INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2013; **72**:937–966
Published online 4 February 2013 in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.3769

A hybrid DSMC/Navier–Stokes frame to solve mixed rarefied/nonrarefied hypersonic flows over nano-plate and micro-cylinder

Masoud Darbandi[1,*,†] and Ehsan Roohi[2]

[1]*Department of Aerospace Engineering, Center of Excellence in Aerospace Systems, Institute for Nanoscience and Nanotechnology, Sharif University of Technology, P.O. Box 11365-8639, Tehran, Iran*
[2]*Department of Mechanical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, P.O. Box 91775-1111, Mashhad, Iran*

SUMMARY

We extend a hybrid DSMC/Navier–Stokes (NS) approach to unify the DSMC and the NS simulators in one framework capable of solving the mixed non-equilibrium and near-equilibrium flow regions efficiently. Furthermore, we use a one-way state-based coupling (Dirichlet–Dirichlet boundary-condition coupling) to transfer the required information from the continuum region to the rarefied one. The current hybrid DSMC–NS frame is applied to the hypersonic flows over nanoflat plate and microcylinder cases. The achieved solutions are compared with the pure DSMC and NS solutions. The results show that the current hybrid approach predicts the surface heat transfer rate and shear stress magnitudes very accurately. Some important conclusions can be drawn from this study. For example, although the shock wave region would be a non-equilibrium region, it is not necessary to use a pure DSMC simulator to solve it entirely. This is important when the researchers wish to predict the surface properties such as velocity slip, temperature jump, wall heat flux rate, and friction drag magnitudes accurately. Our investigation showed that our hybrid solution time would be at least 40% (for the flat plate) and 35% (for the cylinder) of the time that must be spent by a pure DSMC solver to attain the same accuracy. Copyright © 2013 John Wiley & Sons, Ltd.

Received 1 April 2012; Revised 24 September 2012; Accepted 5 December 2012

KEY WORDS: DSMC; Navier–Stokes; hypersonic flow; microflow; nanoflow; hybrid algorithm; continuum breakdown; rarefaction; OpenFOAM

1. INTRODUCTION

Literature shows that the simulation of rarefied microgas/nanogas flow can remarkably help to improve the design of microelectromechanical systems/nanoelectromechanical systems [1]. Indeed, the microelectromechanical system/nanoelectromechanical system devices experience different rarefaction regimes. To describe the degree of gas flow rarefaction, the Knudsen number is defined as the ratio of mean free path of gas molecules, λ, to the one characteristic length in the flow field, L, that is, $\mathrm{Kn} = \lambda/\mathrm{L}$. The rarefaction regimes can be generally categorized as slip ($0.001 < \mathrm{Kn} < 0.1$), transition ($0.1 < \mathrm{Kn} < 10$), and free molecular ($\mathrm{Kn} > 10$) ones. There are different methods to simulate rarefied microflow/nanoflow regimes. The Navier–Stokes (NS) solutions are valid for the slip regime provided that the velocity slip and the temperature jump boundary conditions are suitably incorporated in solving rarefied/nonrarefied microflows/nanoflows. Alternatively, the DSMC [2] is a powerful numerical approach to solve the fluid dynamics problems in the entire rarefaction regimes.

*Correspondence to: Masoud Darbandi, Department of Aerospace Engineering, Center of Excellence in Aerospace Systems, Institute for Nanoscience and Nanotechnology, Sharif University of Technology, P.O. Box 11365-8639, Tehran, Iran.
†E-mail: darbandi@sharif.edu

However, the computational cost of DSMC cannot be justified as the flow approaches the continuum status. In continuum flow fields, the NS equations are a better choice to simulate the flow field efficiently. Therefore, the efficient simulation of gas flow fields with a large length scale variations needs advanced scheme such as a hybrid DSMC–NS frame. In a hybrid DSMC–NS frame, the stochastic particle solver is used to simulate the rarefied regions, and the continuum solver is used to simulate the near-equilibrium regions.

Different hybrid frames have been proposed to couple the continuum and stochastic particle solvers for the last decade. The flux-based coupling has been the most common coupling algorithm. In this algorithm, the fluxes of mass, momentum, and energy are first determined by the DSMC particles at the continuum–molecular interface. Then, they are imposed as the macroscopic flux into adjacent cells located in the continuum region. Similarly, the fluxes approximated by the NS solver are first calculated using the gradients of macroscopic flow variable distributions. Then, they are used to create the distribution of DSMC particles at the interface region. Wadsworth and Erwin [3] developed a flux-based hybrid DSMC–NS scheme and applied it to 1D shock problems. Hash and Hassan [4] used a flux-based coupling algorithm to simulate rarefied hypersonic gas flows. They concluded that the Marshak condition can be considered as the most accurate coupling algorithm. The Marshak condition calculates the properties at the interface using the conservation equations. Bourgat *et al.* [5] presented a coupled NS/Boltzmann approach to simulate the rarefied hypersonic flows. They proposed a kinetic model in the boundary layer coupled through wall friction forces to a global NS solver. Le Tallec and Mallinger [6] developed an adaptive coupling for a hybrid Boltzmann and NS equations and simulated the rarefied hypersonic flows. The coupling was achieved via matching half fluxes at the interface of the Boltzmann and NS domain. The two domains were automatically determined through computing the local kinetic residuals on a preliminary NS solution. Tiwari and Klar [7] presented a domain decomposition approach to couple the Boltzmann and Euler equations. They employed both adaptive time and space discretizations and created a direct coupling procedure. They reported that this strategy would lead to considerable gains in CPU time compared with the CPU time of a full Boltzmann equation. Garcia *et al.* [8] and Wijesinghe *et al.* [9, 10] used a hybrid DSMC/Euler solver with an adaptive mesh and algorithmic refinement scheme to solve the Euler equations. The adaptive mesh refinement method provided local grid refinement while the algorithm refinement feature allowed a transition to the DSMC solver where needed. The difficulty with the flux-based coupling is that the statistical scattering appearing in the DSMC solution usually affects the interfacial fluxes and eventually corrupts the NS solution. Therefore, the flux-based coupling techniques require one to choose a high number of DSMC samples to reduce their statistical scattering. Alternatively, Roveda *et al.* [11, 12] suggested a state-based coupling technique using a Dirichlet–Dirichlet boundary-condition coupling at the interface. A state-based coupling considers an average number of particles inside a DSMC cell adjacent to the interface to obtain the macroscopic values such as velocity, temperature, and density. These averaged magnitudes provide the required Dirichlet boundary-conditions for the NS solver at the interface. Similarly, the solutions of the NS equations for the state variables at the continuum side of the interface are used to generate particles required at the DSMC boundary cells. Aktas and Aluru [13] studied the flow field in microfilters using a hybrid DSMC/continuum solver incorporated with a state-based coupling algorithm. Wang and Boyd [14] and Sun *et al.* [15] used a hybrid information preservation scheme [16] and the NS equations to study rarefied hypersonic gas flows and low speed microflows. Degond *et al.* [17] presented a hybrid model capable of constructing a smooth transition from a rarefied domain to a continuum one. The main feature of their method was that the continuum and kinetic equations would be coupled in a small buffer zone in which the true solution was approximated by adding up the solutions of two set of equations. Unlike a typical domain decomposition method, the advantage of this coupling was that it did not require any boundary condition. Wu *et al.* [18] developed a parallel coupled DSMC–NS scheme using three-dimensional unstructured grids. They used a domain overlapping strategy with Dirichlet–Dirichlet boundary-condition coupling. Kolobov *et al.* [19] developed a unified flow solver to simulate rarefied and continuum gas flows. The developed unified flow solver was able to adapt the computational grid dynamically and to introduce and remove kinetic patches automatically. This may provide substantial saving because the molecular scale solution domains could be restricted only to the required regions.

DOI: 10.1002/fld

Schwartzentruber *et al.* [20, 21] used an explicit state-based DSMC/NS coupling scheme to study 2D hypersonic gas flows over blunt bodies such as cylinders. Abbate *et al.* [22] used the state-based coupling for efficient simulation of transient flows in 1D shock tube and 2D pressure-driven slit flows. Their simulations showed that the results of the state-based method would be independent of the size of the overlap region and the NS/DSMC interface position, as long as the NS equations were used in the limits of their applicability, that is, Kn < 0.1. John and Damodaran [23] used the hybrid NS–DSMC scheme with state-based coupling for simulation of the rarefied flow field in the hard disk interface microgap. Degond *et al.* [24] developed a multiscale kinetic–fluid solver to efficiently solve fluid dynamic problems with non-equilibrium localized regions. They proposed breakdown conditions of fluid using a new criterion based on the distribution function itself. Degond and Dimarco [25] presented a novel numerical algorithm to couple the DSMC method with a finite volume method for solving the Euler equations. To avoid inherent spurious oscillations in their DSMC solutions, they used a new moment-guided Monte Carlo technique [26], which permitted to reduce the variance of their particle method solution suitably.

In this work, we extend a hybrid DSMC–NS frame using a state-based coupling algorithm to simulate the rarefied microgas/nanogas flows. Comparing with the previous studies [18, 20], our hybrid solver is totally one way; that is, the NS equations are not solved in the near-equilibrium region during the hybrid procedure. This feature increases the efficiency and simplicity of hybrid simulations while the surface properties (such as the slip/jump, shear stress, and heat flux magnitudes) are estimated very accurately. Application of one-way DSMC–NS coupling had been reported for simulation of high-altitude plume flow [27–29]. However, our literature survey shows that the past one-way hybrid algorithms have mostly used a fixed interface location between the continuum and molecular solvers, whereas our hybrid algorithm benefits from an adaptive interface location; see Section 2.3.2. Additionally, this work is the first one that uses the one-way hybrid coupling to treat the rarefied hypersonic external flows. Therefore, the current paper can be considered as the first application of one-way adaptive surface hybrid frame to treat the external hypersonic flows.

Using our experience in developing different continuum [30–36] and stochastic particle [37–44] solvers, we extend our hybrid DSMC–NS frame in the context of OpenFOAM [45]. The NS solver implements the velocity slip and the temperature jump boundary conditions at the solid walls, where they are required.

2. COMPUTATIONAL APPROACH

2.1. Boltzmann equation

The Boltzmann equation, which describes the time evolution of velocity distribution function (VDF) of molecules, in the absence of external forces, is written as follows [2]:

$$\frac{\partial}{\partial t}(nf) + \boldsymbol{c}\frac{\partial}{\partial \boldsymbol{r}}(nf) = \int_{-\infty}^{+\infty}\int_{0}^{4\pi} n^2(f'\,f_1' - f\,f_1)\boldsymbol{c}_r\sigma\,\mathrm{d}\Omega\,\mathrm{d}\boldsymbol{c} \tag{1}$$

where n is the number density, $\boldsymbol{r}$ is the space vector, $\boldsymbol{c}$ is the velocity space vector, $\boldsymbol{c}_\mathrm{r}$ is the relative velocity between a molecule with velocity class $\boldsymbol{c}$ and one with velocity class $\boldsymbol{c}_1$, $\sigma\mathrm{d}\Omega$ is the differential cross section for the collision of a molecule of class $\boldsymbol{c}$ with another one having class $\boldsymbol{c}_1$ such that their postcollision velocities become $\boldsymbol{c}'$ and c_1', respectively, and functions f, f_1, f', and f_1' are the corresponding VDFs for one molecule and its collision partner before and after the collision. The first term in Equation (1) describes the time rate for the number of molecules. The second term describes the number of molecule changes due to the molecular movements. The term on the right-hand side is called the collision integral term. This term describes the change of molecular velocities after the molecular collisions.

2.2. Description of DSMC method and algorithm

DSMC is a computational tool to solve the Boltzmann equation on the basis of the direct statistical simulation of molecular processes described by the kinetic theory [2]. It is considered as a particle

DOI: 10.1002/fld

method in which each particle represents a large number of real gas molecules. The DSMC method applies a splitting approach to the particles' VDF to obtain a solution for the Boltzmann equation. If $S_{Q+D}^{\Delta t,h}$ presents the solution of Equation (1), the splitting approach relies on an approximation as follows [46]:

$$S_{D+Q}^{\Delta t,h} \approx S_D^{\Delta t} S_Q^{\Delta t,h} \tag{2}$$

where D stands for the free particle motion, Q describes the particle binary collisions, h stands for the cell size, and Δt is the time step. Equation (2) expresses the primary principle for the DSMC, that is, the decoupling of motion and collision of the simulated particles over the time step intervals, which are small if compared with the local mean collision time. The flowchart of DSMC method is observed in Figure 1. Generally, the DSMC simulation has three main steps of binary collision, free molecular movement, and sampling. They are explained as steps 1, 2, and 3 here [2,47].

Step 1: binary collision modeling

There are three major tasks that need to be performed in this step. First, it needs calculating the number of maximum particle pairs $N_{\text{collision}-\max}$ for a binary collision using

$$N_{\text{collision}-\max} = \frac{N(N-1)F_{\text{num}}(\sigma_{\text{T}}\boldsymbol{c}_{\text{r}})_{\max}\Delta t}{2V_c} \tag{3}$$

where N is the number of particles per cell, F_{num} is the weighting factor, that is, the number of real molecules that is represented by each particle, $(\sigma_{\text{T}}\boldsymbol{c}_{\text{r}})_{\max}$ is the maximum value of collision cross section multiplied by the particle relative velocity in the corresponding cell, and V_c is the cell volume.

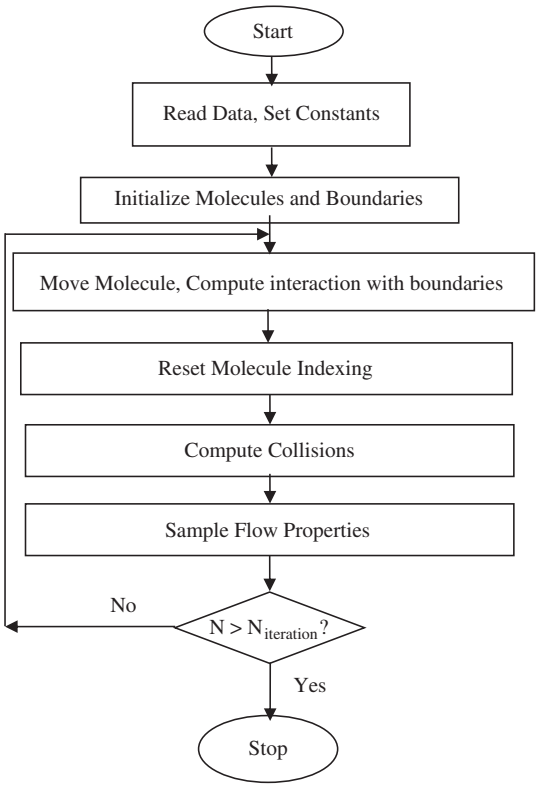


Figure 1. Flowchart of the DSMC algorithm.

