rarefied Gas Dynamics, 24<sup>th</sup> International Symposium*, Vol. 762, 2005, pp. 541-546.

<sup>9</sup>Ivanov, M. S., Markelov, G. N., and Gimelshein, S. F., “Statistical Simulation of Reactive Rarefied Flows: Numerical Approach and Applications,” AIAA Paper 98-2669, 1998.

<sup>10</sup>Gao, D., Zhang, C., and Schwartzentruber, T. E., “A Three-Level Cartesian Geometry Based Implementation of the DSMC Method,” AIAA Paper 2010-450, 2010.

<sup>11</sup>Wu, J.-S., Tseng, K.-C., and Wu, F.-Y., “Parallel Three-Dimensional DSMC Method Using Mesh Refinement and Variable Time-step Scheme,” *Computer Physics Communications*, Vol. 162, 2004, pp. 166-187.

<sup>12</sup>Oran, E. S., Oh, C. K., and Cybyk, B. Z., “Direct Simulation Monte Carlo: Recent Advances and Applications,” *Annual Review of Fluid Mechanics*, Vol. 30, 1998, pp. 403-441.

<sup>13</sup>Borgnakke, C., and Larsen, P. S., “Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture,” *Journal of Computational Physics*, Vol. 18, 1975, pp. 405-420.

<sup>14</sup>Baganoff, D., and McDonald, J. D., “A Collision-Selection Rule for a Particle Simulation Method Suited to Vector Computers,” *Physics of Fluids A*, Vol. 2, No. 7, 1990, pp. 1248-1259.

<sup>15</sup>Macrossan, M. N., “Searching for a Near Neighbor Particle in DSMC Cells Using Pseudo-Subcells,” *Journal of Computational Physics*, Vol. 229, 2010, pp. 5857-5861.

<sup>16</sup>Gallis, M. A., and Torczynski, J. R., “Effect of Collision-Partner Selection Schemes on the Accuracy and Efficiency of the Direct Simulation Monte Carlo Method,” *International Journal for Numerical Methods in Fluids*, 2010 (published online; DOI: 10.1002/flid.2409).

<sup>17</sup>Bird, G. A., Gallis, M. A., Torczynski, J. R., and Rader, D. J., “Accuracy and Efficiency of the Sophisticated Direct Simulation Monte Carlo Algorithm for Simulating Non-continuum Gas Flows,” *Physics of Fluids*, Vol. 21, 2009, 017103.

<sup>18</sup>LeBeau, G. J., Boyles, K. A., and Lumpkin, F. E., “Virtual Sub-cells for the Direct Simulation Monte Carlo Method,” AIAA Paper 2003-1031, 2003.

<sup>19</sup>Sun, Q., and Boyd, I. D., “Evaluation of Macroscopic Properties in the Direct Simulation Monte Carlo Method,” *Journal of Thermophysics and Heat Transfer*, Vol. 19, No. 3, 2005, pp. 329-335.

<sup>20</sup>Lengrand, J. C., Allegre, J., Chpoun, A., and Raffin, M., “Rarefied Hypersonic Flow Over a Sharp Flat Plate: Numerical and Experimental Results,” *Rarefied Gas Dynamics, 18<sup>th</sup> International Symposium*, 1993, pp. 276-284.

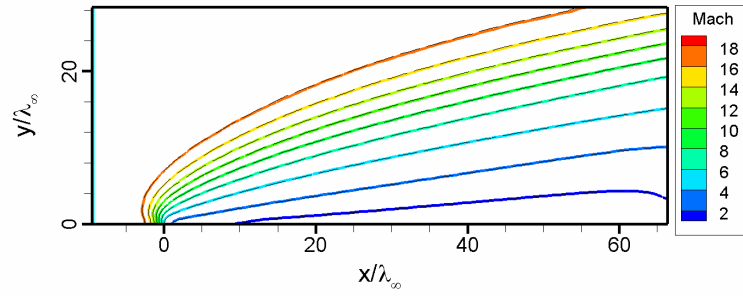
<sup>21</sup>Heffner, K. S., Gottesdiener, L., Chpoun, A., and Lengrand, J. C., “Leading Edge Effect on Rarefied Hypersonic Flow Over a Flat Plate,” AIAA Paper 91-1749.

