A new cut-cell algorithm for DSMC simulations of rarefied gas flows around immersed moving objects

Wenjie Jin[1,*], J. Ruud van Ommen, Chris R. Kleijn

Delft University of Technology, Department of Chemical Engineering, 2629 HZ Delft, The Netherlands

Abstract

Direct Simulation Monte Carlo (DSMC) is a widely applied numerical technique to simulate rarefied gas flows. For flows around immersed moving objects, the use of body fitted meshes is inefficient, whereas published methods using cut-cells in a fixed background mesh have important limitations. We present a novel cut-cell algorithm, which allows for accurate DSMC simulations around arbitrarily shaped moving objects. The molecule-surface interaction occurs exactly at the instantaneous collision point on the moving body surface, and accounts for its instantaneous velocity, thus precisely imposing the desired boundary conditions. A simple algorithm to calculate the effective volume of cut cells is presented and shown to converge linearly with grid refinement. The potential and efficiency of method is demonstrated by calculating rarefied gas flow drag forces on steady and moving immersed spheres. The obtained results are in excellent agreement with results obtained with a body-fitted mesh, and with analytical approximations for high-Knudsen number flows.

Keywords: DSMC, cut-cell, overlap volume, drag

1. Introduction

The Direct Simulation Monte Carlo (DSMC) is a well-established, discrete particle based, numerical method for the computation of rarefied gas flows (Bird, 1994). It is widely applied in fields such as aerospace engineering and Micro-Electro-Mechanical Systems (MEMS), where the rarefaction is important due to low pressures and small dimensions, respectively (Oran et al., 1998; Hong et al., 2008; Akhlaghi et al., 2012).

Compared to possible alternatives, such as Molecular Dynamics (Frenkel and Smit, 2002), multiple-particle collision dynamics (Gompper et al., 2009), dissipative particle dynamics (Hoogerbrugge and Koelman, 1992) and Lattice Boltzmann (Chen and Doolen, 1998), DSMC has grown to be the most widely applied and validated method for simulating gas flows in the rarefied regime, i.e. the transition regime between continuum flow and free molecular flow.

In conventional DSMC simulations, the flow domain is discretized into a number of fixed shape grid cells, wherein the simulated gas molecules can move freely. The grid cells are used exclusively in process of randomly selecting pairs of gas molecules as collision partners, and for calculating average flow properties. When the walls of solid objects contribute as part of the flow boundaries, boundary conditions at those walls are imposed by prescribing appropriate molecule-wall collision laws. Here, the object walls may (i) be approximated by staircases with local grid refinement (Bird, 1994; Zabelok et al., 2015), (ii) coincide with grid cell faces (requiring the use of non-Cartesian, body fitted meshes to accommodate complexly shaped immersed bodies), or (iii) cut through grid cells.

The first approach can be seen as a special case of the coincided grid with an additional staircase approximation to the walls. The last approach has been called the cut-cell method

*Corresponding author

Email address: W.Jin-1@tudelft.nl (Wenjie Jin)

[1]Phone: +31 152787084

(Zhang and Schwartzentruber, 2012; Burt et al., 2012). It is somewhat similar to the so-called immersed boundary method (IBM), proposed by Peskin (Peskin, 2002) to impose boundary conditions on the walls of immersed objects in continuum based flow simulations. However, whereas in IBM fictitious external forces, localized near the boundary, have to be imposed to satisfy boundary conditions, in DSMC cut-cell methods such fictitious forces are not needed, as the boundary conditions are imposed explicitly and exactly through the molecule-wall collision laws. On the other hand, the cell effective volume has to be computed for the cut cells in order to achieve the correct molecular collision probabilities.

The cut-cell method is particularly advantageous over the use of body fitted meshes when the immersed object moves with respect to the grid, e.g. when studying gas flows in MEMS with oscillating parts, or Brownian aerosol particle movement in gas flows. In such situations, the use of body fitted meshes would require body fitted grids to be regenerated at each time step, which is computationally inefficient.

For simply shaped moving immersed objects, of which the surface is limited to flat planes aligned with the cell faces, the cut-cell method has been demonstrated for applications with 1-D moving piston (Rader et al., 2011) and turbomolecular pumps (Versluis et al., 2009). The cut-cell method has also been demonstrated for applications with 2-D static immersed bodies (Lo et al., 2014).

For complexly shaped 3-D immersed objects, two main approaches for cut-cell DSMC simulations have been proposed (Zhang and Schwartzentruber, 2012; Burt et al., 2012), which mainly differ by the way in which the immersed solid object surfaces are represented numerically. Both approaches make use of random markers to distinguish between the inside and outside of the immersed object.