DOI: 10.1002/fld

Second, benefiting from the acceptance–rejection technique, each pair of particles chosen randomly in the cell is checked for a collision probability using

$$\frac{(\sigma_{\mathrm{T}}\boldsymbol{c}_{\mathrm{r}})}{(\sigma_{\mathrm{T}}\boldsymbol{c}_{\mathrm{r}})_{\max}} > R_{\mathrm{f}} \tag{4}$$

where R_f is a uniformly distributed random number between 0 and 1.
Third, if the collision is accepted, the particle velocities are updated to their post-collision magnitudes. The conservations of momentum and energy require that the magnitude of the relative velocity of collision pairs (i, j) remains unchanged during the collision, that is, $\left(|\boldsymbol{c}'_{\mathrm{r}}| = |\boldsymbol{c}_r| = \left|\boldsymbol{c}_{\mathrm{i}} - \boldsymbol{c}_{\mathrm{j}}\right|\right)$. Next, the following procedure is used to calculate the direction of post-collision velocities. First, collision (ε) and deflection angles (χ) are calculated using two independent random numbers as follows:

$$\begin{aligned} \varepsilon &= 2\pi R_{\mathrm{f},1} \\ \cos\chi &= 2R_{\mathrm{f},2}^{1/\beta} - 1 \end{aligned} \tag{5}$$

Using a transformation from the Cartesian coordinates to the collision plane, the components of the post-collision relative velocity are calculated from

$$\begin{aligned} \mathrm{u}'_r &= \cos\chi\mathbf{u}_{\mathrm{r}} + \sin\chi\cos\varepsilon\left(\mathbf{v}_{\mathrm{r}}^2 + \mathbf{w}_{\mathrm{r}}^2\right)^{0.5} \\ \mathrm{v}'_{\mathrm{r}} &= \cos\chi\mathbf{v}_{\mathrm{r}} + \sin\chi(\mathbf{c}_{\mathrm{r}}\mathbf{w}_{\mathrm{r}}\cos\varepsilon - \mathbf{u}_r\mathbf{v}_{\mathrm{r}}\sin\varepsilon)/\left(\mathbf{v}_{\mathrm{r}}^2 + \mathbf{w}_{\mathrm{r}}^2\right)^{0.5} \\ \mathrm{w}'_r &= \cos\chi\mathbf{w}_{\mathrm{r}} - \sin\chi(\mathbf{c}_{\mathrm{r}}\mathbf{v}_{\mathrm{r}}\cos\varepsilon - \mathbf{u}_{\mathrm{r}}\mathbf{w}_{\mathrm{r}}\sin\varepsilon)/\left(\mathbf{v}_{\mathrm{r}}^2 + \mathbf{w}_{\mathrm{r}}^2\right)^{0.5} \end{aligned} \tag{6}$$

Accordingly, the velocity components of the colliding particles can be approximated using

$$\begin{aligned} \boldsymbol{c}'_{\mathrm{i}} &= \frac{1}{2\left[(\boldsymbol{c}_{\mathrm{i}} + \boldsymbol{c}_{\mathrm{j}}) + \boldsymbol{c}'_{\mathrm{r}}\right]} \\ \boldsymbol{c}'_{\mathrm{j}} &= \frac{1}{2\left[(\boldsymbol{c}_{\mathrm{i}} + \boldsymbol{c}_{\mathrm{j}}) + \boldsymbol{c}'_{\mathrm{r}}\right]} \end{aligned} \tag{7}$$

Step 2: free molecular movement

The distance traversed by the particles in each direction is equal to the chosen time step (Δt) multiplied by the corresponding velocity components (c_i). This yields

$$\mathrm{x}_{\mathrm{i}} = \mathrm{x}_{0,\mathrm{i}} + \mathrm{c}_{\mathrm{i}}\Delta t \tag{8}$$

Indexing of particles is performed after each particle movement. The boundary conditions should also be applied during the movement step. There are two major types of boundary conditions. The first type is the wall boundary condition. In this type, the velocity of a particle reflected from a diffusive wall is randomly attributed using the one half-range Maxwellian distribution, which is determined via the wall temperature magnitude as follows [2]:

$$\begin{aligned} u &= \sqrt{-\log(R_{\mathrm{f}})}V_{\mathrm{mpf}}\sin(2\pi R_{\mathrm{f}}) \\ v &= \sqrt{-\log(R_{\mathrm{f}})}V_{\mathrm{mpf}} \\ w &= \sqrt{-\log(R_{\mathrm{f}})}V_{\mathrm{mpf}}\cos(2\pi R_{\mathrm{f}}) \end{aligned} \tag{9}$$

where $V_{\mathrm{mpf}} = \sqrt{2RT}$ is the most probable speed of the particles at the wall temperature and R is the gas constant. In case of reflection from a symmetry boundary, the reflection is considered specular; that is, the normal velocity component is being reversed while the tangential component remains unchanged.

DOI: 10.1002/fld

The second type of boundary condition is the free stream boundary condition. In this type, the incoming flow velocity $\boldsymbol{U}_{\text{in}}$ and temperature T_{in} are normally given for the supersonic flow cases. Assuming a Maxwellian VDF for the incoming particles, Bird [2] derived the density flux (F) of incoming particles from the free stream boundary j using

$$F_j = \frac{n_j}{2\sqrt{\pi}V'_{\text{mpf},j}}\left\{\exp\left(-s_j^2\cos^2 v\right) + \sqrt{\pi}s_j\cos v[1+\text{erf}(s_j\cos v)]\right\} \tag{10}$$

where $s_j = \text{U}_j V'_{\text{mpf}}$, $V'_{\text{mpf}} = 1/\sqrt{2RT}$, v is an angle equal to either zero for the inlet boundary or 2π radians for the outlet boundary, and erf is the error function. The thermal velocity of particles entering into the domain can be calculated according to the acceptance–rejection technique. Considering this point, the resulting velocity components (c_x, c_y, c_z) for the particles, which enter into the domain from the inlet boundary, are obtained from

$$\begin{aligned} \text{c}_x &= \left(U_j + 3V'_{\text{mpf}}\right)R_{\text{f}} \\ \text{c}_y &= A\cos\phi \\ \text{c}_z &= A\sin\phi \end{aligned} \tag{11}$$

where $A = \sqrt{-\ln(R_{\text{f}})}V'_{\text{mpf}}$ and $\phi = 2\pi R_{\text{f}}$.

Step 3: sampling procedure

In this step, the macroscopic flow properties are sampled in each cell using the molecular velocities as soon as the flow reaches the steady-state condition. The sampling process is continued long after reaching the steady solution to suppress the inherent statistical scatters, which normally show up in the DSMC solutions. In such extent, the flow velocity and temperature magnitudes are obtained from [47]

$$N = \sum_1^{N_t} N_i \tag{12}$$

$$\boldsymbol{U}_i = \frac{1}{N}\sum_1^{N_t}\sum_1^{N_i}\boldsymbol{c}_i \tag{13}$$

$$T = \frac{m}{3k}\left(\frac{1}{N}\sum_1^{N_t}\sum_1^{N_i}\left|\boldsymbol{c}^2\right| - \left|\boldsymbol{U}^2\right|\right) \tag{14}$$

where N_i is the number of particles in one cell at the ith sampling time step and N is the total number of particles sampled during the total time steps of N_t in the corresponding cell.

2.2.1. The current DSMC solver. We have performed major contributions in developing a general DSMC solver named dsmcFoam [48] under the framework of OpenFOAM [45]. This solver was developed as a joint venture research work between Sharif University of Technology, University of Strathclyde, and OpenCFD, Ltd. OpenFOAM is a flexible set of efficient, object-oriented C++ modules for solving complex fluid flows [45]. It is mainly a finite-volume package designed to solve systems of differential equations for a wide range of fluid flow applications. Furthermore, dsmcFoam is a parallel DSMC solver that can model arbitrary geometries and arbitrary number of gas species on parallel computers. It uses the Variable Hard Sphere (VHS) collision model and the Larsen–Borgnakke internal energy redistribution model [2] to simulate the intermolecular collisions. In general, the features of dsmcFoam are quite comparable with other famous DSMC codes such as MONACO [14, 15] and Parallel Direct Simulation monte Carlo (PDSC) [18]. These three codes are able to simulate flow in arbitrary geometries using the 3D unstructured meshes. Moreover, dsmcFoam is one module of a general open source code with a hierarchical structure. The inheritance features of the dsmcFoam let any new user benefit from many features of OpenFOAM without a deep knowledge of coding.

DOI: 10.1002/fld

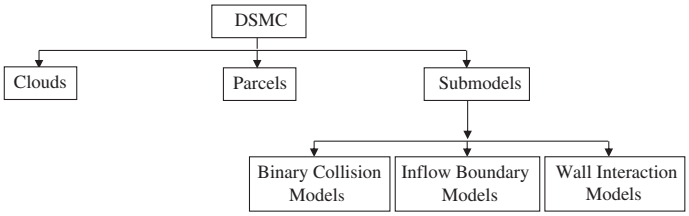


Figure 2. Directory of the dsmcFoam solver.

2.2.2. The dsmcFoam structure. In the dsmcFoam solver, the calculations are performed through using functions classified in three main categories of 'Parcels', 'Clouds', and 'Submodels', as shown in Figure 2. The category 'Parcels' includes the classes that trace the properties for one particle. However, the 'Clouds' category considers the evolution of a set of particles. For example, the 'Parcels' category contains functions that read and calculate the individual particle properties (such as velocity, position, internal energy, species type, and owner cell). Additionally, it traces the particle's movement. The interactions of particles with boundaries are also treated using the functions defined in this category. The 'Clouds' category, on the other hand, contains some other key functions, which are used in some important DSMC steps including the initialization of the simulation and the intermolecular collisions. The functions defined in the 'Clouds' category benefit from the physical models defined in the 'Submodels' category for binary collisions, wall-interaction collisions, and flux and velocity of the incoming particles, that is, inflow boundary conditions. These models were described previously.

More details on the dsmcFoam algorithms can be found in reference [48]. Briefly, the dsmcFoam sub-algorithms can be categorized in two major parts:

1. Particle tracking in unstructured, arbitrary, polyhedral meshes: Indeed, the particle tracking algorithm of dsmcFoam has been inherited from the algorithms originally extended for the molecular dynamics solver of the OpenFOAM. The full details of the tracking algorithm can be found in reference [49].
2. Particle distributing in arbitrary cell volumes for initial random configuration of the particles and uniformly allocating incoming particles from the free-stream boundaries: In practice, to distribute particles in the cells volume and free-stream faces, each cell/face is first divided into a number of tetrahedral/triangle cells. Selecting a random number, particles are positioned in the corresponding tetrahedral/triangle, whose volume/area fraction is larger than the random number.

2.3. Description of the NS equations and the utilized numerical scheme

The compressible NS equations can be derived from the Chapman–Enskog (CE) expansion of the Boltzmann equation. These equations, namely the conservations of mass, momentum, and the total energy, are written as follows [50]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \tag{15}$$

$$\frac{\partial (\rho \boldsymbol{V})}{\partial t} + \nabla \cdot (\boldsymbol{V}(\rho \boldsymbol{V})) + \nabla p + \nabla \cdot \boldsymbol{\tau} = 0 \tag{16}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\boldsymbol{V}(\rho E)) + \nabla \cdot (\boldsymbol{V} p) + \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{V}) + \nabla \cdot \boldsymbol{q} = 0 \tag{17}$$

where ρ is the fluid density, p is the pressure, $E = e + |\boldsymbol{V}^2| / 2$ is the total energy, e is the internal energy per unit mass, $\boldsymbol{q}$ is the conductive heat transfer, and $\boldsymbol{\tau}$ is the shear stress tensor, which is related to the velocity field via the Stokes law as follows:

DOI: 10.1002/fld

$$\boldsymbol{\tau} = -2\mu \ \mathrm{dev}(\mathbf{D}) \tag{18}$$

$$\mathbf{D} = 0.5[\nabla \boldsymbol{V} + (\nabla \boldsymbol{V})^{\mathrm{T}}] \tag{19}$$

where μ is the dynamic viscosity and dev denotes the deviatoric of a tensor, that is, $\mathbf{D} - (1/3)\,\mathrm{tr}(\mathbf{D})\boldsymbol{I}$, where $\boldsymbol{I}$ is the unit tensor and tr is the trace of a tensor. In the present study, we consider a calorically perfect gas for all simulations. So, $p = \rho RT$ and $e = c_{\mathrm{v}}T$, where c_{v} is the specific heat at constant volume and R is the gas constant.

2.3.1. Discretization of the NS equations. In this work, we use the 'rhoCentralFoam' module of OpenFOAM as our NS solver [51]. RhoCentralFoam is an explicit, collocated finite-volume, density-based solver to simulate viscous compressible flows. In the finite-volume method, the governing equations are integrated over all cell volumes. The divergence and gradient terms are then replaced with integrals over the cell surface using the Gauss's theorem. These integrations require the flux evaluation over the cell faces, which are subsequently approximated using suitable interpolations of the neighboring cell center values. In 'rhoCentralFoam', the required interpolations are performed using the second-order semi-discrete, non-staggered schemes of Kurganov and Tadmor (KT) [52] and Kurganov, Noelle, and Petrova (KNP) [53]. Here, we briefly describe the discretization of a general tensor ($\boldsymbol{\Psi}$) field by interpolating the values of cell centers to approximate the values at the faces of the same cell [51, 54].

The 'convective terms' in the NS equations, $(\nabla \cdot [\rho \boldsymbol{V}], \nabla \cdot [\boldsymbol{V}(\rho \boldsymbol{V})], \nabla \cdot [\boldsymbol{V}(\rho E)] + \nabla \cdot [\boldsymbol{V} p])$, are integrated, interpolated, and linearized over a control volume. They eventually yield

$$\int\int\int_{\vartheta} \nabla \cdot [\boldsymbol{V}\boldsymbol{\Psi}] \mathrm{d}\vartheta = \int\int_{\mathrm{S}} \mathrm{d}\boldsymbol{S} \cdot [\boldsymbol{V}\boldsymbol{\Psi}] \approx \sum\nolimits_{\mathrm{f}} \boldsymbol{S}_{\mathrm{f}} \cdot [\boldsymbol{V}_{\mathrm{f}}\boldsymbol{\Psi}_{\mathrm{f}}] = \sum_{\mathrm{f}} \phi_{\mathrm{f}}\boldsymbol{\Psi}_{\mathrm{f}} \tag{20}$$

where Σ_f denotes a summation over the entire cell faces and $\phi_f = \boldsymbol{S}_f \cdot \boldsymbol{V}_f$ is the volumetric flux. For compressible flows, the fluid properties are transported by both the flow motion and the wave propagation. This point necessitates the fact that the flux interpolation needs to be established on the basis of a full convective–diffusive transportation and wave propagation behaviors. They hence take into account the impact of all directions in the target interpolations. This should be achieved via employing the KT and KNP methods. In this regards, the interpolation technique is separated into directions corresponding to flow out–to and into the face of owner cell. The directions are denoted as $f+$ for $+\boldsymbol{S}_f$ direction and $f-$ for $-\boldsymbol{S}_f$ direction, where $\boldsymbol{S}_f$ is the face area vector. Considering the above descriptions, Equation (20) reduces to the following [54]:

$$\sum_f \phi_f \boldsymbol{\Psi}_f = \sum_f [\alpha \phi_{f+} \boldsymbol{\Psi}_{f+} + (1-\alpha)\phi_{f-}\boldsymbol{\Psi}_{f-} + w_f(\boldsymbol{\Psi}_{f-} - \boldsymbol{\Psi}_{f+})] \tag{21}$$

The first and second terms in the right-hand side of Equation (21) indicate the flux evaluations at the $f+$ and $f-$ directions, respectively. These fluxes, which are associated with the local speeds of propagation, can be calculated from

$$\boldsymbol{\Psi}_{f+} = \max(c_{f+}|\boldsymbol{S}_f| + \phi_{f+}, c_{f-}|\boldsymbol{S}_f| + \phi_{f-}, 0) \tag{22}$$

$$\boldsymbol{\Psi}_{f-} = \max(c_{f+}|\boldsymbol{S}_f| - \phi_{f+}, c_{f-}|\boldsymbol{S}_f| - \phi_{f-}, 0) \tag{23}$$