<sup>22</sup>Allegre, J., Bisch, D., and Lengrand, J. C., “Experimental Rarefied Density Flowfields at Hypersonic Conditions Over 70-Degree Blunted Cone,” *Journal of Spacecraft and Rockets*, Vol. 34, No. 6, 1997, pp. 714-718.

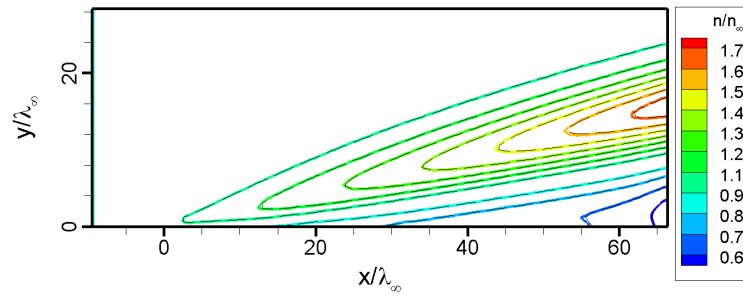
<sup>23</sup>Allegre, J., Bisch, D., and Lengrand, J. C., “Experimental Rarefied Heat Transfer at Hypersonic Conditions Over 70-Degree Blunted Cone,” *Journal of Spacecraft and Rockets*, Vol. 34, No. 6, 1997, pp. 724-728.

**Table 1. Parameters for HAP and MONACO simulations of hypersonic N<sub>2</sub> flow over a flat plate.**

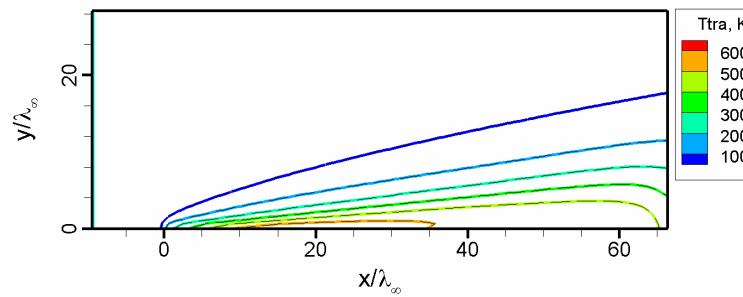
Freestream Mach number	20.2
Global Knudsen number	0.015
Freestream temperature	13.32 K
Plate surface temperature	290 K
Freestream number density	$3.716 \times 10^{20} \text{ m}^{-3}$
Freestream mean free path	1.688 mm
Cell length	1 mm
Number of cells	128×48
Time step interval	$3.102 \times 10^{-7} \text{ s}$
Numerical weight	$2.3225 \times 10^{12}$
Total number of time steps	20,000
Number of time steps before sampling	2000



**Figure 1. Contours of Mach number. HAP results are shown as colored lines, and MONACO results are shown as black dashed lines.**

