Computers & Fluids 48 (2011) 113–124

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Two-level virtual mesh refinement algorithm in a parallelized DSMC Code using unstructured grids

C.C. Su [a], K.C. Tseng [b], J.S. Wu [a,*], H.M. Cave [a,c], M.C. Jermy [c], Y.Y. Lian [b]

[a] Department of Mechanical Engineering, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu, Taiwan
[b] National Space Organization, 8F, 9 Zhan-Ye 1st Road, Hsinchu Science Park, Hsinchu, Taiwan
[c] Department of Mechanical Engineering, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand

ARTICLE INFO

Article history:
Received 27 April 2010
Received in revised form 16 January 2011
Accepted 6 April 2011
Available online 19 April 2011

Keywords:
Virtual mesh refinement
Parallel direct simulation Monte Carlo
Unstructured grid

ABSTRACT

In this paper, a virtual mesh refinement (VMR) algorithm for unstructured grids is developed and verified in the parallelized direct simulation Monte Carlo code (PDSC). This algorithm is a two-level virtual mesh refinement, in which the background mesh is refined based on an initial DSMC simulation. The refined cells are arranged in a way similar to the structured grid, which makes the particle tracing on them very efficient, unlike unstructured grids. These virtual cells are used for particle collision and sampling. Only those refined cells which contain the centroid of background cells are used for outputting macroscopic data. Two hypersonic flows with purely quadrilateral and mixed triangular–quadrilateral mesh were used to verify this new mesh-refining algorithm. Results show that the case using VMR can faithfully reproduce the benchmark case with a much reduced computational time.

1. Introduction

The direct simulation Monte Carlo (DSMC) method is a computational tool for simulating flows in which effects at the molecular scale become significant [1]. The Boltzmann equation, which is appropriate for modeling these rarefied flows, is extremely difficult to solve numerically due to its high dimensionality and the complexity of the collision term. DSMC provides a particle based alternative for obtaining realistic numerical solutions. In DSMC the movement and collision behavior of a large number of representative particles within the flow field are decoupled over a time step which is a small fraction of the local mean collision time. The computational domain itself is divided into either a structured or unstructured grid of cells which are then used to select particles for collisions on a probabilistic basis and also are used for sampling the macroscopic flow properties. The method has been shown to provide a solution to the Boltzmann equation as the number of simulated particles tends toward the true value within the flow field [2]. The DSMC cells are used for particle collisions and sampling of macroscopic properties, in which the cell sizes must be much smaller than the local mean free path for a meaningful simulation. Unfortunately, it is generally impossible to know the distribution of local mean free path a priori. To obtain a better spatial and physical resolution, an appropriate solution-based mesh-refining approach is strongly needed.

In the past, several mesh-refining strategies have been developed in the DSMC community. For Cartesian structured grids, a two-level mesh-refining approach has often been adopted [1,3]. One of the major advantages of this mesh-refining approach is the fast particle tracking on the background mesh. However, the mesh-refining process may become awkward along the complex geometry of objects. For unstructured grids, an isotropic mesh-refining method based on h-refinement concept has been proposed [4,5]. One of the major advantages of this approach is the capability in body-fitting any complicated geometry of objects. However, there exist several major disadvantages, which include: (1) difficulty in maintaining mesh quality even with complicated mesh quality control algorithms, (2) very complicated hanging-node removal algorithms, especially for three-dimensional cases [4] and (3) inefficient particle tracking on the refined cells. Thus, an alternative algorithm of mesh refinement on unstructured grids, which is free of the above problems, is critical in applying unstructured grids in the parallel DSMC method [6].