Here, $c_{f\pm} = \sqrt{\gamma RT_{f\pm}}$ are the speeds of sound at the cell face. In the KT method, the $f+$ and $f-$ contribute equally so that the weighting coefficient becomes $\alpha = 0.5$. Consequently, the KT method is classified as a 'central' scheme. However, the KNP method calculates α on the basis of a one-sided local speed of wave propagation. Evidently, the weighting should be biased in the upwind direction. Therefore, this scheme is called 'central-upwind'. Considering the above descriptions, the weighting factor can be generally defined as

$$\alpha = \begin{cases} 0.5 & \text{KT method} \\ \dfrac{\boldsymbol{\Psi}_{f+}}{(\boldsymbol{\Psi}_{f+} + \boldsymbol{\Psi}_{f-})} & \text{KNP method} \end{cases} \tag{24}$$

DOI: 10.1002/fld

The third term in the right-hand side of Equation (21) is only required in cases where the convection term is a part of a substantive derivative, that is $\nabla \cdot [V(\rho V)]$ in Equation (16), for which $\partial(\rho V)/\partial t$ completes the substantive derivative of ρV. This diffusive volumetric flux (w_f) is calculated from

$$w_f = \begin{cases} \alpha \ \max(\Psi_{f+}, \Psi_{f-}) & \text{KT} \quad \text{method} \\ \alpha(1-\alpha)(\Psi_{f+} + \Psi_{f-}) & \text{KNP} \quad \text{method} \end{cases} \tag{25}$$

The current method derives the Ψ_{f+} and Ψ_{f-} face interpolated magnitudes for different variables such as temperature and density from the values of adjacent cell center values. The interpolation procedure uses a limiter to switch between the low and high order schemes on the basis of a flux limiter function $\beta(r)$, where $r \geqslant 0$ represents the ratio of successive gradients of the interpolated variables. On a polyhedral mesh and for the $f+$ direction, r is given by

$$r = 2\frac{d \cdot (\Psi\nabla)_{\mathrm{p}}}{(\nabla_{\mathrm{d}}\Psi)_{\mathrm{f}}} - 1 \tag{26}$$

where $(\Psi\nabla)_{\mathrm{p}}$ is the full gradient of Ψ at the owner cell P with a linear interpolation and $(\nabla_{\mathrm{d}}\Psi)_{\mathrm{f}} = \Psi_{\mathrm{N}} - \Psi_{\mathrm{P}}$ is the gradient component normal to the face, scaled by the length $|\boldsymbol{d}|$, where $\boldsymbol{d}$ is a distance vector that connects the centroid of the owner cell P to that of the neighboring cell N. The Ψ_{f+} and Ψ_{f-} are interpolated using the primitive first-order upwind and second-order linear interpolation schemes. In fact, the chosen limiters are those of the total variation diminishing scheme, in which $\beta(r)/r = \beta(1/r)$, for example Minmod $\beta(r) = \max[0, \min(1, r)]$ and van Leer $\beta(r) = (r+|r|)/(1+r)$. Considering the above descriptions, the magnitude of Ψ in the $f+$ direction is obtained from

$$\Psi_{f+} = (1 - g_{f+})\Psi_{\mathrm{p}} + g_{f+}\Psi_{\mathrm{N}} \tag{27}$$

where $g_{f+} = \beta(1 - w_f)$. It is seen that $\beta = 0$ results in a pure upwind interpolation, $\beta = 1$ in a linear interpolation, and $\beta = 2$ in a downwind interpolation.

Back to Equations (16) and (17), the 'gradient terms' in the NS equations are integrated and converted to the following [54]:

$$\int\int\int_v \nabla\Psi \mathrm{d}v = \int\int_{\boldsymbol{S}} \Psi \mathrm{d}\boldsymbol{S} \approx \sum_f \boldsymbol{S}_f \Psi_f \tag{28}$$

However, the KT and KNP methods split the interpolation magnitude considering the $f+$ and $f-$ directions as follows:

$$\sum_f \boldsymbol{S}_f \Psi_f = \sum_f [\alpha \boldsymbol{S}_f \Psi_{f+} + (1-\alpha)\boldsymbol{S}_f \Psi_{f-}] \tag{29}$$

where the Ψ_{f+} and Ψ_{f-} interpolations use the preceding described limiters.

Back to Equations (16) and (17), the Laplacian terms can be generalized using a general diffusion coefficient definition of Γ. These terms are initially discretized to

$$\int\int\int_v \nabla \cdot (\Gamma\nabla\Psi)\mathrm{d}v = \int\int_{\boldsymbol{S}} \mathrm{d}\boldsymbol{S} \cdot (\Gamma\nabla\Psi) \approx \sum_f \Gamma_f \boldsymbol{S}_f \cdot (\nabla\Psi)_f \tag{30}$$

where Γ_f is interpolated linearly using its neighboring cell center values. It is because we assume that the diffusion is a function of cell temperature. Considering a general case in which the face is non-orthogonal, that is $\boldsymbol{S}_f$ is not parallel to $\boldsymbol{d}$, the evaluation of $\boldsymbol{S}_f \cdot (\nabla\Psi)_f$ is divided into two parts including an orthogonal part (which considers the neighbor and owner cell values) and a non-orthogonal part, which takes into account a full gradient (which is calculated at the cell centers and interpolated at the cell faces) as follows:

$$\boldsymbol{S}_f \cdot (\nabla\Psi)_f = A(\Psi_{\mathrm{N}} - \Psi_{\mathrm{P}}) + \boldsymbol{a} \cdot (\nabla\Psi)_f \tag{31}$$

where $A = \frac{|\boldsymbol{S}_f|^2}{\boldsymbol{S}_f \cdot \boldsymbol{d}}$ and $\boldsymbol{a} = \boldsymbol{S}_f - A\boldsymbol{d}$.

DOI: 10.1002/fld

2.3.2. Solution algorithm of the NS equation. 'RhoCenteralFoam' is an explicit solver; that is, all new solutions at the current time step are calculated explicitly from the known magnitudes of previous time step. Because the explicit approaches are faced with a time step limit, a sequential operator splitting approach is employed to consider the diffusive terms as suitable implicit corrections of an approximate inviscid solution. There are three main steps in the solution algorithm [51]:

Step 1. Solve the inviscid equations, which are given by

$$\left[\frac{\partial(\rho \boldsymbol{V}')}{\partial t}\right]_{\text{inv}} + \boldsymbol{\nabla}\cdot(\boldsymbol{V}(\rho \boldsymbol{V}')) + \boldsymbol{\nabla}\text{p} = 0 \tag{32}$$

$$\left[\frac{\partial(\rho E')}{\partial t}\right]_{\text{inv}} + \boldsymbol{\nabla}\cdot(\boldsymbol{V}(\rho E' + \text{p})) + \boldsymbol{\nabla}\cdot(\boldsymbol{\tau}\cdot\boldsymbol{V}) = 0 \tag{33}$$

where the time derivative terms represent the time rate of only inviscid fluxes. Note that the prime denotes the quantities calculated on the basis of the inviscid assumption.

Step 2. Update the primitive variables, for example, $\boldsymbol{V}$ and T, from the solution of Equations (32) and (33). For the temperature, it is updated from $T = \left(\frac{\rho(\rho E')}{\rho} - \frac{|\boldsymbol{V}'|^2}{2}\right)/C_v$.

Step 3. Solve a diffusion correction equation implicitly for an array of primitive variables. In this case, the time derivative represents the part due to diffusion only, that is, $(\partial/\partial t)_{\text{vis}}$. Then, the velocity and temperature fields are updated via solving suitable diffusion correction equations as follows:

$$\left[\frac{\partial(\rho \boldsymbol{V})}{\partial t}\right]_{\text{vis}} - \boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\boldsymbol{V}) - \boldsymbol{\nabla}\cdot(\boldsymbol{\tau}_{\text{exp}}) = 0 \tag{34}$$

$$\left[\frac{\partial(\rho c_v T)}{\partial t}\right]_{\text{vis}} - (\kappa\boldsymbol{\nabla}T) = 0 \tag{35}$$

where κ is the heat conduction coefficient. In all above equations, the stress tensor is treated explicitly.

2.3.3. Wall boundary conditions. The NS equations are typically valid in the slip regime if we suitably apply the velocity slip and temperature jump boundary conditions [30, 35–37]. To extent the applicability of the NS solver in a hybrid algorithm, we employ the first-order Maxwell velocity slip and Smoluchowski temperature jump boundary conditions at the walls as follows [54, 55]:

$$\mathbf{V}_{\text{g}} - \mathbf{V}_{\text{w}} = -\frac{2-\sigma_{\text{u}}}{\sigma_{\text{u}}}\lambda\boldsymbol{\nabla}_{\text{n}}(\mathbf{S}\cdot\mathbf{V}) - \frac{2-\sigma_{\text{u}}}{\sigma_{\text{u}}}\frac{\lambda}{\mu}\mathbf{S}\cdot(\mathbf{n}\cdot\boldsymbol{\Pi}_{\text{mc}}) - \frac{3}{4}\frac{\mu}{\rho}\frac{\mathbf{S}\cdot\boldsymbol{\nabla}T}{T} \tag{36}$$

$$T_{\text{g}} - T_{\text{w}} = -\frac{2-\sigma_{\text{T}}}{\sigma_{\text{T}}}\frac{2\gamma}{(\gamma+1)\,\text{Pr}}\lambda\boldsymbol{\nabla}_{\text{n}}T \tag{37}$$

where $\mathbf{n}$ represents the unit normal vector to the surface and μ is the dynamic viscosity. Additionally, the subscripts g and w refer to gas adjacent to wall and wall, respectively. σ_u and σ_T are the tangential momentum and thermal accommodation coefficients, respectively. γ is the specific heat ratio and Pr is the Prandtl number. The tensor $\mathbf{S} = \mathbf{I} - \mathbf{nn}$, considering $\mathbf{I}$ as the identity tensor, removes the normal components of any nonscalar field. Π_{mc} is expressed as

$$\boldsymbol{\Pi}_{\text{mc}} = \mu(\nabla\mathbf{V})^{\text{T}} - (2/3)\mathbf{I}_{\text{tr}}(\nabla\mathbf{V}) \tag{38}$$

where the subscript tr denotes the trace. To be consistent with DSMC simulations presented here, we consider both of the accommodation coefficients equal to one. Moreover, we use the power-law relation to present gas viscosity as a function of temperature magnitudes as follows [2]:

DOI: 10.1002/fld

$$\mu = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right)^{\omega} \tag{39}$$

$$\mu_{\text{ref}} = \frac{15\sqrt{m\pi k T_{\text{ref}}}}{2\pi D_{\text{ref}}^2 (5-2\omega)(7-2\omega)} \tag{40}$$

where ω is the macroscopic viscosity temperature exponent, D is the molecular diameter and k is the Boltzmann constant. We apply the power-law relation to be consistent with the VHS model used in our DSMC solver.

2.4. The hybrid strategy

2.4.1. The breakdown parameter. Any extended hybrid algorithm should address the locations of continuum-stochastic interface properly using an appropriate continuum breakdown parameter. A hybrid solver can enhance its computational efficiency by restricting the DSMC solutions to only non-equilibrium regions suitably. A continuum breakdown parameter can be also used to locate the interface between continuum and non-equilibrium regions. This parameter can be derived from relations between the Boltzmann and NS equations. Different breakdown parameters were suggested by previous researchers. For example, Tiwari [56] proposed a criterion on the basis of Grad's 13 moments expansion to perform an automatic domain decomposition and that to couple the Boltzmann and Euler equations. Levermore *et al.* [57] derived nondimensional, non-negative definite matrices from the moments of a molecular distribution. The breakdown criteria were obtained on the basis of the deviations of the eigenvalues of this matrix from their equilibrium values. A key breakdown parameter is the local Knudsen number based on the gradient's local length (GLL) of the flow properties. It is defined as

$$\text{Kn}_{\text{GLL}\phi} = \frac{\lambda}{\phi} |\nabla \phi| \tag{41}$$

where ϕ is an arbitrary flow parameter and $\text{Kn}_{\text{GLL}} = \max(\text{Kn}_{\text{GLL},\rho}, \text{Kn}_{\text{GLL},|V|}, \text{Kn}_{\text{GLL},T}, 5 \times (T_{\text{tr}} - T_{\text{rot}})/T_{\text{rot}})$ [58]. $\text{Kn}_{\text{GLL},|V|}$ is computed on the basis of the velocity magnitude, that is, $|V| = \sqrt{u^2 + v^2 + w^2}$. It should be noted that the normalization of velocity gradient in $\text{Kn}_{GLL,V}$ is carried out using the local speed of sound a in low-speed regions of the flow. The last term in the Kn_{GLL} considers the non-equilibrium condition between translation and rotational degrees of freedom. The local mean free path used in the gradient-length Knudsen number is calculated using the VHS formula as follows [2]:

$$\lambda = \frac{2(5-2\omega)(7-2\omega)}{15} \sqrt{\frac{m}{2\pi k T}} \left(\frac{\mu}{\rho} \right) \tag{42}$$

Another interesting breakdown parameter is the B parameter, which is taken as the maximum value of the normalized shear stress and heat flux magnitudes [59]. It is defined as

$$B = \max \left(|\boldsymbol{q}_i^*|, |\boldsymbol{\tau}_{ij}^*| \right) \tag{43}$$

where the normalized heat flux and shear stress magnitudes are calculated from

$$\boldsymbol{q}_i^* = -\frac{\kappa}{p} \left(\frac{2m}{\kappa T} \right)^{0.5} \nabla T \tag{44}$$

$$\boldsymbol{\tau}_{ij}^* = \frac{\mu}{P} \left(\boldsymbol{V}_{i,j} + \boldsymbol{V}_{j,i} - \frac{2}{3} \boldsymbol{V}_{k,k} \boldsymbol{\delta}_{ij} \right) \tag{45}$$

where P is the pressure, $\boldsymbol{V}$ is the velocity vector. Indices i, j, and k stand for the three velocity components. The continuum breakdown occurs whenever $\text{Kn}_{\text{GLL}} > 0.05$ [58] or if $B > 0.1$ [59]. In this work, we use Kn_{GLL} as the breakdown parameter to determine the continuum–molecular interface.

DOI: 10.1002/fld

Although $\mathrm{Kn}_{\mathrm{GLL}}$ has been widely used to determine the continuum breakdown [14, 15, 20, 21], there are some inherent limitations and approximations in this parameter. In particular, this parameter does not successfully capture the continuum breakdown within the boundary layer [60]. As stated by Wang and Boyd [60], $\mathrm{Kn}_{\mathrm{GLL}}$ successfully captures the breakdown at the wall but not in most of the boundary layer region. To simulate boundary layers with a hybrid frame efficiently, Lian *et al.* [61] recently suggested a new continuum breakdown parameter on the basis of the pressure magnitude. According to this new parameter, the thermal equilibrium generally remains well in most of the boundary layer region. They reported improved hybrid cycle efficiencies using their new breakdown parameter.

In order to calculate the gradient in Equation (41), it is assumed that the flow parameters (ϕ) vary linearly within each cell in both x and y directions. As a result, this gradient can be calculated in any type of unstructured grids using

$$\phi - \phi_0 \equiv \Delta\phi = \Delta x \phi_{\mathrm{x}} + \Delta y \phi_{\mathrm{y}} + \Delta z \phi_{\mathrm{z}} \equiv \Delta \boldsymbol{X} \cdot \nabla \phi \tag{46}$$

where ϕ_0 is the parameter magnitude at the cell center or node. Considering the nodes surrounding the centre node, a matrix equation can be subsequently constructed as follows:

$$\begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \Delta x_2 & \Delta y_2 & \Delta z_2 \\ \Delta x_3 & \Delta y_3 & \Delta z_3 \\ \Delta x_4 & \Delta y_4 & \Delta z_4 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} = \begin{Bmatrix} \Delta\varphi_1 \\ \Delta\varphi_2 \\ \Delta\varphi_3 \\ \Delta\varphi_4 \\ \vdots \end{Bmatrix} \tag{47}$$

The number of coefficients ($\Delta x_i, \Delta y_i, \Delta z_i$) depends on the number of neighboring nodes surrounding the center node. This equation is generally overspecified. To find the gradient vector, the matrix can be inverted using the least-squares approach. The resulting gradients are then substituted in Equation (41).