The first approach (Zhang and Schwartzentruber, 2012; Burt et al., 2012) represents the approximate shape of the immersed surface by small contiguous triangular facets. Molecules interact with these facets during the molecular streaming step in the DSMC algorithm. In this approach, the facet size determines the accuracy by which the surface, and thus the location of the boundary, is being represented, whereas the grid size determines the resolution of the flow field simulations. The effective volume of cut cells, which is needed during the molecule-molecule collision step of the algorithm, is computed either by polyhedral decomposition utilizing the facets and the cell faces, or by Monte Carlo random markers. Shrestha et al. (2015) have applied the latter to a 3-D moving object, and simulated the Brownian motion of a spherical particle immersed in a rarefied gas. It is stated that the random marker based approach is easy to implement but computationally expensive.

The second approach (Burt et al., 2012) uses an analytical expression for the immersed surface shape as an input. This shape is subsequently approximated by planar faces, which are determined by finding the two smallest possible cuboids which respectively contain all random points inside, and outside, of the immersed object. This method may result in gaps between the faces, which become significant when the ratio of the surface curvature radius to the cell size is small.

Summarizing the present state-of-art, there is a clear need for a computationally efficient DSMC cut-cell algorithms that allow for an accurate, gap-free representation of immersed surfaces and the cell effective volumes. In the present work, we present, demonstrate and validate such an algorithm.

Compared to earlier cut-cell methods, (i) our algorithm utilizes an exact analytical representation of the immersed surface for computing the DSMC particle streaming step and thus the particle-surface interaction. Therefore, this step does not rely on a triangulated surface representation and does not leave gaps between approximated surface elements; (ii) in our algorithm both the flow resolution and the accuracy of the cut-cell effective volume scale with cell size; (iii) the molecules collide and interact exactly at the instantaneous location of the moving surface; (iv) our algorithm avoids the use of expensive Monte Carlo methods in calculating the effective volume of cut cells. Thus, like IBM methods in continuum flow simulations, our method explicitly imposes the correct boundary con-

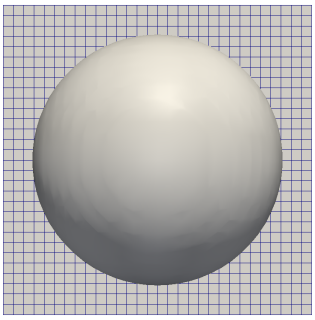

Figure 1: An immersed sphere in a fixed grid (clipped).

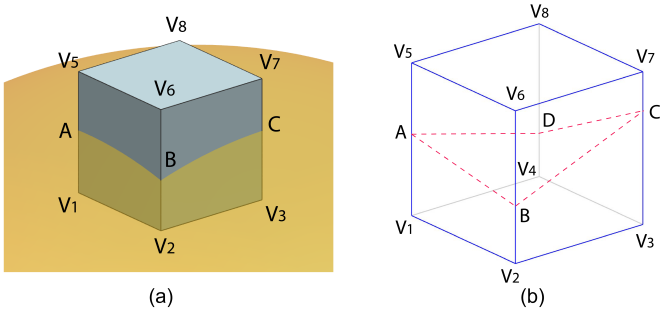

Figure 2: (a) A grid cell intersected by an immersed sphere and (b) polyhedra generated using intersecting points. The latter is only used for effective volume calculation in determining particle-particle collisions, not for surface reconstruction in calculating particle-surface interactions.

ditions exactly at the instantaneous location of the analytically expressed moving surface. On the other hand, our method is limited to geometries of which the surface can be expressed by analytical expressions. Also, the current implementation of our method does not take into account the possibility of the occurrence of split cells, i.e. cells which are divided into multiple independent regions by the immersed object. Rather, we only focus on objects of which the local curvature is larger than the local cell size.

2. Cut-cell algorithm in DSMC

For the sake of simplicity, a perfect sphere is selected as an immersed object for the illustration of the algorithm. Later we will discuss more complex body shapes.

Figure 1 shows an immersed sphere in a fixed Cartesian grid. In DSMC, there are two main steps for gas molecular dynamics: 1) the molecular streaming step and 2) the molecular collision step, which are decoupled from each other at each time step as the essential assumption of the method. For an immersed body with a pre-known and analytically expressed shape, the interaction of the particles with the immersed boundary is calculated exactly at the landing point of the particle on the analytically expressed boundary, as shown later in section 3.4. On the other hand, the intermolecular collision step requires computation of the cell volume for a correct collision probability between two simulated molecules in a cell. Equation (1) (Bird, 1994) shows this probability P and the empty volume of the occupied cell is required as the denominator:

$$P = F_N \sigma_T c_r \Delta t / V_c \tag{1}$$

Here, F_N is the number of real molecules that is represented by one simulated molecule, σ_T is the total collision cross-section, c_r is the relative velocity between the two molecules, Δt is the time step and V_c is the empty volume of the cell.