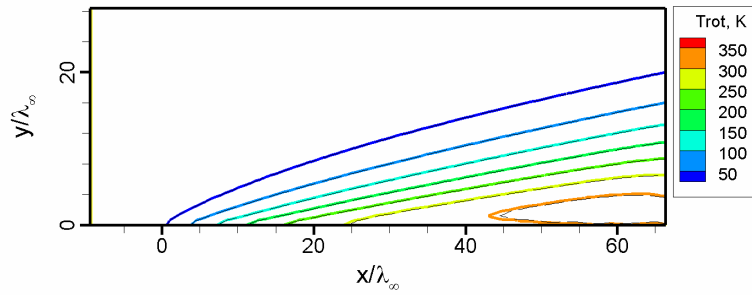


**Figure 2. Contours of normalized number density from HAP and MONACO simulations.**

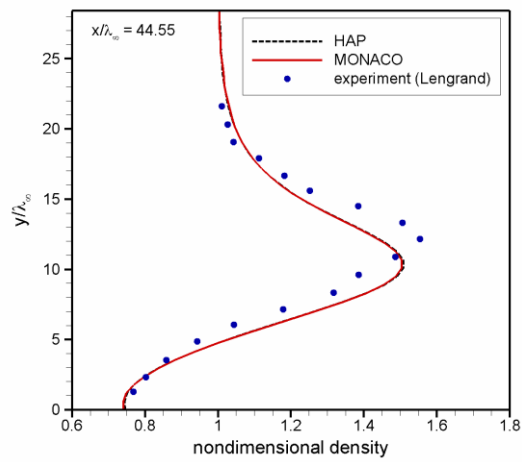


**Figure 3. Contours of translational temperature.**

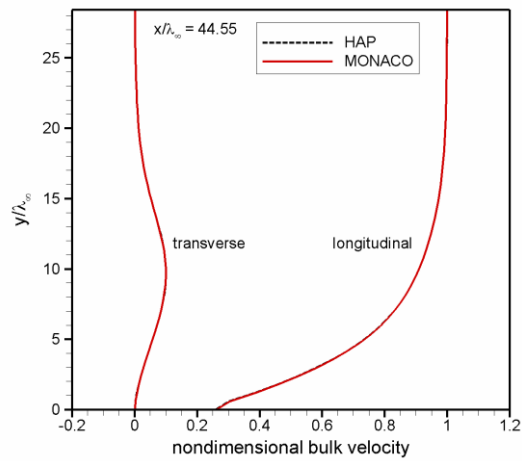




**Figure 4. Contours of rotational temperature.**



**Figure 5. Density profiles along a transverse plane.**



**Figure 6. Bulk velocity profiles along a transverse plane.**

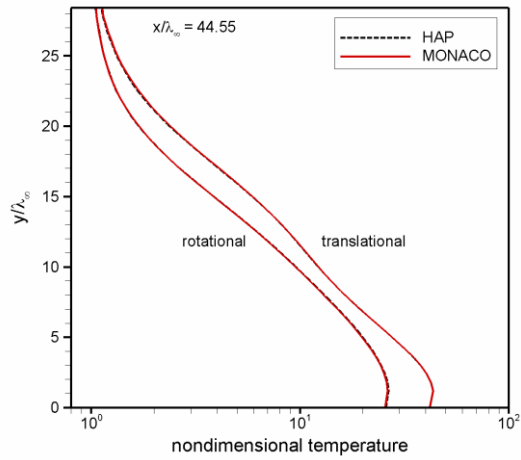


Figure 7. Profiles of translational and rotational temperature along a transverse plane.

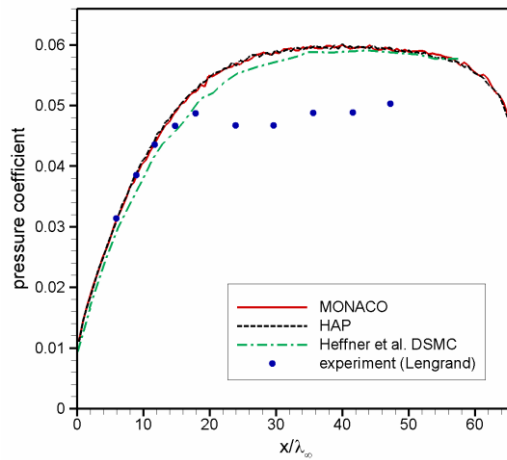


Figure 8. Pressure coefficient along the plate surface.

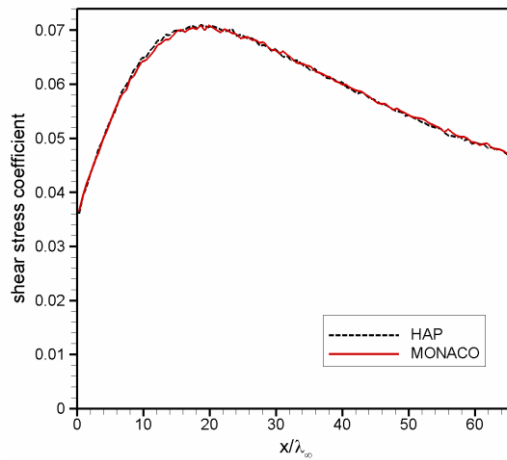


Figure 9. Shear stress coefficient along the plate surface.