2.4.2. The hybrid cycle. We consider a small overlap region at the interface between the two solvers. Evidently, both solvers are expected to result in valid solutions in the overlap region. The benefit of using the overlap region is that the hybrid solution is not influenced by the exact position of the interface. Our hybrid cycle is a one-way, state-based frame and is a major modification of the schemes reported by Wu *et al.* [18] and Schwartzentruber *et al.* [20, 21]. In fact, our hybrid algorithm is the first application of one-way hybrid coupling to solve the external hypersonic flows. The flowchart of the proposed hybrid algorithm is shown in Figure 3. It consists of the following steps:

1. Obtain the NS solution for the entire domain. This solution is inaccurate in non-equilibrium regions but it is acceptable in the near-equilibrium regions.
2. Apply the breakdown parameter to the NS solution and determine the non-equilibrium (DSMC) regions.
3. Extend an overlap region in the NS domain in a manner to specify a small region of the NS domain as the overlap (non-equilibrium) region. The overlap region is a location where $\mathrm{Kn}_{\mathrm{GLL}} - \Delta\mathrm{Kn} < \mathrm{Kn} < \mathrm{Kn}_{\mathrm{GLL}}$. The criterion $\Delta\mathrm{Kn}$ is specified by the user.
4. Use the DSMC solver to obtain solution in the rarefied and overlap regions. Boundary conditions at the overlap region are supplied from the NS solution. The details are provided shortly.
5. Check the convergence criterion using

$$\mathbf{AAE}\varphi = \sum_{i=1}^{i=N_{\mathrm{cells}}} \left| \frac{(\varphi_{\mathrm{Hybrid}} - \varphi_{\mathrm{Exact}})}{\varphi_{\mathrm{Hybrid}}} \right|_{\mathrm{Surface\ or\ overlap\ outer\ boundary}} < \varepsilon \tag{48}$$

This criterion is defined on the basis of the accumulated absolute error (AAE) of any property (ϕ) on the outer boundary of the overlap region or on the surface. The parameter ϕ can be

DOI: 10.1002/fld

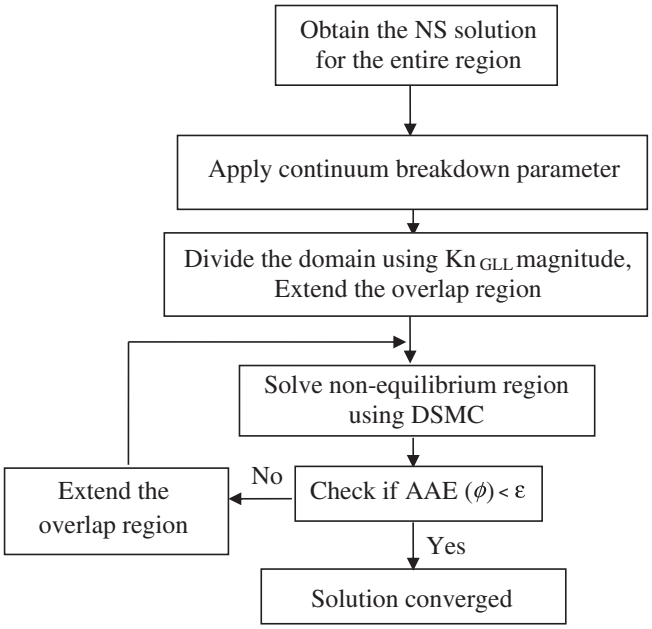


Figure 3. Flowchart of the hybrid frame.

any parameter of interest. The accumulation is performed over all the overlap/surficial cells. Of course, to check the convergence, we need the exact solution on the surface or interface. The exact solution can be obtained from the experimental measurement. It can also be a full DSMC or Boltzmann solution (typically available from the literature). If the convergence criterion is satisfied, the hybrid cycle is finished.

6. If AAE $>$ ε, extend the overlap region inside the NS domain. This could be carried out by decreasing ΔKn or alternatively by any other suitable strategy.
7. Go to step 4, use the previously created particles in the DSMC region at the preceding iteration and create particles in the new DSMC cells applying the Maxwellian VDF.
8. The hybrid iteration continues until the convergence parameter (AAE) indicates that the interface location does not need any more changes at the current stage.

One additional row (or column) of cells in the NS domain adjacent to the overlap region is considered as the particle buffer region. Particles are created in the overlap region according to the thermal and hydrodynamic properties, (T, V, ρ), and their gradients in the buffer region employing the Maxwellian velocity distribution. In other words, the NS boundary conditions are transferred into the DSMC solver through determination of the number of particles and their properties such as the most probable speed of particles in the buffer zone. Even though the NS solution is consistent with the CE VDF [20, 21], the generated CE distribution is not necessarily the correct velocity distribution for a DSMC solver, as the DSMC and NS solutions do not necessarily need to agree well on the whole interface [62]. This point will be later shown; see Figure 13. Consequently, we generate DSMC particles at the buffer zone according to a less-expensive Maxwellian distribution. Of course, because the number of injected particles through the interface, whose 'approximated distribution functions' differ from their real distributions, is small compared with the total number of injected particles through the interface, this simplification does not degrade the accuracy of our proposed hybrid frame solution.

Unlike the hybrid algorithm suggested by Schwartzentruber *et al.* [21], we do not solve the NS equations in the near-equilibrium region during our hybrid procedure. In other words, our hybrid frame is one way; that is, only the NS boundary conditions are transferred into the DSMC region. The benefit of this procedure is to simplify the implementation of a hybrid algorithm. We will show later that the main objective of our hybrid frame is to predict the surface properties such as

DOI: 10.1002/fld

the velocity slip, temperature jump, heat flux, and shear stress magnitudes accurately. In a hybrid simulation of a hypersonic external flow, the calculation of these parameters is performed by the DSMC solver because the flow over the surface is usually taken as non-equilibrium, which is consistent with the DSMC solution. Consequently, we do not focus on the solution of the flow field far from the main object. This would justify the use of a one-way hybrid algorithm in the current work.

The size of overlap region plays a critical role in the current algorithm. If the first estimate of ΔKn is small, more iterations should be performed during the procedure. However, a larger approximation for ΔKn would result in more computational time for the DSMC solver in our hybrid frame. Our experience has shown that the computational costs would decrease considerably if we choose a slightly over-expanded size for the interface region between the computational zones of the two solvers. It is because this strategy results in less number of visits to steps 4 to 6. Our first estimate is taken as $\Delta \mathrm{Kn} = 0.01$ for all simulations, which is described in Section 3.

Indeed, this hybrid algorithm is designed for solving steady-state problems. In our hybrid code, the data switch should be performed manually. This means that the user should specify new boundary conditions for the DSMC solver at step 3 above. Our experience showed that the number of iterations would not exceed 2 or 3 for the test cases solved in Section 3.

It should be noted that the suggested frame could be extended to 3D applications without needing drastic changes. In 3D applications, the hybrid interface will be a face rather than a line. However, the determinations of interface location and the boundary transfer process are performed similar to those performed in 2D case. However, a dynamic implementation for a hybrid algorithm is an important issue, and it highly depends on the original DSMC and NS codes structures. A dynamic process requires the addition of some functions to the code to extract data from the NS solver at the target interfaces. These dynamic functions should also split the original NS domain into the near-equilibrium and rarefied regions and allocate the rarefied region to the DSMC solver. These functions should provide the required boundary conditions for the DSMC solver from the NS solution. They should also update the interface location during the solution procedure.

At this stage, there might be a question how one may efficiently check the convergence criterion when one deals with a DSMC fluctuating solution. Indeed, Degond *et al.* [24] have reported some remedies for suppressing the negative fluctuation effects over the interface for the flux-based hybrid solutions. Because our hybrid algorithm is a state-based one, there is no need to evaluate the fluxes at the interface. As is known, the fluxes are more prone for fluctuations than the macroscopic properties. Additionally, employing a one-way hybrid algorithm, we do not need to transfer any possible DSMC fluctuating solutions into the NS region because that region is not updated during the hybrid iterations and it does not require any boundary condition taken from the DSMC solution. On the other hand, any fluctuations in the macroscopic quantities may affect the convergence criterion magnitude. These fluctuations would occur when the bulk velocity of the flow is much smaller than the most probable thermal velocity, that is, in low speed subsonic flows. In such cases, we need to employ a flux-corrected filter to post-process the noisy DSMC solutions [63], as is first suggested by Reference [64]. Because the present work focuses on high speed flows, there is no need to employ any post-process filtering because the interfacial solution is almost smooth.

It should be noted that the numbers of interfacial cells for the DSMC and NS solvers are not normally the same. In practice, the number of NS solver interfacial data is much less than that of the DSMC cells at the interface. To remedy this problem, the NS solution at the interface and their cellular information are stored in an array. Then, these data are suitably interpolated to drive the magnitudes at the exact locations of the DSMC interfacial cells. We benefit from such storage and data manipulation in our hybrid solver.

3. RESULTS AND DISCUSSION

In this section, we present the results of our extended hybrid DSMC–NS solver. We have chosen two test cases to illustrate the capabilities of our hybrid solver. They are the flow over a nanoscale flat plate and a microscale cylinder. For validation purposes, we used the Bird DSMC2D code [2] to derive results for the flow over the nanoplate test case. On the other hand, the results of cylinder

DOI: 10.1002/fld

flow can be found in the past studies [65, 66]. Therefore, we can readily evaluate the accuracy of our hybrid solver.

3.1. Test 1: nanoscale flat plate

We first examine the hypersonic flow over a nanoscale flat plate. Figure 4 shows the geometry of plate and the imposed boundary conditions. The gas is monatomic argon. The plate length is 100 nm, and the height of the computational domain is 60 nm. The inlet Mach and Knudsen numbers are 4.38 and 0.00129, respectively. The Knudsen number is defined according to the plate length. The freestream gas is at a temperature of 300 K and a pressure of 4.14×10^7 Pa. The plate surface temperature is 500 K. For the DSMC solver, we apply the freestream conditions at the inlet, outlet, and upper boundaries. For the NS solver, we specify the velocity, temperature, and pressure magnitudes at the inlet and upper boundaries. However, it is not necessary to specify any boundary conditions at the supersonic outlet. We specify a short length (equal to 10% of the plate length) to upstream of the plate leading edge as the symmetric (or specular reflector) boundary condition for the DSMC solver in order to provide a more realistic inlet velocity profile at the beginning of plate. This idea has been repeatedly used by the current authors in different events [30, 35, 36]. The plate itself is considered one with a diffusive surface. The NS time step is 2×10^{-12} s. The Courant number for the NS solver is set as 0.55.

Before applying our hybrid algorithm, it is instructive to evaluate the accuracy of our individual DSMC and NS solvers. Figure 5 shows the grid independency test for the NS solver. This figure shows the velocity slip and the surface pressure distributions over the flat plate surface employing five different structured gird resolutions of 300×60, 300×180, 300×360, 300×720, and 300×900 cells. Of course, it is not necessary to refine grid in the x-direction because the variation of flow properties in the longitudinal direction is small [67, 68]. As is seen in this figure, the solutions are very close for the last two grids. Consequently, we use a grid with 300×720 cells to present our current results.

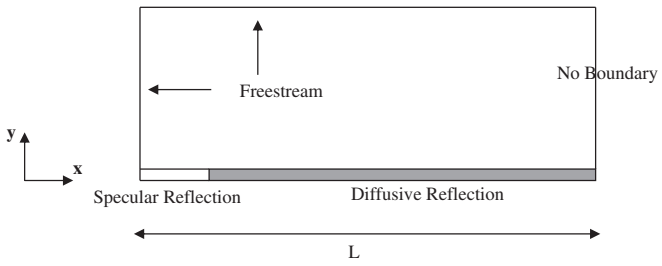


Figure 4. Flat plate geometry and the chosen boundary conditions for the DSMC solver, test 1.

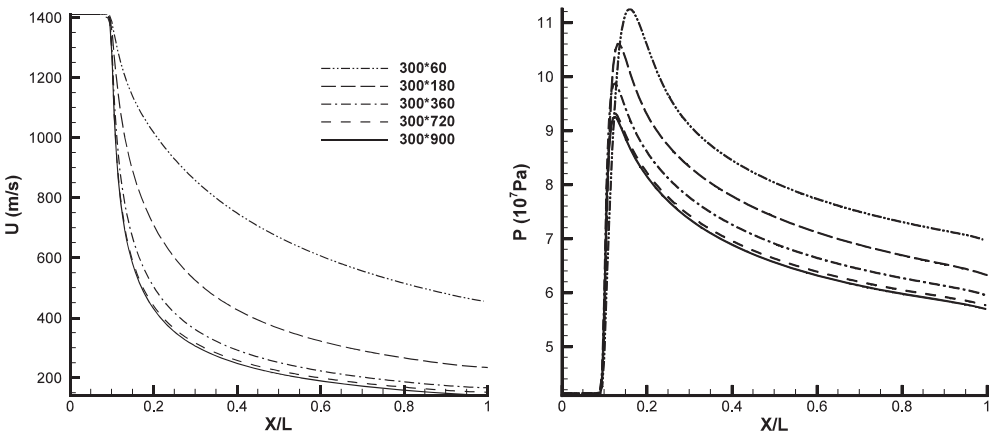


Figure 5. Grid independency test for the NS solution, test 1.

DOI: 10.1002/fld

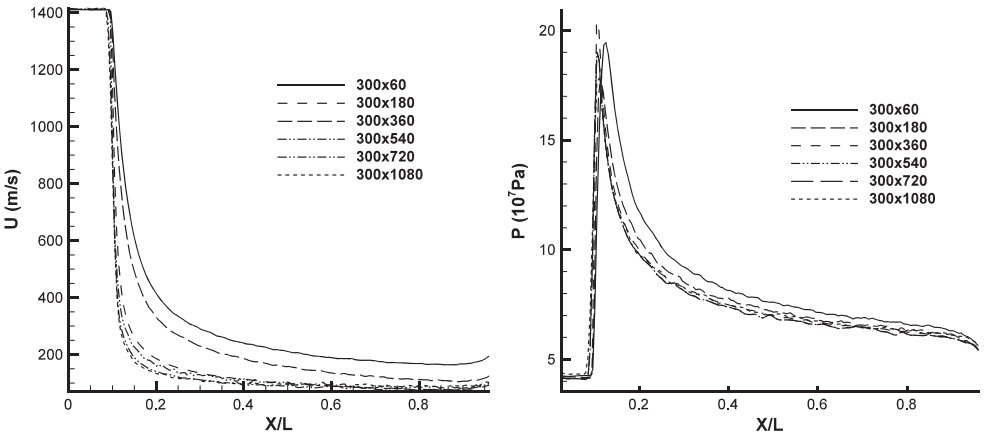


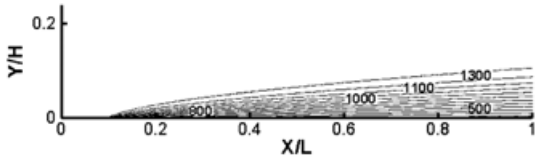
Figure 6. Grid independency test for the DSMC solution, test 1.