When an immersed boundary intersects the grid cells, the cut cells are partially overlapped by the immersed body and therefore the overlap volume should be subtracted from the original cell volume to provide an effective empty volume V_c and thus correct collision probability.

2.1. Overlap volume computation algorithm

Figure 2 shows a grid cell that is intersected by an immersed sphere. In this example, there are four intersected edges from the cell. Nevertheless, in other cases with different number of intersected edges the computation of overlap volume follows the same algorithm summarized below:

1. Find all the intersected edges of the cell and compute the coordinates of the intersecting points at each edge, namely A, B, C and D in Figure 2.
2. Generate polyhedra using A, B, C, D, V_1, V_2, V_3 and V_4 as vertices. Since A, B, C and D are not necessarily on

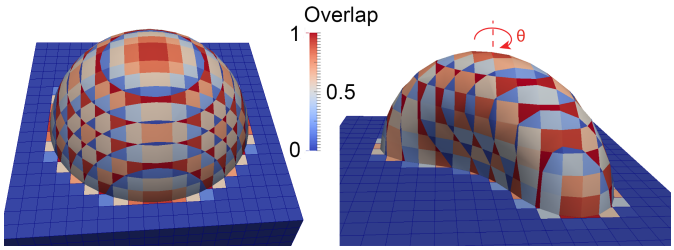

Figure 3: Immersed sphere (left) and red blood cell (right) reproduced from contours of 50% overlap. The color map shows the overlap fraction of each grid cell.

the same plane, there are more than one possible polyhedra from the configuration depending on either using $\overline{AC}$ or $\overline{BD}$ as one of the edges of the polyhedron.

3. Each of the possible polyhedra is decomposed into a number of pyramids by connecting its faces with the center of volume. Then the volume of each pyramid is computed and summed up to yield the volume of each initial polyhedron.
4. In this example, since a sphere is a convex body, the maximum volume from the above possible polyhedra could serve as the best approximation for the overlap volume. However, to be more general, we use an average over the maximum and minimum as a good approximation, as shown later in this section.

In this example, the grid resolution is defined by $N = d/\Delta x$, where d is the sphere diameter and Δx is the cell size. Figure 3 (left) shows an immersed sphere reproduced by the contour of 50% overlap using the average volume of the polyhedra at $N = 16$, which renders an excellent spherical surface. The color map on the surface indicates how the boundary cells are intersected by the sphere surface.

Since it is the effective volume of the boundary cells that directly contributes to the molecular collision probability, the relative error in the reconstructed sphere volume is defined as,

$$\epsilon_{rel} = \frac{V_{IB} - \sum\limits_{all\ cells} V_{overlap}}{\sum\limits_{boundary\ cells} V_{overlap}} \qquad (2)$$

Here, V_{IB} is the actual volume of the immersed body, $V_{overlap}$ is the calculated overlap volume of each cell including the ones that are completely enclosed by the sphere and $V_{overlap,\ boundary}$ is the overlap volume of a partially overlapped boundary cell.

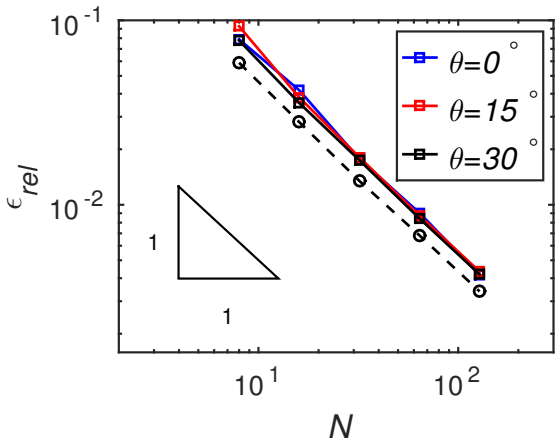

Figure 4: Relative error in the computed overlap volume against different grid resolutions for the sphere (circles) and the red blood cell (squares).

Figure 4 shows ϵ_{rel} for different grid resolutions. The relative error in the computed overlap volume decreases linearly with increasing N.