Recently, Su et al. [7] had proposed a transient adaptive sub-cell (TAS) approach in DSMC utilizing unstructured grids to ensure better collision quality. In this paper, a new mesh-refining process for DSMC on unstructured grids is proposed, largely based on the TAS concept, and used to virtually refine the background cells. It is named two-level virtual mesh refinement (VMR). The paper is organized as follows. Detailed procedures of the two-level VMR are introduced, followed by several verification tests using different kinds of mesh. A hypersonic M-15 scramjet flow is then used to demonstrate its capability in practical flow simulation. Finally,

* Corresponding author. Tel.: +886 3 573 1693; fax: +886 3 611 0023.
E-mail address: chongsin@faculty.nctu.edu.tw (J.S. Wu).

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.04.002

conclusions are summarized and possible future directions for research are outlined.

2. Numerical method

2.1. Parallel implementation of DSMC

In general, the DSMC procedure involves: (1) moving the particles collisionlessly over a small time step and applying boundary conditions to particles which collide with boundaries, (2) indexing the particles within the grid of collision cells, (3) selecting particles from within the cells on a probabilistic basis and applying the collision routines to these, and (4) sampling the macroscopic flow properties from the collision cells. The cells are used for collision and sampling. To maintain a good quality of collisions, the cell size should be 1/2–1/3 of local mean free path. This is difficult to guarantee in practice, since the solution is generally not known a priori. In this section, we introduce a new mesh-refining procedure for the DSMC method which utilizes unstructured grids. This procedure is termed two-level virtual mesh refinement (VMR). It is termed "two-level" because the background grid is considered the first level while the refined grids are the second level.

Fig. 1 shows the temporal evolution of the particle number in DSMC with the VMR module, which is described next. The steps include: (1) The initial DSMC simulation on the background grids, (2) virtual mesh refinement based on the data obtained in Step (1), (3) adjusting the time step size and particle weighting in the refined cells accordingly, (4) generating and randomly distributing particles in the refined cells based on Maxwellian distribution of velocities, and (5) final DSMC simulation on the refined grids. Note the TAS function is used throughout the whole procedures to ensure good collision quality. Some of the details in the above procedures are described in the following.

In Step (1), the initial DSMC simulation is conducted on the background cells. The area of background cells is calculated with the simple coordinate geometry technique. In this step, the simulation is the same as usual parallel DSMC simulation.

In Step (2), the results of the initial DSMC simulation are used to determine the local mean free path in each background cell, which is then compared with the corresponding cell size. The result of this comparison is then used to calculate the number of refined

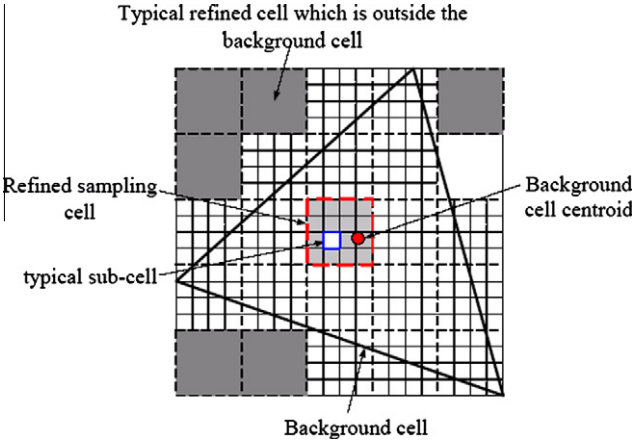

Fig. 2. Typical refined cells (dashed lines) on a triangular background cell (solid lines) along with TAS.

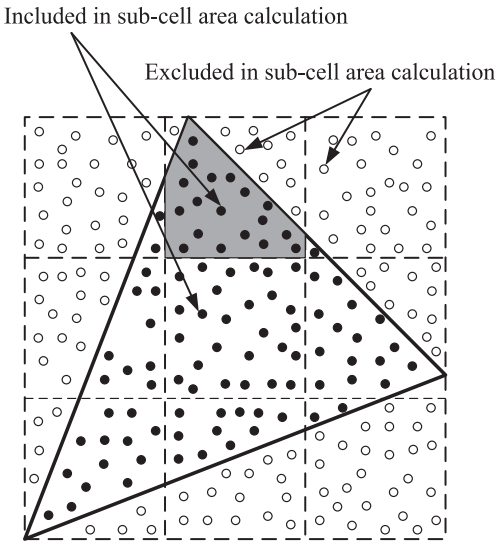

Fig. 3. Schematic of the calculation of the area of overlap between a refined cell and an unstructured background cell using the Monte Carlo method.