A similar grid refinement study is performed for the DSMC solution; see Figure 6. We use six structured grid resolutions of 300×60, 300×180, 300×360, 300×540, 300×720 and 300×1080 cells. We put at least 20 particles in each cell. As is observed in Figure 6, the two last solutions for the slip velocity magnitudes are very close to each other; however, the surface pressure distributions show much less sensitivity to the grid size. Generally, our study showed that the derived magnitudes would not be affected that much for grids finer than 300×360. Considering this achievement, we used a grid size with 300×720 cells to perform our DSMC study. Additionally, considering the use of two subcells in each direction, the subcell sizes would be $\lambda/2$ in both x- and y-directions. The DSMC time step is chosen to be a fraction of the mean collision time. As a result of this restriction, the particles do not cross more than one cell during one time step. This ensures the decoupling between particles' motions and collisions. In this test case, the time step is chosen as 4×10^{-13} s, which is about 25% of the mean collision time. Considering this time step as our relative movement time and assuming a particle with a speed of V_{mpf}, we obtain

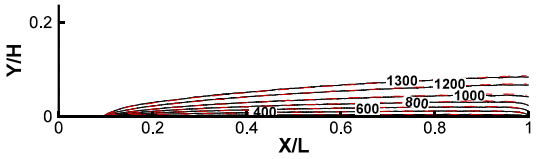
$$\Delta x_i \approx 0.25 \frac{\lambda}{V_{\text{mpf}}} \times V_{\text{mpf}} = 0.25\lambda < \Delta x_{\text{cell}} \tag{49}$$

So, it is concluded that this particle would move around 0.25λ, which is smaller than the size of cell and subcell. Figure 7 presents the velocity and temperature contours for the NS and DSMC solvers, separately. To validate the current solution, we compare the dsmcFoam solution with the solutions obtained from the DSMC2D code, developed by Bird [2]. It is observed that both of our DSMC velocity and temperature solutions are in excellent agreement with those of the DSMC2D code. Even though the flow global Kn number is in early slip flow regime, the DSMC and NS solutions perform differently in the shock wave and boundary layer regions. In other words, the characteristics of this hypersonic flow cause a strong non-equilibrium behavior in these regions, which is beyond the valid range of current slip/jump boundary conditions applied in the rhoCentralFoam solver. Other similar investigations in the rarefied flat plate flow also report that the Maxwell velocity slip and Smoluchowski temperature jump boundary conditions are ineffective for such high speed hypersonic flows [54, 69]. Can we use other extended boundary conditions to work better with hypersonic flow test cases? Indeed, there have been alternative slip/jump boundary conditions to solve the hypersonic flow cases; however, the results indicate that the achieved accuracy is not considerably superior compared with that of the Maxwell/Smoluchowski boundary conditions, for example, see the derivations and results reported in Reference [69]. Because of a higher complexity in implementing such slip/jump boundary conditions, we did not take them in our study. It is because the main focus of this work is to develop a new hybrid algorithm in the frame of OpenFOAM; therefore, we normally apply the available boundary condition options in the rhoCenteralFoam code.

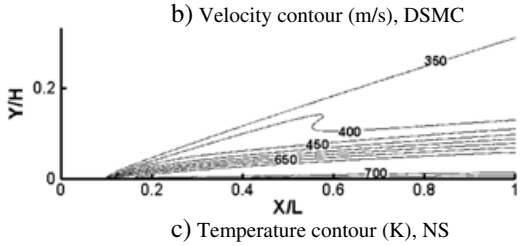
DOI: 10.1002/fld



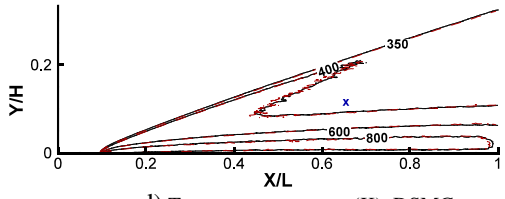
a) Velocity contour (m/s), NS



b) Velocity contour (m/s), DSMC



c) Temperature contour (K), NS



d) Temperature contour (K), DSMC

Figure 7. Velocity and temperature contours over the flat plate using the NS solver and two DSMC solvers, namely dsmcFoam (solid line) and DSMC2D code [2] (dashed line, red), test 1.

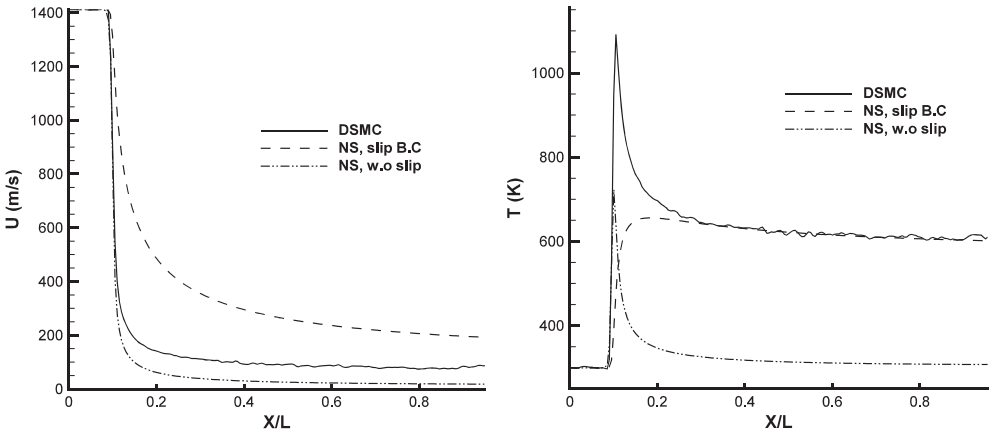


Figure 8. Comparison of gas velocity and temperature values at the flat plate surface using the DSMC and NS (with and without the first order slip boundary conditions) solutions.

Figure 8 compares the velocity slip and temperature jump at the flat plate surface derived from the NS (without and with the first-order slip/jump boundary conditions) and the DSMC solutions. We extracted the slip and jump properties from the DSMC solver according to a macroscopic sampling of properties near the surface rather than a direct sampling from the molecular velocities. This figure shows that there are considerable velocity slip and temperature/pressure jumps at the flat plate

DOI: 10.1002/fld

surface. They are mostly due to the leading edge shock wave and the flow rarefaction along the plate surface. Both solvers indicate that the shock starts at the same location at the tip of plate; however, the DSMC solver predicts a stronger shock there. The NS solver subject to slip/jump boundary conditions underpredicts the velocity slip and temperature jump highly at the beginning of plate leading edge. At the end of the plate, the NS solution predicts a velocity slip about 50% higher than that of DSMC. However, the NS temperature jump is quite close to the DSMC solution after $X/L = 0.5$. From this location, both of gas velocity and temperature magnitudes are approximately constant at the surface. The NS solver, without slip/jump boundary conditions, performs a rapid decrease in temperature as soon as the shock wave region ends up; that is, the flow temperature decreases to the wall temperature magnitude after the shock wave.

To investigate the problem more precisely, we are encouraged to focus on a local gradient-based Knudsen number, that is, Equation (41), to determine the non-equilibrium regions. Figure 9 depicts the $\mathrm{Kn}_{\mathrm{GLL,max}}$ contours calculated by the NS and DSMC solvers. As is seen, both solvers show that $\mathrm{Kn}_{\mathrm{GLL,max}}$ is well above the threshold value of 0.05 in the whole boundary layer region and a portion of shock wave adjacent to the wall. Therefore, the continuum solution cannot be counted on in these regions. Interestingly, $\mathrm{Kn}_{\mathrm{GLL,max}}$ contour shows that the shock wave is in near-equilibrium condition far from the plate. However, as we show later, the NS flow property solutions in the shock wave region differ from the DSMC ones, and that it cannot provide a suitable boundary conditions for the hybrid frame unless the shock wave reaches the end of plate, where it has become sufficiently weak. According to our hybrid algorithm, we apply $\mathrm{Kn}_{\mathrm{GLL,max}}$ criterion to the NS solution and simulate the regions located under the horizontal line, which include the $\mathrm{Kn}_{\mathrm{GLL,max}} > 0.05$ region, using the DSMC solver. Figure 10 schematically shows the domain attributed to the DSMC solver in our hybrid treatment. The first interface location (excluding the overlap region) was $Y/H_{\mathrm{interface}} = 0.05$. The interface location changes if the condition of $\mathrm{AAE}(\phi)_{\mathrm{interface}} < \varepsilon$ is

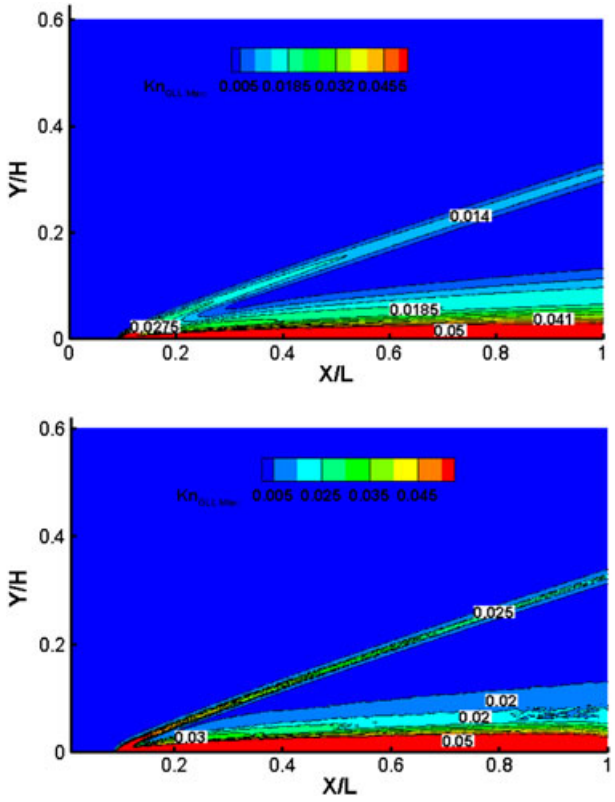


Figure 9. Comparing the magnitude of $\mathrm{Kn}_{\mathrm{GLL,max}}$ calculated by the current NS solver subject to slip/jump boundary conditions (top) and DSMC solver (bottom) individually, test 1.

DOI: 10.1002/fld

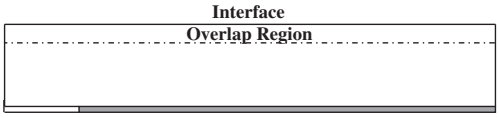


Figure 10. Solution domain for the hybrid solver, test 1.

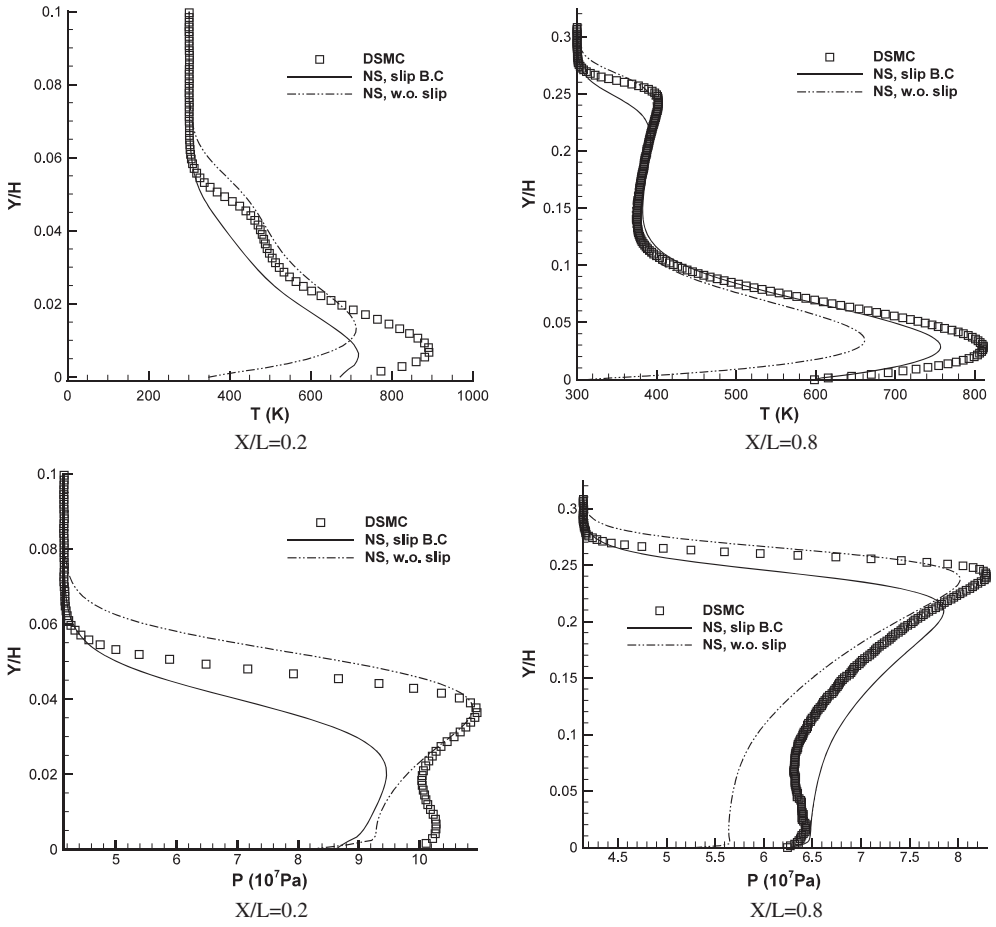


Figure 11. Temperature and pressure profiles at different axial sections on the flat plate using the DSMC and NS (with and without applying the slip boundary conditions) solvers.

not satisfied at the end of our hybrid solution iteration. Because we are interested in calculating hydrodynamics and thermal properties over the flat plate, we use $\phi = (U, P, T)$ in Equation (48). Consequently, $\mathrm{AAE}(\phi) = \max(\mathrm{AAE}(U), \mathrm{AAE}(P), \mathrm{AAE}(T))$. The value of ε is set as 1×10^{-3}. As was discussed in Section 2.3.2, the interface location is changed two times in the current test; that is, it is changed to $Y/H_{\text{interface}} = 0.14$ and 0.18 via changing $\Delta\mathrm{Kn}$.

Before investigating the boundary conditions at the interface, it is instructive to discuss the usefulness of employing slip/jump boundary conditions in the NS solver within a hybrid frame. Figure 11 shows the temperature and pressure profiles obtained from the NS and DSMC solvers at two cross sections of $X/L = 0.2$ and 0.8. Specifically, this figure elaborates the effects of slip/jump boundary conditions on the solution accuracy in the longitudinal direction. Close to our first overlap location, $Y/H_{\text{overlap}} = 0.06$, and at both $X/L = 0.2$ and 0.8 sections, the use of both slip/jump boundary conditions has resulted in better agreements with the DSMC solutions. Close to the $Y/H_{\text{interface}} = 0.19$ interface location, the temperature magnitudes for the slip and no-slip considerations are close to each other. However, the no-slip pressure distribution is in better agreement with the DSMC pressure

DOI: 10.1002/fld

distribution. Our experience showed that the latest interface would be located somewhere relatively far from the wall boundary for both test cases investigated here. As is seen in Figure 11, there is not a considerable benefit to employ the slip/jump boundary conditions in regions far from the wall because this can increase the computational time of the NS solver unnecessarily. A longer solution time would be normal because the calculations of the gradient terms appeared in the slip/jump equations, for example, Equations (36) and (37), enforce additional computation time to the preceding time. Having a fine grid at the wall, the computational cost may not suggest we use other slip/jump boundary conditions, which can improve the solution accuracy slightly. Of course, basically, this conclusion is not applicable to internal flow cases, where the employment of slip/jump boundary conditions can usually increase the ranges of NS equation's applicability.