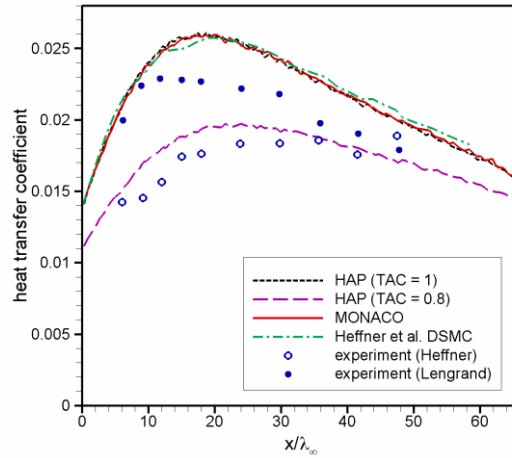


Figure 10. Heat transfer coefficient along the plate surface.

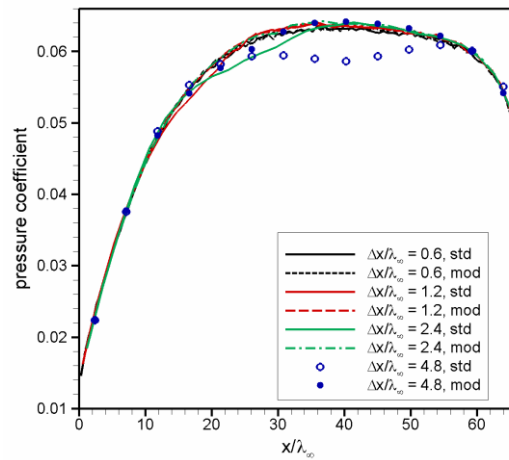


Figure 11. Surface pressure coefficient for various cell sizes, with and without gradient-based collision probability modifications.

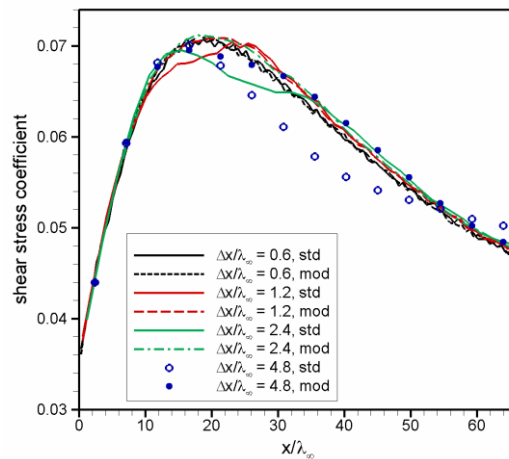


Figure 12. Surface shear stress coefficient for various cell sizes.

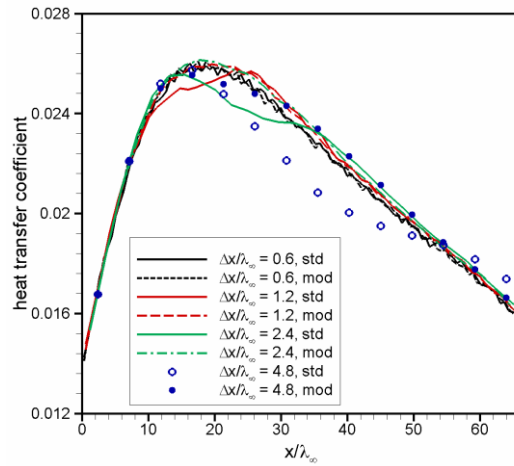


Figure 13. Surface heat transfer coefficient for various cell sizes.

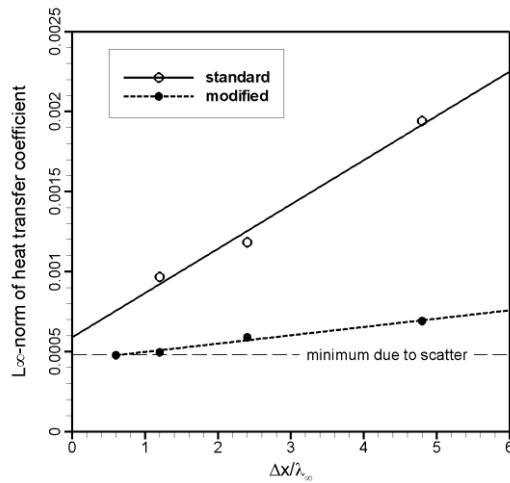


Figure 14.  $L_{\infty}$ -norm of heat transfer coefficient as a function of cell size, with and without gradient-based collision probability modifications.