In order to illustrate that the same algorithm works for any other arbitrary shape, a red blood cell shaped body is chosen as a second example. The shape can be expressed by Equation (3) (Evans and Fung, 1972) and is plotted in Figure 5.

$$z(r) = 1/2\sqrt{1-(\frac{r}{R})^2}\,(C_0 + C_2(\frac{r}{R})^2 + C_4(\frac{r}{R})^4) \qquad (3)$$

Here, R is 3.91 µm, C_0 is 0.81 µm, C_2 is 7.83 µm and C_4 is -4.31 µm. The major diameter of the red blood cell is $d = 2R$ and again the grid resolution is defined as $N = d/\Delta x$.

Figure 3 (right) shows a red blood cell reproduced by the contour of 50% overlap at $N = 16$, which agrees well with the shape shown in Figure 5. ϵ_{rel} is again defined by Equation (2). To study the sensitivity of the method to the alignment of the body with the grid, the immersed red blood cell is tilted at different angles θ with respect to the background grid, as shown in Figure 3 (right). Figure 4 shows ϵ_{rel} against different grid resolutions. The results for $\theta = 0°, 15°$ and $30°$ are very close to each other and they all converge linearly in N. Since the resolution is defined based on the major diameter d, the mi-

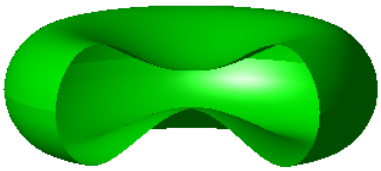

Figure 5: Red blood cell shape expressed by Equation (3) (a quarter is clipped out).

nor axis is relatively poorly resolved, which explains the higher relative errors compared to that of the sphere.

Thus it is concluded that the presented algorithm works for any arbitrarily shaped immersed objects and ϵ_{rel} converges with order 1 for the grid refinement N.

3. Validation with drag computation

The above overlap volume calculation algorithm has been implemented into an open source DSMC solver "dsmcFoam" in OpenFOAM (Scanlon et al., 2010) and the drag exerted on an immersed sphere in a creeping flow is computed for rarefied gas flows with Knudsen number larger than 0.2. The result is then validated by comparing with the drag computed from the conventional DSMC with a body-fitted grid, and with the analytical approximations.

3.1. Computational setup

For the following simulations, a sphere with diameter d is located at the center of a $(10d)^3$ cubic domain with free stream boundary conditions at two opposite planes, and periodic boundary conditions at the rest of boundaries. The simulated gas is argon of which the molecular properties are taken from Bird (1994) with the variable hard sphere (VHS) collision model.

The direct input parameters are:

1. sphere diameter: $d = 2R = 5.0 \cdot 10^{-7}$ m
2. grid size: $\Delta x = \frac{d}{N}$, $8 \leq N \leq 19$
3. domain size: $L = 10 \times d = 5.0 \cdot 10^{-6}$ µm
4. number of cells: $N_{cells} = 10^3 \cdot N^3$
5. number of DSMC particles: $N_{particles} > 10 \cdot N_{cells}$
6. temperature: $T = 300$ K
7. pressure: $P = 3.13 \cdot 10^3 \sim 5.0 \cdot 10^4$ Pa
8. reference viscosity: $\mu_{ref} = 2.12 \cdot 10^{-5}$ kg m^{-1} s^{-1} at $T_{ref} = 273.15$ K
9. viscosity index: $\omega = 0.81$
10. free stream gas velocity: $u = 40 \sim 1.25$ m s^{-1}

The derived parameters are:

1. mean thermal velocity: $\overline{c_m} = \sqrt{\frac{8k_bT}{\pi m}} = 398.75$ m s^{-1}
2. mean free path: $\lambda = \frac{2\mu}{\overline{c_m}\rho} = \frac{2\mu k_bT}{\overline{c_m}m} \cdot \frac{1}{P} = 1.43 \cdot 10^{-7} \sim 2.29 \cdot 10^{-6}$ m
3. viscosity: $\mu = \mu_{ref}(\frac{T}{T_{ref}})^{\omega} = 2.29 \cdot 10^{-5}$ kg m^{-1} s^{-1}
4. Knudsen number: $Kn = \frac{\lambda}{R} = \frac{2\mu k_bT}{\overline{c_m}mR} \cdot \frac{1}{P} = 0.28 \sim 9.18$
5. Reynolds number: $Re = \frac{u\rho R}{\mu} = \frac{Rm}{\mu k_bT} \cdot (u \cdot P) = 0.022$

where k_b is the Boltzmann constant, m is the gas molecular mass and ρ is the gas density. The mean free path is calculated in the same way as Phillips (1975) in order to keep consistency. In all the simulations we fulfill the common DSMC criteria, i.e. time step $\Delta t < \frac{1}{8}\lambda/\overline{c_m}$, $\Delta x < \frac{1}{3}\lambda$ and each cell contains on average at least 10 DSMC molecules to ensure a well resolved simulation (Garcia and Wagner, 2000; Sun et al., 2011). The gas in the domain at time $t = 0$ is initialized with the mass-averaged velocity of u.