Figure 12(a–c) compares the pressure distributions derived from the DSMC and NS (subject to the slip/jump boundary conditions) solvers at three hybrid overlap boundaries, that is, $Y/L = 0.06$, 0.15, and 0.19. At the first glance, the NS solution predicts a lower pressure distribution than that of the DSMC over the shock wave zone. However, the continuum solution overpredicts the DSMC solution after the shock. For the two other overlap sections, there is almost a constant difference between the NS and DSMC pressure distributions at the shock wave and afterward regions. However, the regions of error decrease as the shock approaches the end of plate; see Figure 12(c). Figure 12(d) shows the gas pressure distribution on the plate surface obtained from the DSMC and hybrid solvers for three consecutive iterations. The first hybrid iteration results in an erroneous

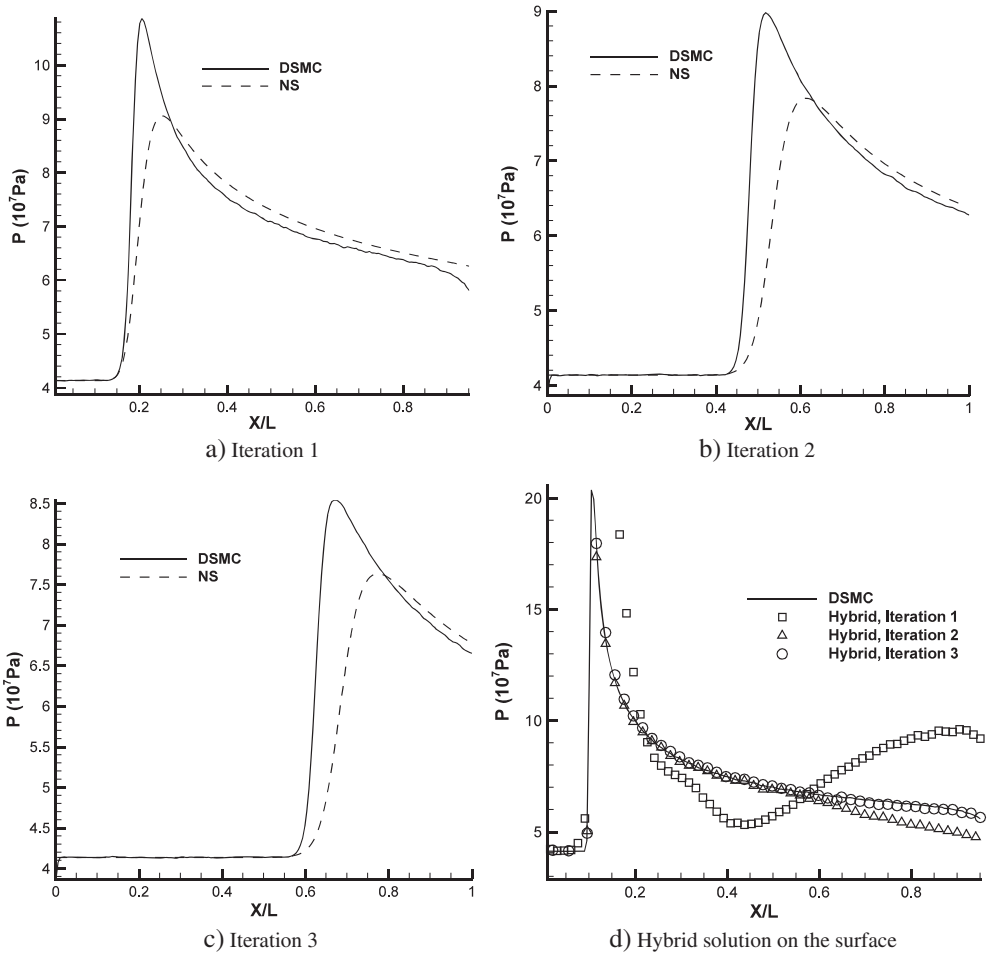


Figure 12. (a)–(c) Pressure distributions at the chosen hybrid overlap boundaries derived from the DSMC and NS solvers and (d) gas pressure distribution over the surface in different hybrid iterations.

DOI: 10.1002/fld

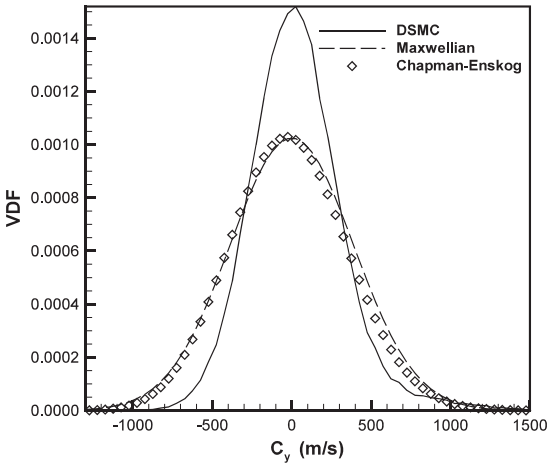


Figure 13. Comparison of the velocity distribution function from the DSMC simulation with that of the Maxwellian and Chapman–Enskog distribution; the cell location is $(X/L, Y/H) = (0.65, 0.18)$; see symbol x in Figure 7(d).

solution for the entire plate including the shock region. It is due to a large error influence originated from the NS solution into the DSMC boundary cells; see Figure 12(a). The error is transferred back into the particle region through the boundary cells and stays in the hybrid solution. The interface is moved further up into the NS region via decreasing ΔKn. The second hybrid iteration predicts the surface pressure at the shock wave region (up to $X/L < 0.6$) suitably while it underpredicts the pressure at the end of plate. However, the third iteration correctly captures the surface pressure despite supplying incorrect boundary conditions after $X/L = 0.6$; see Figure 12(c).

Figure 13 compares the VDF(c_y) derived from the DSMC simulation with the Maxwellian and CE distribution. The interfacial cell is located at $(X/L, Y/H) = (0.65, 0.18)$, which is indicated by symbol x in Figure 7. This location is one of the most critical interfacial points because it is almost where the maximum difference between the NS and DSMC solutions is expected to occur, see Figure 12(c). Evidently, the real VDF would be different from both the Maxwellian and CE distributions. As is seen, the two distributions are quite close to each other. This indicates that there is not an obvious superiority in sampling procedure to use the CE distribution at the interface. Schwartzentruber *et al.* [21] also reported similar conclusions through comparing the DSMC's VDF result with the CE one. Therefore, we simply employ sampling from the Maxwellian distribution in the current work. It is also important to note that the sampling from the Maxwellian VDF has been reported by other researchers as well, for example, see Reference [70]. They also report achieving suitable accuracy.

Figure 14 illustrates the present hybrid solutions for the gas velocity slip and temperature jump magnitudes and the friction and heat flux coefficients along the flat plate surface and compares them with those of the current pure DSMC prediction. As is seen, there are good agreements between them. The computational domain and the number of simulator particles for the DSMC solver decrease in the hybrid frame by 66% and 53%, respectively. Comparing with a pure DSMC solver, the current hybrid frame needs at least 40% less computational time including the time required for hybrid cycle iterations.

3.2. Test 2: microcylinder

A hypersonic gas flow over a microscale cylinder is a suitable test case for any hybrid algorithm because the length scales of the flow change considerably in the same solution domain. Here, we consider the flow of argon with a Mach number of 10 over a microcylinder with a diameter of 0.3024 μm. The inlet flow temperature is 200 K while the surface temperature of the microcylinder is 500 K. The freestream number density is 4.247×10^{26} molecules/m^3, and the global Knudsen

DOI: 10.1002/fld

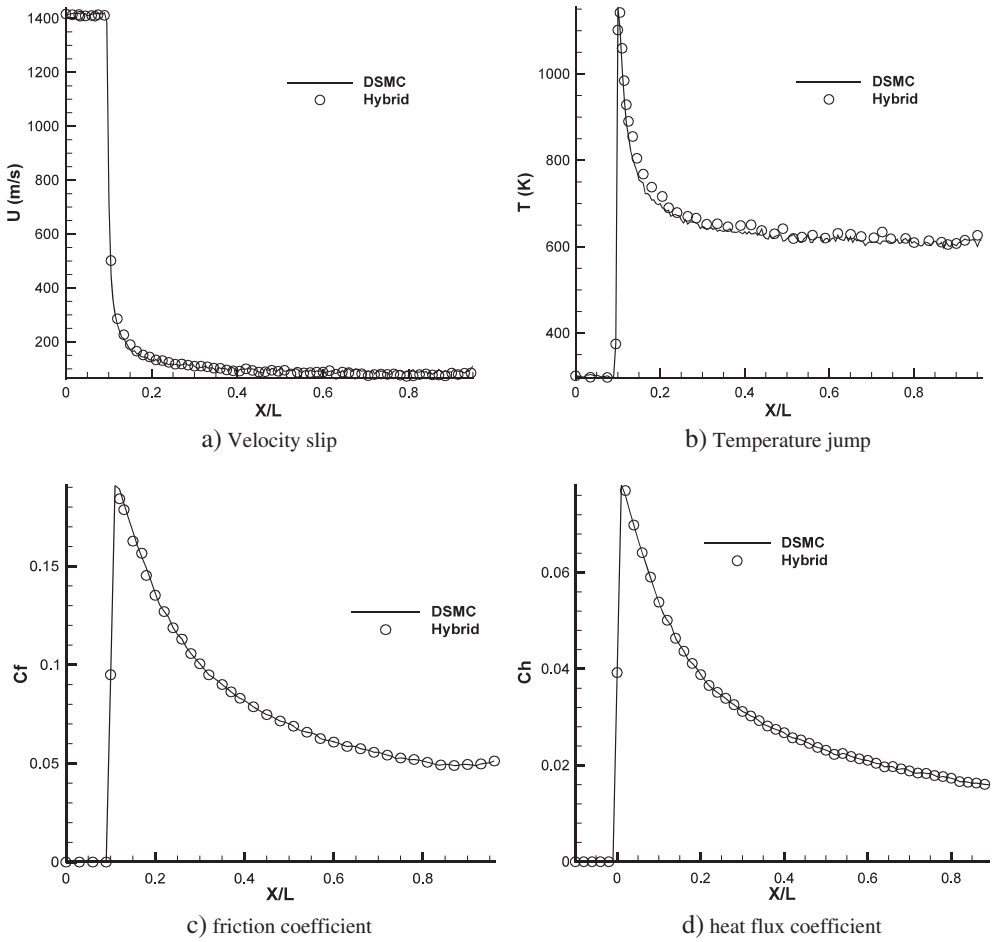


Figure 14. Distributions of different flow properties at the flat plate surface using the pure DSMC and hybrid solvers, test 1. (a) Velocity slip, (b) temperature jump, (c) friction coefficient, and (d) heat flux coefficient.

number is 0.01 based on the microcylinder diameter and the freestream mean free path. We utilize a time step of 2.0×10^{-12} s in the current calculations. We use one million particles for a full DSMC simulation. The Courant number for the NS solver is set as 0.40. Similar to our previous test case, we first evaluate the performance of our DSMC and NS solutions individually. The DSMC and NS meshes are unstructured and have totals of 30,000 and 44,000 triangular cells, respectively. We performed our DSMC and NS solutions on different mesh sizes to ensure their accuracies. The NS mesh size is selected after a mesh refinement study so that the final solutions reported in this paper are surely independent from the cell size. The DSMC mesh is chosen such that $(\Delta x, \Delta y) < \lambda/3$. However, the cells located behind the cylinder and in the wake region are much larger than those located around and in front of the cylinder. In fact, as the flow becomes rarefied more seriously in the wake region, it is necessary to use sufficiently large cells to ensure sufficient particles inside each cell located in the region. We put at least 35 particles in each cell. As will be shown later, the current NS, DSMC, and hybrid solutions are in proper agreement with the magnitudes reported in the past reliable publications. They indicate that we have chosen suitable cell sizes and number of particles in this study.

Figure 15 shows the solution domain and the applied boundary conditions for the NS (top) and DSMC (bottom) solvers. As is seen, the computational domain is different for the two solvers. We implement freestream boundary conditions at the inlet and outlet sections and symmetry conditions at the centerline. First, we validate our DSMC and NS solutions against the solutions reported by

DOI: 10.1002/fld

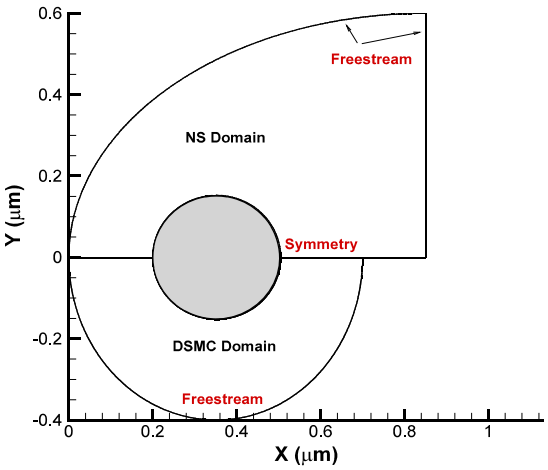
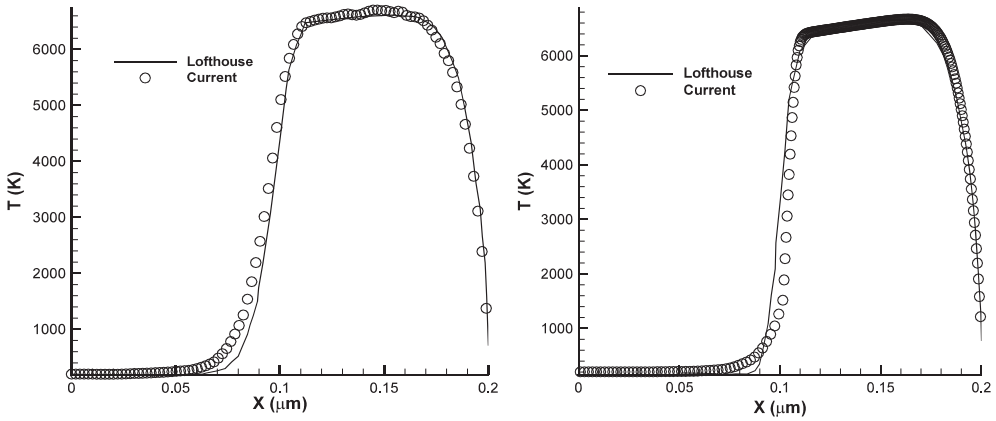


Figure 15. Domain for the NS (top) and DSMC (bottom) solvers and the chosen boundary conditions to treat the flow over a microcylinder, test 2.



a) DSMC solution b) NS solution

Figure 16. The current DSMC and NS temperature distributions along the microcylinder stagnation line and comparison with DSMC (a) and NS (b) solutions of Lofthouse [65], test 2.

Lofthouse [65]. Figure 16 presents the temperature distributions along the stagnation line, that is, a line extending ahead from the stagnation point at (0.2 μm, 0) to (0, 0). The figure compares the current DSMC and NS solutions with those reported by Lofthouse [65]. Except a small difference near the shock starting point, there are suitable agreements between the NS and DSMC solutions. We next compare the full DSMC solution with the NS one. Figure 17 illustrates the velocity components, temperature, and Mach number contours for both of our NS and DSMC solvers. The top face contours show the NS solution, and the bottom ones show the DSMC solution. As before, the continuum and stochastic particle solutions perform differently in predicting the shock wave thickness, the boundary layer over the microcylinder, and the wake region behind the microcylinder. Comparing with the NS solution, the DSMC solution provides a much thicker shock wave and locates it upstream of the microcylinder. Otherwise, the NS and DSMC solutions perform almost similar solutions in the other parts of the computational domain.