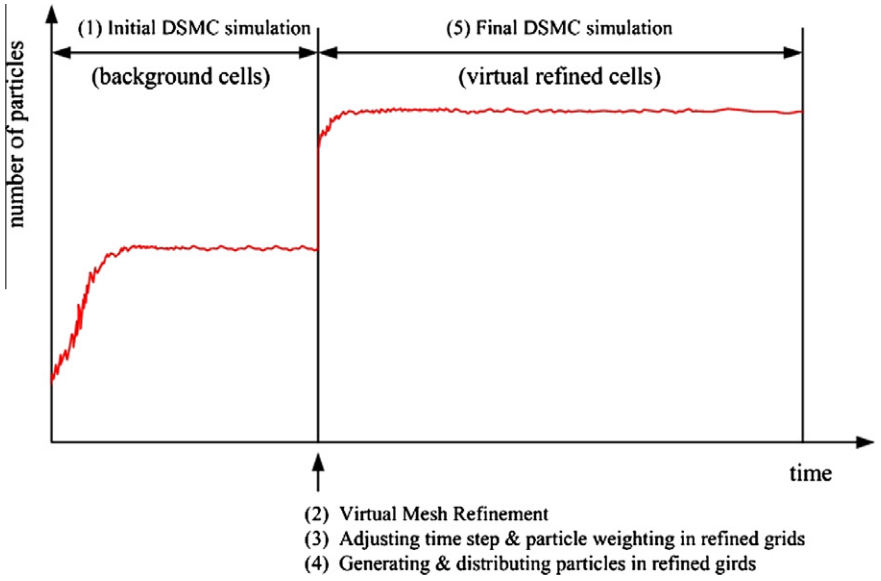

Fig. 1. Temporal evolution of the proposed DSMC procedure with the VMR module.

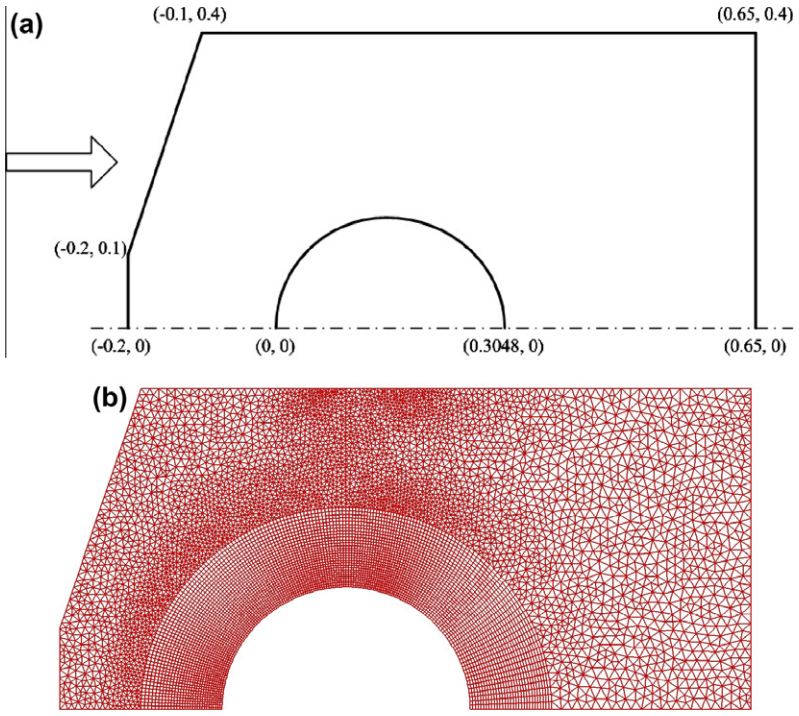

Fig. 4. Mach-10 hypersonic flow past a circular cylinder. (a) Schematic diagram showing locations in normalized units. (b) Mesh of mixed quadrilateral and triangular cells (12,825 cells).