A fully diffusive boundary condition has been employed at the sphere surface as it is the most commonly used boundary condition in literature. The drag force is directly computed at each time step from the momentum difference of the reflected molecules before and after the reflection. The drag force F_d exerted on the sphere is normalized by the Stokes drag as

$$F_d^{\star} = \frac{F_d}{F_{Stokes}} = \frac{F_d}{6\pi\mu Ru} \tag{4}$$

We verified that the calculated drag values at consecutive time steps in our simulations are nearly uncorrelated, with the autocorrelation function dropping to values below 10^{-2} at $t = \Delta t$. We took the viscous diffusion time $t_\mu = \frac{(\frac{L}{2})^2\rho}{\mu}$ as an estimate for the time needed for the flow to reach quasi-steady state. And the drag is computed by averaging over $10,000$ time steps, starting from $t = 4 \times t_\mu$.

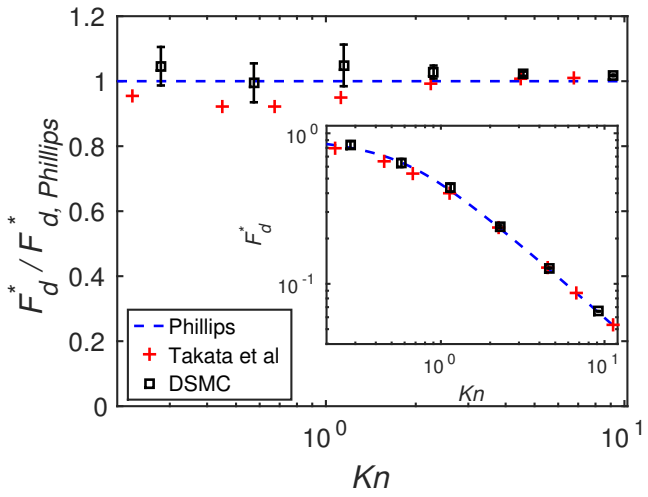

Figure 6: Drag force on a sphere as a function of Knudsen number, for the cut-cell DSMC simulations compared to analytical approximations by Phillips (1975) and Takata et al. (1993). The DSMC data with $Kn = 0.28$ is averaged over $400,000$ time steps; $Kn = 0.57$ is averaged over $50,000$ time steps; while the rest are averaged over $10,000$ time steps. Main panel: Forces are normalized by the corresponding analytical approximations from Phillips (1975), plotted on a log-lin scale. Inset: Forces are normalized by the Stokes drag and plotted on a log-log scale.

3.2. Kn dependence

First, Kn is varied by changing the pressure of the gas while keeping Re constant by changing the free stream velocity accordingly.

$F_d^\star$ values for a sphere simulated with the cut-cell method at different Kn are shown in Figure 6, compared with the analytical approximations from Phillips (1975) and Takata et al. (1993). All the error bars in the figures of this paper indicate 95% confidence intervals based on the standard deviation of mean. The drag force predicted by our DSMC simulation agrees well with the analytical results for all the shown Knudsen numbers.

3.3. Comparison with body-fitted mesh DSMC

In this section, $F_d^\star$ calculated from the presented cut-cell method is compared to that from DSMC with a conventional body-fitted mesh. A typical body-fitted grid is shown in Figure 7. The sphere surface is meshed with triangular elements which are inflated two layers outwards. The rest of flow domain