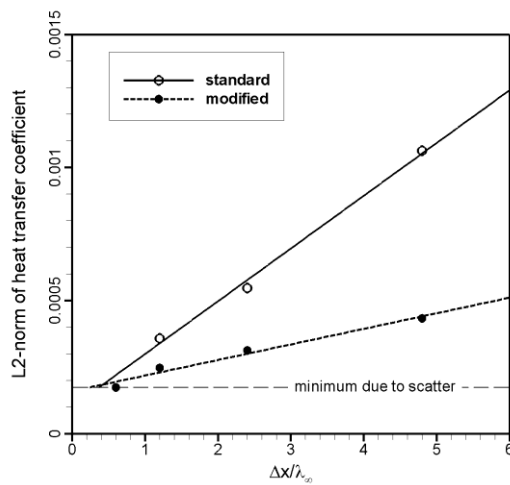


Figure 15.  $L_2$ -norm of heat transfer coefficient as a function of cell size.

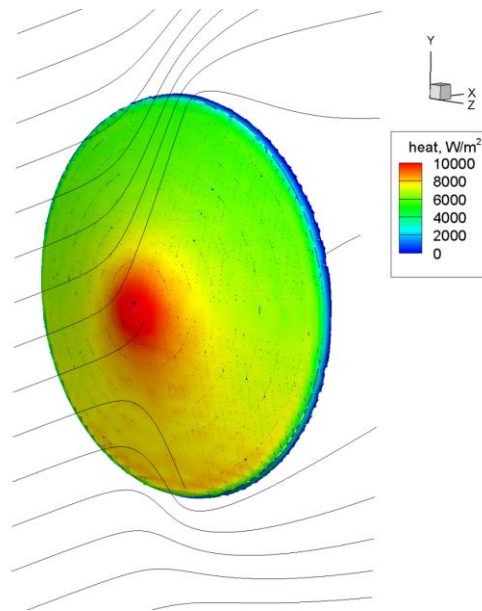


Figure 16. Contours of surface heat flux, and streamlines along symmetry plane, for hypersonic flow over 70° blunted cone at 10° angle of attack.

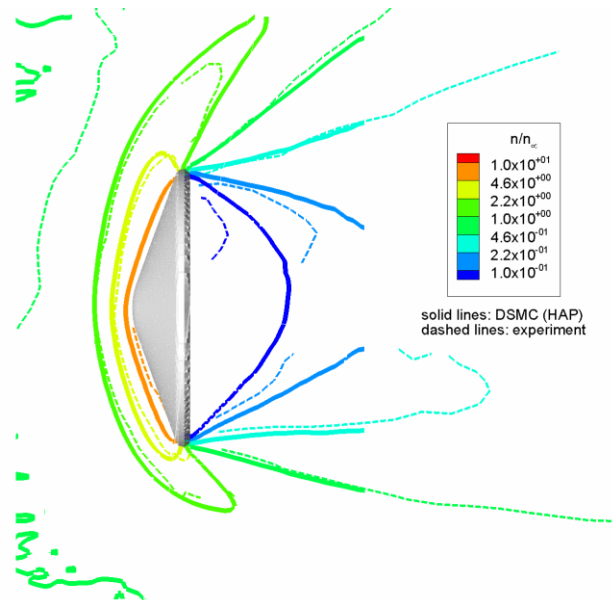
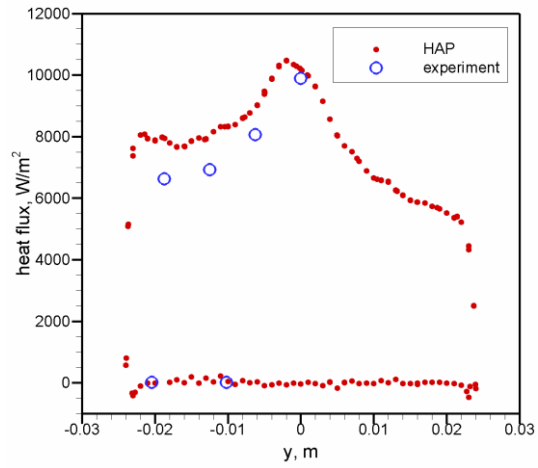


Figure 17. Contours of normalized density along symmetry plane. Dashed lines indicate experimental data.



**Figure 18. Variation in surface heat flux along symmetry plane.**