To determine the exact position of the DSMC–NS interface, we calculate the breakdown parameter on the flow field using both the stochastic and continuum solvers. Figure 18 shows the contours of $\mathrm{Kn}_{\mathrm{GLL,max}}$ calculated by both the NS and DSMC solvers. The figure shows boundaries with $\mathrm{Kn}_{\mathrm{GLL,max}} = 0.05$. Indeed, the regions of shock wave, boundary layer, and wake are the places where the breakdown parameter magnitude exceeds the threshold value. These locations correspond

DOI: 10.1002/fld

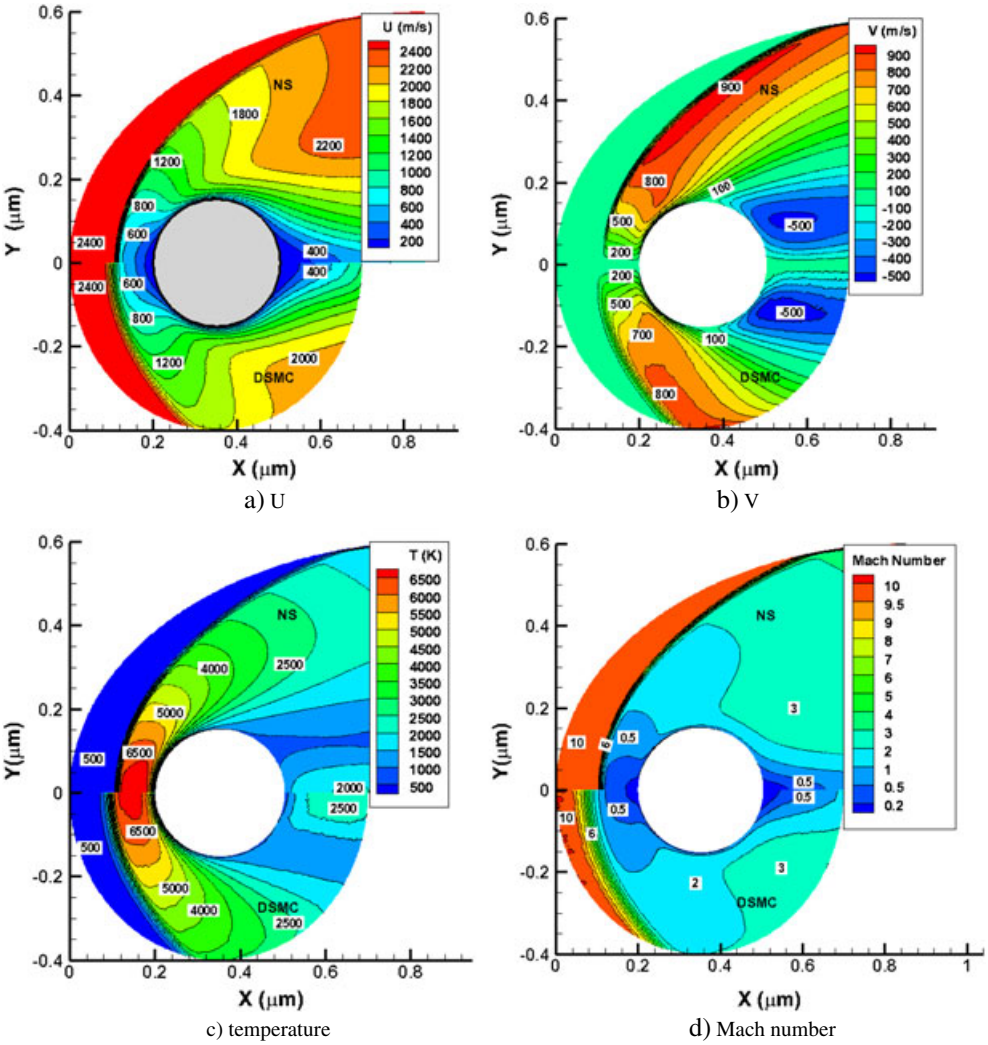


Figure 17. Comparing contours of different flow field variables using the NS and DSMC solvers, test 2. (a) U, (b) V, (c) temperature, and (d) Mach number.

to the places where the NS solution deviates considerably from the corresponding DSMC one. In our hybrid frame, the DSMC solver is applied in these regions. Similar to the previous test case, the application of continuum breakdown parameter to the NS solution determines that the DSMC solver should be used in the wake, boundary layer, and the entire shock wave regions. However, we restrict our DSMC solver to the shock wave region having a width equal to the width of wake behind the cylinder. We will show that this interface selection can reduce the computational costs effectively while the solution accuracy is preserved completely. Figure 19 shows the location of final interface. The DSMC region is expanded up to $Y = 0.20\mu$m, while the breakdown parameter predicts a height of $Y = 0.18\mu$m for the non-equilibrium regions. In fact, a small width of $Y = 0.02\mu$m is considered as the overlap region. Comparing with Figure 15, we observe that the DSMC computational domain is decreased by about 40%.

Before presenting the results of our hybrid algorithm, we need to elaborate why some parts of the shock wave have been taken outside the chosen molecular region. The first reason for this choice is to decrease the size of DSMC domain as much as possible and consequently to increase the computational efficiency of our hybrid algorithm more effectively. Schwartzentruber *et al.* [21] have already discussed the physical reason for this choice. They applied a hybrid DSMC–NS simulation

DOI: 10.1002/fld

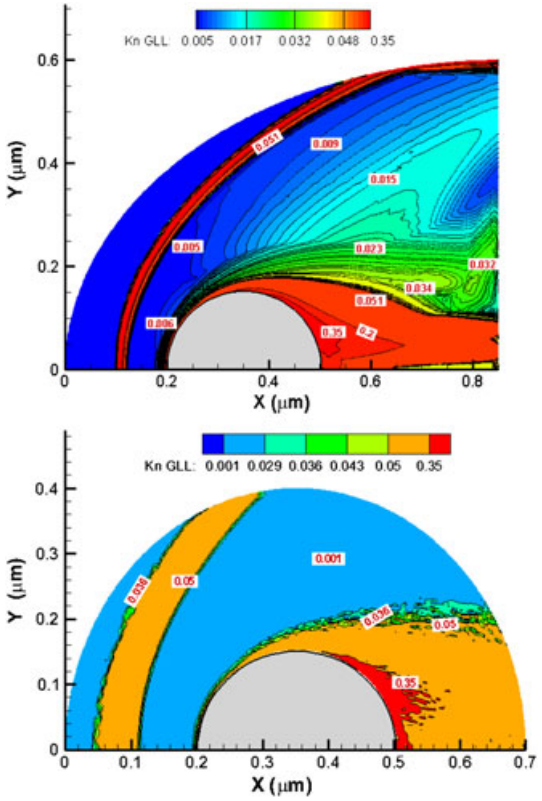


Figure 18. Contours of $Kn_{GLL,max}$ calculated by the NS (top) and DSMC (bottom) solvers, test 2.

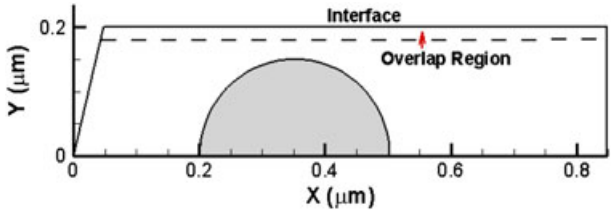


Figure 19. Simulation domain in the hybrid frame, test 2.

over a cylinder and reported that the post-shock states are almost the same for both the DSMC and NS solutions. Therefore, they concluded that the shock wave modeling may not necessarily require a stochastic-based solver. However, the molecular simulations are necessary for the boundary-layer and nearwake regions. Figure 20 shows the DSMC temperature distribution at the overlap, that is, $Y = 0.2\mu$m, and compares it with that of the NS solvers. As is observed, the post-shock solutions are identical for both solvers, although the NS equations locate the shock wave a bit closer to the microcylinder surface. Therefore, the conclusion of Schwartzentruber *et al.* [21] is reconfirmed here. Although our chosen interface intersects the non-equilibrium region of shock wave, it does not affect the accuracy of our hybrid solution for the surface properties. In other words, our hybrid solution is converged (AAE(ϕ) $< \varepsilon$) at the interface and achieves correct values for the surface heat flux and shear stress magnitudes. The accuracy is evaluated against our DSMC solution and that of Reference [66]; see Figure 21. We should note that the correct position of the interface can affect the accuracy of our hybrid solution effectively. Achieving accurate solutions for the heat flux and shear stress magnitudes, we are convinced that the chosen interface position should be correct. It should be noted that although the simulation of cylinder flow using a hybrid DSMC–NS solver has been already performed by Schwartzentruber *et al.* [21], the choice of molecular domain used in

DOI: 10.1002/fld

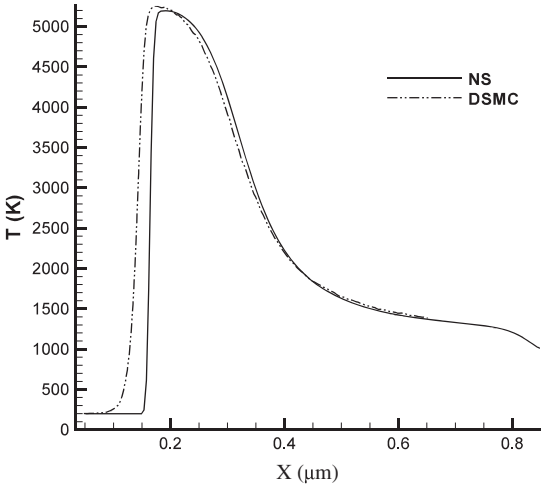


Figure 20. Comparing the DSMC temperature distribution with that of the NS solution at the interface, test 2.

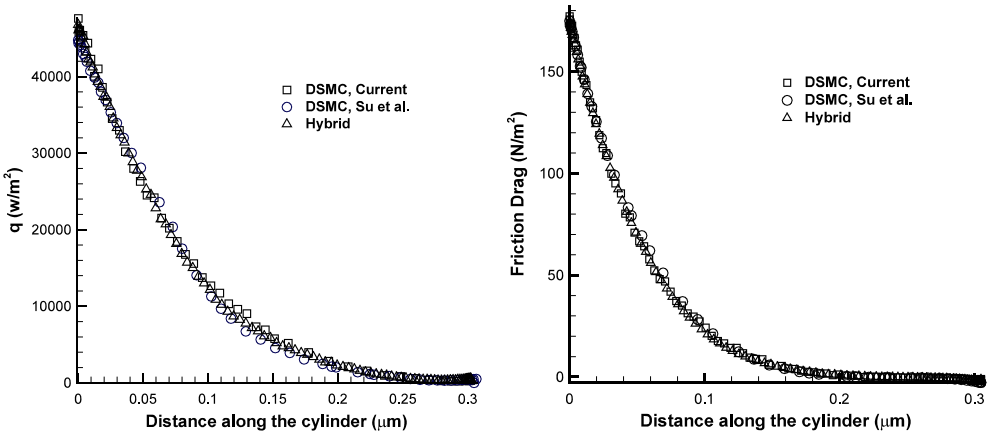


Figure 21. Heat flux (left) and friction drag per area (right) at the microcylinder surface using the current DSMC and hybrid solvers and comparison with the DSMC solver of Su *et al.* [66], test 2.

this study is different from that of the reference. Additionally, unlike the reference, we did not solve the NS equations during hybrid frame iterations. This consequently resulted in less computational times for our hybrid frame.

Figure 21 shows the final solution for the heat flux and friction force magnitudes per unit area over the microcylinder using both the current hybrid and DSMC solvers as well as the DSMC solution of Su *et al.* [66]. The number of iterations to reach the final hybrid interface has been three. In the current test, we aim to use the hybrid frame and to calculate the heat flux and shear stress values over the cylinder surface. Therefore, we used $\phi = (\tau_s, q_s)$ in Equation (48) as the convergence criterion. Consequently, $\mathrm{AAE}(\phi) = \max(\mathrm{AAE}(\tau_s), \mathrm{AAE}(q_s))$. The cutoff value is set as $\varepsilon = 10^{-3}$. The hybrid solution agrees well with both DSMC solutions. As is observed, the current hybrid method is able to obtain critical parameters in the flow field, such as the heat transfer rate and the friction drag magnitudes, very accurately considering much less computational expenses. Although the upper part of the hybrid interface is located in the shock wave region, where the boundary conditions are supplied from the continuum solver, the hybrid solver provides the heat flux and shear stress magnitudes much closer to the pure DSMC solution. This accuracy indirectly confirms the correct interface location in this specific problem.

DOI: 10.1002/fld

Computational efficiency of any hybrid DSMC–NS solver strongly depends on the problem size, number of cells, number of particles, and the convergence rate. For the microcylinder test case, the size of DSMC computational domain is at least 40% less in our hybrid frame. This subsequently decreased the number of required particles to 350,000 particles compared with one million particles used for our full DSMC simulation. In other words, there is a decrease of 41% in the number of DSMC cells. Since the DSMC simulation time linearly changes with the change of total number of particles, the DSMC simulation time decreases to 33% using our hybrid solver. As there is negligible NS solution overhead in using our hybrid frame, the total gain of the current hybrid frame would be 65% comparing with the time required for hybrid cycle iterations.

3.3. Comments on current hybrid solution

As was described in Section 2.4.2, the overlap region may be extended to a new location during the eight steps of our hybrid solution algorithm. In the microcylinder test, the first interface location was at a horizontal line crossing the $\mathrm{Kn}_{\mathrm{GLL}} = 0.05$ region. At this interface, the solutions of the NS and DSMC weakly fitted each other. During the hybrid iteration, the interface location was traversed a bit farther. Eventually, the hybrid algorithm converged to an interface performing a better match between the two solutions; see Figure 20. Our conclusion was that the one-way hybrid algorithm would not necessarily converge unless the boundary conditions supplied by the NS solver suitably fitted the DSMC solution along a good portion of the achieved interface. In other words, the achieved interface may contain some limited non-equilibrium regions. It is because the post-shock properties are almost identical for both the DSMC and NS solvers; see Figure 20. Therefore, the achieved interfaces for both the flat plate and cylinder cases may include some limited regions of weak shock. This can be counted as another difference between our one-way hybrid algorithm and those reported in literature [18, 20, 21].

It should be noted that the current hybrid algorithm can be applied only to test cases in which there are distinguishable boundaries between the two equilibrium and non-equilibrium regions, that is, in flows with low Kn numbers. In cases with full non-equilibrium regions, it is recommended to apply a full DSMC method and to simulate the entire computational domain. For example, the hybrid algorithm is not recommended to be applied for treating the transition flow regime. So, if one wishes to repeat the test cases presented in this work at higher Kn numbers, one will find that the shock waves become much thicker and that the rarefaction regions will cover most of the solution domain. In such cases, it is impossible to distinct the two equilibrium and non-equilibrium regions from each other.

There is some idea to improve the current one-way algorithm. For example, one may suggest we assume a two-way coupling, start updating the NS boundary conditions, and solve the region via the NS solver. Evidently, there will be some improvements in the achieved boundary conditions and that the DSMC solution will be more accurate in the next step. As a result, this strategy can eventually provide a suitable solution for the far-field domain. This idea is not new and can be found in the literature [18, 20, 21]. Using this strategy, the computational cost will increase because it needs to re-solve the NS region repeatedly. On the other hand, the algorithm needs more iteration to locate the interface boundary. Consequently, this strategy would not be recommended in solving the test cases, where we intend to calculate the surface properties. In such cases, the recommendation is to apply the current one-way algorithm. As was shown previously, the hybrid solution accuracy presented in Section 3 justifies the use of the current one-way algorithm in solving such specific problems.