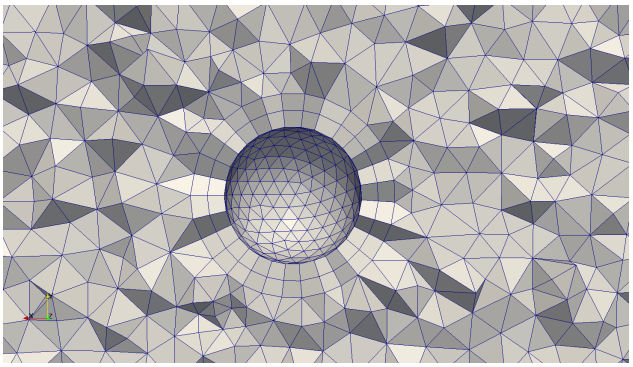

Figure 7: Body-fitted mesh for conventional DSMC

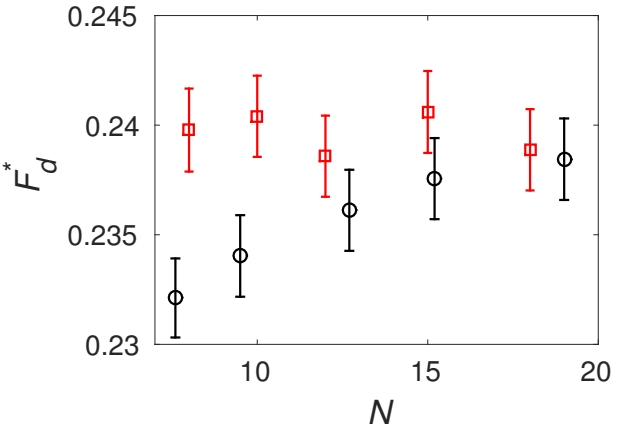

Figure 8: Normalized drag force computed from DSMC with the cut-cell method (red squares) and from the body-fitted grid (black circles), for $Kn = 2.29$.

is meshed with tetrahedron cells. For the body-fitted mesh we define

$$N = \frac{d}{\sqrt{A_{tet,surface}}} \tag{5}$$

where $A_{tet,surface}$ is the area of the surface triangular elements.

A comparison between the body-fitted mesh results and the cut-cell method results at $Kn = 2.29$ is shown in Figure 8. Here, the total number of simulated molecules is kept the same for different N, in order to render similar confidence intervals. $F_d^\star$ values from the cut-cell method show less grid dependency than that from the body-fitted grid, probably due to the utilization of the exact analytic surface for the bounce back of the molecules.

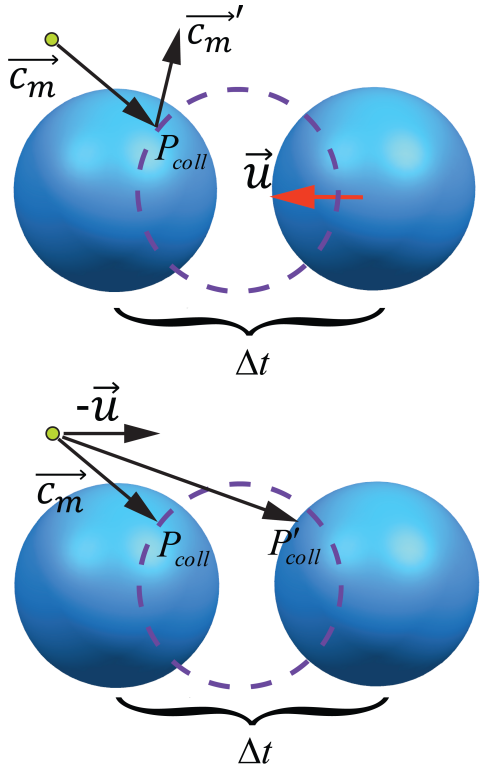

Figure 9: Transfer of the reference frame between the sphere and the flow domain.

3.4. Stationary and moving sphere with immersed boundaries

In this section, the presented cut-cell method is demonstrated with a fully three-dimensional moving immersed object. A moving sphere with a predefined velocity $\vec{u} = -u \cdot \vec{e_x}$ is simulated with the cut-cell method in a stationary gas, where $\vec{e_x}$ is the unit vector in the streamwise direction. The sphere velocity is constant during the simulation, assuming that the sphere mass is infinitely large compared to that of the gas molecule. The computed drag $F_d^\star$ is compared with that from a steady sphere in a moving flow with velocity $+u \cdot \vec{e_x}$.

The computational domain is elongated in the stream-wise direction by $u \cdot t_{total}$ to allow the movement of the sphere, where t_{total} is the total simulation time. The sphere moves over a distance $-u \cdot \Delta t$ at each time step and the incoming molecules need to be reflected at the exact, analytically calculated, landing point P_{coll} as shown in Figure 9. This is achieved by switching back and forth between the steady observer reference frame and the reference frame of the moving sphere. First, the molecule streaming is calculated in the moving sphere reference frame by adding a velocity $-u$ to each molecule's thermal velocity $\vec{c_m}$. Since the sphere is steady in its own reference frame, the calculation procedure of the post reflection molecular velocities and positions is exactly the same as that for a steady sphere. Subsequently, the velocities and positions of the molecules are transferred back to the steady observer frame of reference by taking into account the movement of sphere during Δt.

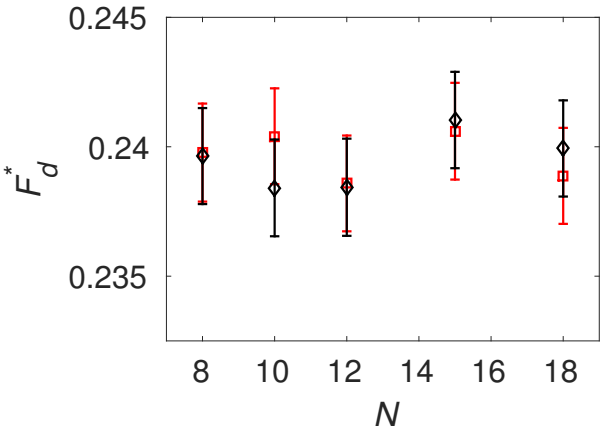

Figure 10: Normalized drag force computed from DSMC with the cut-cell method for a steady sphere (red squares) and for a moving sphere (black diamonds), for $Kn = 2.29$.

The simulations are conducted at $Kn = 2.29$. Figure 10 shows the comparison of the scaled drag force $F_d^\star$ between the moving sphere and the steady sphere, both calculated with the cut-cell method. The two results agree well with each other, illustrating that this method works for a fully three-dimensional immersed moving body.

4. Conclusion

A new cut-cell algorithm has been implemented and demonstrated in conjunction with DSMC for simulating rarefied gas flow around an immersed moving body. Based on the presented study, the following conclusions are drawn:

1. An analytically expressed surface is used to accurately bounce back the molecules. The cut cell effective volume is computed by representing the immersed boundary with

the Lagrangian intersecting points and thus reconstructing all the possible polyhedra and take the mean average. It has been shown that for any arbitrary immersed bodies, the relative error in the computed overlap volume decreases linearly with the grid refinement.

2. The drag force on a sphere computed with the present cut-cell method converges to the same value as that calculated with a conventional body-fitted grid, both in good agreement with approximate analytical solutions. However, with the cut-cell method grid-independent results are obtained for coarser grids as compared to the body-fitted grid method.
3. The drag on a moving sphere in a stagnant gas computed with the cut-cell method agrees well with that of a steady sphere in a flowing gas. Thus it is shown that the present method can apply to fully three-dimensional immersed moving bodies in a rarefied gas flow.

5. Acknowledgment

This work is supported by NanoNextNL, a micro and nanotechnology consortium of the Government of the Netherlands and 130 partners.

References

Akhlaghi, H., Roohi, E., Stefanov, S., 2012. A new iterative wall heat flux specifying technique in DSMC for heating/cooling simulations of MEMS/NEMS. International Journal of Thermal Sciences 59, 111–125. URL: http://dx.doi.org/10.1016/j.ijthermalsci.2012.04.002, doi:10.1016/j.ijthermalsci.2012.04.002.

Bird, G.A., 1994. Molecular gas dynamics and the direct simulation of gas flows. Hardcover. URL: http://www.worldcat.org/isbn/0198561954.

Burt, J.M., Josyula, E., Boyd, I.D., 2012. Novel Cartesian Implementation of the Direct Simulation Monte Carlo Method. Journal of Thermophysics and Heat Transfer 26, 258–270. URL: http://dx.doi.org/10.2514/1.t3733, doi:10.2514/1.t3733.

Chen, S., Doolen, G.D., 1998. Lattice boltzmann method for fluid flows. Annual Review of Fluid Mechanics 30, 329–364. URL: http://dx.doi.org/10.1146/annurev.fluid.30.1.329, doi:10.1146/annurev.fluid.30.1.329.

Evans, E., Fung, Y.C., 1972. Improved measurements of the erythrocyte geometry. Microvascular Research 4, 335–347. URL: http://dx.doi.org/10.1016/0026-2862(72)90069-6, doi:10.1016/0026-2862(72)90069-6.

Frenkel, D., Smit, B., 2002. Understanding molecular simulation : from algorithms to applications. URL: http://www.worldcat.org/isbn/9780122673511.

Garcia, A.L., Wagner, W., 2000. Time step truncation error in direct simulation Monte Carlo. Physics of Fluids (1994-present) 12, 2621–2633. URL: http://dx.doi.org/10.1063/1.1289691, doi:10.1063/1.1289691.

Gompper, G., Ihle, T., Kroll, D.M., Winkler, R.G., 2009. Multi-Particle Collision Dynamics: A Particle-Based Mesoscale Simulation Approach to the Hydrodynamics of Complex Fluids, in: Holm, C., Kremer, K. (Eds.), Advanced Computer Simulation Approaches for Soft Matter Sciences III. Springer Berlin Heidelberg, Berlin, Heidelberg. volume 221 of *Advances in Polymer Science*. chapter 1, pp. 1–87. URL: http://dx.doi.org/10.1007/978-3-540-87706-6_1, doi:10.1007/978-3-540-87706-6_1.