4. CONCLUSION

We presented a numerical simulation tool that spatially couples a continuum solver for the NS equation with a stochastic particle method (DSMC) for steady-state simulations of non-equilibrium gas flows. We extended a one-way state-based hybrid DSMC–NS framework to solve the mixed rarefied and nonrarefied flows over nanoplate and microcylinder test. We used the continuum rhoCentralFoam solver from OpenFOAM and developed a stochastic particle method solver, that is, dsmcFoam, to perform this study. The rarefied flow regions were solved using the DSMC solver

DOI: 10.1002/fld

while the NS solver was used to provide the required boundary conditions at the interface located in the near-equilibrium region. We used a Dirichlet–Dirichlet boundary conditions coupling to transfer the information from the continuum solver to the DSMC one at their interface. Our results showed that the high speed flow over the nanoflat plate would be dominated by the shock-wave/boundary layer interaction. Additionally, the NS equations could not predict correct velocity slip and temperature jump magnitudes, especially at the shock wave/plate interaction point even though the flow Knudsen number was quite low, that is, $\mathrm{Kn} = 0.00129$. Moreover, if the rarefied domain was small, the boundary condition errors at the hybrid interface would affect the interior domain solution. However, increasing the size of rarefied domain during the hybrid iterations would prevent negative effects of the incorrect NS boundary conditions on the interior rarefied flow field behavior. For the flow around a microcylinder, we observed that the NS equations would perform remarkable inaccuracy in the shock wave, boundary layer, and wake regions. Although these regions should be solved with a molecular algorithm, we observed that the post-shock flow properties including the surface properties would not be influenced by the exact solution of the shock wave. Consequently, we decreased the size of molecular region properly, used the DSMC solver, and solved only a portion of the shock wave, located in the region with a height equal to the height of the microcylinder wake. We presented the results of our hybrid solver for the slip velocity and temperature jump distributions as well as the heat flux and friction drag magnitudes and compared them with those of our pure DSMC simulator. We achieved suitable accuracy for the hybrid method while spending much less computational time. For example, the required computational times for our hybrid solver were about 40% and 35% of the times required for our pure DSMC solver in solving the flow over the flat plate and cylinder cases, respectively.

ACKNOWLEDGEMENTS

The authors would like to thank the financial support received from the Centre for Research and Technology (Sharif University of Technology) and the Centre for High-Technology Development (the Ministry of Industries and Business (Iran)) under contract no. 8795-S-006.

REFERENCES

1. Kandlikar SG, Garimella S, Li D, Colin S, King MR. *Heat Transfer and Fluid Flow in Minichannels and Microchannels*. Elsevier: London, 2006.
2. Bird GA. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon: Oxford, 1994.
3. Wadsworth DC, Erwin DA. One-dimensional hybrid continuum/particle simulation approach for rarefied hypersonic flows. *AIAA Paper 1990–1690, 5th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, Seattle, WA, June 18–20, 1990.
4. Hash DB, Hassan HA. Assessment of schemes for coupling Monte Carlo and Navier–Stokes solution methods. *Journal of Thermophysics and Heat Transfer* 1996; **10**:242–249.
5. Bourgat JF, LeTallec P, Tidriri MD. Coupling Boltzmann and Navier–Stokes equations by friction. *Journal of Computational Physics* 1996; **127**:227–245.
6. LeTallec P, Mallinger F. *Coupling Boltzmann and Navier–Stokes Equations by Half Fluxes, Journal of Computational Physics* 1997; **136**:51–67.
7. Tiwari S, Klar A. An adaptive domain decomposition procedure for Boltzmann and Euler equations. *Journal of Computational and Applied Mathematics* 1998; **90**:223–237.
8. Garcia AL, Bell JB, Crutchfield WY, Alder BJ. Adaptive mesh and algorithm refinement using direct simulation Monte Carlo. *Journal of Computational Physics* 1999; **154**:134–155.
9. Wijesinghe HS, Hornung RD, Garcia AL, Hadjiconstantinou NG. Three-dimensional continuum-atomistic simulations for multiscale hydrodynamics. *Journal of Fluids Engineering* 2004; **126**(5):768–776.
10. Wijesinghe HS, Hadjiconstantinou NG. Discussion of hybrid atomistic–continuum methods for multiscale hydrodynamics. *International Journal of Multiscale Computational Engineering* 2004; **2**(2):189–202.
11. Roveda R, Goldstein DB, Varghese PL. Hybrid Euler/particle approach for continuum/rarefied flows. *Journal of Spacecraft and Rockets* 1998; **35**(3):258–265.
12. Roveda R, Goldstein DB, Varghese PL. Hybrid Euler/direct simulation Monte Carlo calculation of unsteady slit flow. *Journal of Spacecraft and Rockets* 2000; **37**(6):753–760.
13. Aktas O, Aluru NR. A combined continuum/DSMC technique for multiscale analysis of microfluidic filters. *Journal of Computational Physics* 2002; **178**(2):342–372.
14. Wang WL, Boyd ID. Hybrid DSMC-CFD simulations of hypersonic flow over sharp and blunted bodies. *AIAA Paper 2003–3644, 36th AIAA Thermophysics Conference*, Orlando, FL, June 23–26, 2003.

DOI: 10.1002/fld

15. Sun Q, Boyd ID, Candler G. A hybrid continuum/particle approach for modelling subsonic, rarefied gas flows. *Journal of Computational Physics* 2004; **194**(1):256–277.
16. Fan J, Shen C. Statistical simulation of low-speed unidirectional flows in transition regime. In *Rarefied Gas Dynamics*, Brun R, Campargue R, Gatignol R, Lengrand JC (eds), Cepadus Editions. Toulouse: France, 1999.
17. Degond P, Jin S, Mieussens L. A smooth transition between kinetic and hydrodynamic equations. *Journal of Computational Physics* 2005; **209**:665–694.
18. Wu JS, Lian YY, Cheng G, Koomullil RP, Tseng KC. Development and verification of a coupled DSMC–NS scheme using unstructured mesh. *Journal of Computational Physics* 2006; **219**:579–607.
19. Kolobov VI, Arslanbekov RR, Aristov VV, Frolova AA, Zabelok SA. Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement. *Journal of Computational Physics* 2007; **223**(2):589–608.
20. Schwartzentruber TE, Scalabrin LC, Boyd ID. A modular particle-continuum numerical method for hypersonic non-equilibrium gas flows. *Journal of Computational Physics* 2007; **225**:1159–1174.
21. Schwartzentruber TE, Scalabrin LC, Boyd ID. Hybrid particle–continuum simulations of non-equilibrium hypersonic blunt-body flow fields. *Journal of Thermophysics and Heat Transfer* 2008; **22**(1):29–37.
22. Abbate G, Thijsse BJ, Kleijn CR. Validation of a hybrid Navier–Stokes/DSMC method for multiscale transient and steady-state gas flows. *International Journal of Multiscale Computational Engineering* 2008; **6**(1):1–12.
23. John B, Damodaran M. Computation of head–disk interface gap micro flow-fields using DSMC and continuum–atomistic hybrid methods. *International Journal for Numerical Methods in Fluids* 2009; **61**(11):1273–1298.
24. Degond P, Dimarco G, Mieussens L. A multiscale kinetic-fluid solver with dynamic localization of kinetic effects. *Journal of Computational Physics* 2010; **229**:4907–4933.
25. Degond P, Dimarco G. Fluid simulations with localized Boltzmann upscaling by direct simulation Monte-Carlo. *Journal of Computational Physics* 2012; **231**:2414–2437.
26. Degond P, Dimarco G, Pareschi L. The moment guided Monte Carlo method. *International Journal for Numerical Methods in Fluids* 2011; **67**(2):189–213.
27. Papp JL, Wilmoth RG, Chartrand CC, Dash SM. Simulation of high-altitude plume flow fields using a hybrid continuum CFD/DSMC approach. *AIAA Paper 2006-4412, 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Sacramento, California, July 9–12, 2006.
28. Gimelshein SF, Levin DA, Alexeenko AA. Modeling of chemically reacting flows from a side jet at high altitudes. *Journal of Spacecraft and Rockets* 2004; **41**(4):582–591.
29. Nanson RA. Navier-Stokes/direct simulation Monte Carlo modeling of small cold gas thruster nozzle and plume flows. *PhD thesis*, Worcester Polytechnic Institute, 2002.
30. Darbandi M, Schneider GE. Momentum variable procedure for solving compressible and incompressible flows. *AIAA Journal* 1997; **35**:1801–1805.
31. Darbandi M, Schneider GE. Performance of an analogy-based all-speed procedure without any explicit damping. *Computational Mechanics* 2000; **26**:459–469.
32. Darbandi M, Mokarizadeh V. A modified pressure-based algorithm to solve the flow fields with shock and expansion waves. *Numerical Heat Transfer, Part B* 2004; **46**:497–504.
33. Darbandi M, Vakilipour S. Developing consistent inlet boundary conditions to study the entrance zone in microchannels. *Journal of Thermophysics and Heat Transfer* 2007; **21**(3):596–607.
34. Darbandi M, Roohi E, Mokarizadeh V. Conceptual linearization of Euler governing equations to solve high speed compressible flow using a pressure-based method. *Numerical Methods for Partial Differential Equations* 2008; **24**(2):583–604.
35. Vakilipour S, Darbandi M. Advancement in numerical study of gas flow and heat transfer in microchannels. *Journal of Thermophysics and Heat Transfer* 2009; **23**(1):205–208.
36. Darbandi M, Vakilipour S. Solution of thermally developing zone in short micro/nano scale channels. *Journal of Heat Transfer* 2009; **131**:044501.
37. Roohi E, Darbandi M. Extending the Navier–Stokes solutions to transition regime in two-dimensional micro-/nanochannel flows using information preservation scheme. *Physics of Fluids* 2009; **21**:082001.
38. Roohi E, Darbandi M, Mirjalili V. DSMC solution of subsonic flow through micro-nano scale channels. *Journal of Heat Transfer* 2009; **131**(9):092402.
39. Darbandi M, Roohi E. Study of gas flows in micro-nano scale nozzles using an unstructured DSMC solver. *Microfluidics Nanofluidics* 2011; **10**(2):321–335.
40. Darbandi M, Roohi E. DSMC simulation of subsonic flow through nanochannels and micro/nano steps. *International Communication in Heat and Mass Transfer* 2011; **38**(10):1444–1449.
41. Roohi E, Darbandi M. Recommendations on performance of parallel DSMC algorithm in solving subsonic nanoflows. *Applied Mathematical Modelling* 2012; **36**(5):2314–2321.
42. Ejtehadi O, Roohi E, Abolfazli J. Investigation of basic molecular gas structural effects on hydrodynamics and thermal behaviours of rarefied shear driven micro/nano flow using DSMC. *International Communication in Heat and Mass Transfer* 2012; **39**(3):439–448.
43. Mohammadzadeh A, Roohi E, Niazmand H, Stefanov S, Myong RS. Detailed investigation of thermal and hydrodynamic behaviour in micro/nano cavity using DSMC. *Physical Review E* 2012:056305.
44. Darbandi M, Roohi E. Applying a hybrid DSMC/Navier–Stokes frame to explore the effect of splitter catalyst plates in micro/nano propulsion systems. *Sensors & Actuators: A. Physical* 2013; **189**(1):409–419. DOI: http://dx.doi.org/10.1016/j.sna.2012.09.005.

45. OpenFOAM: the Open Source CFD Toolbox. User Guide, 2009. Version 1.6.
46. Stefanov S. On DSMC calculations of rarefied gas flows with small number of particles in cells. *SIAM Journal on Scientific Computing* 2011; **33**:677–702.
47. Liou WW, Fang Y. *Microfluid Mechanics: Principles and Modeling*. McGraw-Hill: London, 2006.
48. Scanlon T, Roohi E, White C, Darbandi M, Reese J, An os. Parallel dsmc code for rarefied gas flows in arbitrary geometries,. *Computers & Fluids* 2010; **39**(10):2078–2089.
49. Macpherson GB, Nordin N, Weller HG. Particle tracking in unstructured, arbitrary polyhedral meshes for use in CFD and molecular dynamics. *Communication in Numerical Methods in Engineering* 2009; **25**(3):263–273.
50. Anderson JD. *Hypersonic and High Temperature Gas Dynamics*. McGraw Hill: USA, 1989.
51. Greenshields CJ, Weller HG, Gasparini L, Reese JM. Implementation of semi- discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows. *International Journal for Numerical Methods in Fluids* 2010; **63**:1–21.
52. Kurganov A, Tadmor E. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of Computational Physics* 2001; **160**:241–282.
53. Kurganov A, Noelle S, Petrova G. Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations. *SIAM Journal on Scientific Computing* 2001; **23**:707–740.
54. Le N. Nonequilibrium boundary conditions for the Navier–Stokes–Fourier equations in hypersonic gas flow simulations. *PhD thesis*, University of Strathclyde, UK, 2010.
55. Lockerby DA, Reese JM, Emerson DR, Barber RW. Velocity boundary condition at solid wall in rarefied gas calculation. *Physical Review E* 2004; **70**:017303.
56. Tiwari S. Coupling of the Boltzmann and Euler equations with automatic domain decomposition. *Journal of Computational Physics* 1998; **144**:710–726.
57. Levermore D, Morokoff WJ, Nadiga BT. Moment realizability and the validity of the Navier–Stokes equations for rarefied gas dynamics. *Physics of Fluids* 1998; **10**(12):3214–3226.
58. Boyd I, Chen G, Candler GV. Predicting failure of the continuum fluid equations in transitional hypersonic flows. *Physics of Fluids* 1995; **7**:210–219.
59. Garcia AL, Bell JB, Crutchfield WY, Alder BJ. Adaptive mesh and algorithm refinement using direct simulation Monte Carlo. *Journal of Computational Physics* 1999; **154**:134–139.
60. Wang WL, Boyd ID. Predicting continuum breakdown in hypersonic viscous flows. *Physics of Fluids* 2003; **15**:91–100.
61. Lian YY, Chen YS, Tseng KC, Wu JS, Wu B, Yang L. Improved parallelized hybrid DSMC–NS method. *Computers & Fluids* 2011; **45**(1):254–260.
62. Schwartzentruber TE, Boyd ID. A hybrid particle-continuum method applied to shock waves. *Journal of Computational Physics* 2006; **215**:402–416.
63. Amiri A, Roohi E, Niazmand H, Stefanov S. Low speed/low rarefaction flow simulation in micro/nano cavity using DSMC method with small number of particle per cells. *Journal of Physics: Conference Series* 2012; **362**:012007.
64. Kaplan CR, Oran ES. Nonlinear filtering for low-velocity gaseous microflows. *AIAA Journal* 2002; **40**:82–90.
65. Lofthouse AJ. Nonequilibrium hypersonic aerothermodynamics using the direct simulation Monte Carlo and Navier–Stokes models. *PhD thesis*, University of Michigan, USA, 2008.
66. Su CC, Tseng KC, Cave HM, Wu JS, Lian YY, Kuo TC, Jermy MC. Implementation of a transient adaptive sub-cell module for the parallel-DSMC code using unstructured grids. *Computers & Fluids* 2010; **39**:1136–1145.
67. LeBeau GJ, Boyles KA, Lumpkin FE. Virtual sub-cells for the direct simulation Monte Carlo method. *AIAA Paper 2003–3644, 41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, June 6–9, 2003.
68. Shen C, Fan J, Xie C. Statistical simulation of rarefied gas flows in micro-channels. *Journal of Computational Physics* 2003; **189**:512–526.
69. Greenshields CJ, Reese JM. Rarefied hypersonic flow simulations using the Navier–Stokes equations with non-equilibrium boundary conditions. *Progress in Aerospace Sciences* 2012; **52**:80–87.
70. La Torre F, Kenjeres S, Kleijn C, Moerel J. Hybrid simulations of rarefied supersonic gas flows in micro-nozzles. *Computers & Fluids* 2011; **49**:312–322.

DOI: 10.1002/fld