Hong, Z.C., Zhen, C.E., Yang, C.Y., 2008. Fluid Dynamics and Heat Transfer Analysis of Three Dimensional Microchannel Flows with Microstructures. Numerical Heat Transfer, Part A: Applications 54, 293–314. URL: http://dx.doi.org/10.1080/10407780701790128, doi:10.1080/10407780701790128.

Hoogerbrugge, P.J., Koelman, J.M.V.A., 1992. Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dynamics. Europhysics Letters (EPL) 19, 155–160. URL: http://dx.doi.org/10.1209/0295-5075/19/3/001, doi:10.1209/0295-5075/19/3/001.

Lo, M.C., Su, C.C., Wu, J.S., Kuo, F.A., 2014. Development of parallel direct simulation Monte Carlo method using a cut-cell Cartesian grid on a single graphics processor. Computers & Fluids 101, 114–125. URL: http://dx.doi.org/10.1016/j.compfluid.2014.06.003, doi:10.1016/j.compfluid.2014.06.003.

Oran, E.S., Oh, C.K., Cybyk, B.Z., 1998. DIRECT SIMULATION MONTE CARLO: Recent Advances and Applications. Annual Review of Fluid Mechanics 30, 403–441. URL: http://dx.doi.org/10.1146/annurev.fluid.30.1.403, doi:10.1146/annurev.fluid.30.1.403.

Peskin, C.S., 2002. The immersed boundary method. Acta Numerica 11, 479–517. URL: http://dx.doi.org/10.1017/s0962492902000077, doi:10.1017/s0962492902000077.

Phillips, W.F., 1975. Drag on a small sphere moving through a gas. Physics of Fluids (1958-1988) 18, 1089–1093. URL: http://dx.doi.org/10.1063/1.861292, doi:10.1063/1.861292.

Rader, D.J., Gallis, M.A., Torczynski, J.R., 2011. DSMC Moving-Boundary Algorithms for Simulating MEMS Geometries with Opening and Closing Gaps, in: American Institute of Physics Conference Series, AIP. pp. 760–765. URL: http://dx.doi.org/10.1063/1.3562738, doi:10.1063/1.3562738.

Scanlon, T.J., Roohi, E., White, C., Darbandi, M., Reese, J.M., 2010. An open source, parallel DSMC code for rarefied gas flows in arbitrary geometries.

Computers & Fluids 39, 2078–2089. URL: http://dx.doi.org/10.1016/j.compfluid.2010.07.014, doi:10.1016/j.compfluid.2010.07.014.

Shrestha, S., Tiwari, S., Klar, A., Hardt, S., 2015. Numerical simulation of a moving rigid body in a rarefied gas. Journal of Computational Physics 292, 239–252. URL: http://dx.doi.org/10.1016/j.jcp.2015.03.030, doi:10.1016/j.jcp.2015.03.030.

Sun, Z.X., Tang, Z., He, Y.L., Tao, W.Q., 2011. Proper cell dimension and number of particles per cell for DSMC. Computers & Fluids 50, 1–9. URL: http://dx.doi.org/10.1016/j.compfluid.2011.04.013, doi:10.1016/j.compfluid.2011.04.013.

Takata, S., Sone, Y., Aoki, K., 1993. Numerical analysis of a uniform flow of a rarefied gas past a sphere on the basis of the Boltzmann equation for hardsphere molecules. Physics of Fluids A: Fluid Dynamics (1989-1993) 5, 716–737. URL: http://dx.doi.org/10.1063/1.858655, doi:10.1063/1.858655.

Versluis, R., Dorsman, R., Thielen, L., Roos, M.E., 2009. Numerical investigation of turbomolecular pumps using the direct simulation Monte Carlo method with moving surfaces. Journal of Vacuum Science & Technology A 27, 543–547. URL: http://dx.doi.org/10.1116/1.3119668, doi:10.1116/1.3119668.

Zabelok, S., Arslanbekov, R., Kolobov, V., 2015. Adaptive kinetic-fluid solvers for heterogeneous computing architectures. Journal of Computational Physics 303, 455–469. URL: http://dx.doi.org/10.1016/j.jcp.2015.10.003, doi:10.1016/j.jcp.2015.10.003.

Zhang, C., Schwartzentruber, T.E., 2012. Robust cut-cell algorithms for DSMC implementations employing multi-level Cartesian grids. Computers & Fluids 69, 122–135. URL: http://dx.doi.org/10.1016/j.compfluid.2012.08.013, doi:10.1016/j.compfluid.2012.08.013.