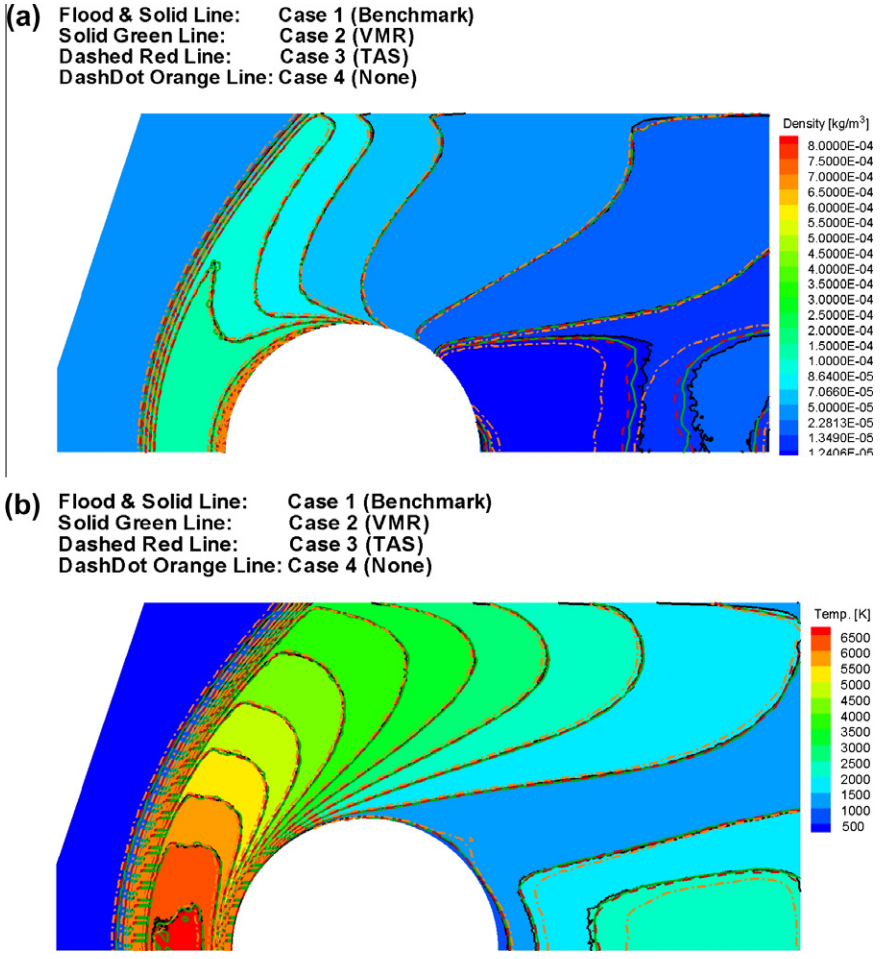

Fig. 5. Contours of properties of Mach-10 hypersonic flow past a circular cylinder. (a) Density. (b) Temperature.

cells in each coordinate direction required to resolve the local mean free path in background cell. Note the refined cells are laid out on a Cartesian grid with the same cell size in square shape. The size of the refined cell is normally less than one half of the local mean free path, although it can be controlled by the user. A typical example is schematically shown in Fig. 2. One important advantage is the particle tracing becomes very efficient, due to the use of a Cartesian grid for the refined cells. The refined cell in each background cell, which contains the background cell centroid, is also identified in this step. This will be used in the final data output. In addition, the area of each refined cell ("area" in two-dimensional case; "volume" in three-dimensional case), which is geometrically inside the background cell, is calculated using the Monte Carlo (MC) method. Note the area of the refined cell (or volume) is required in calculating the number of collision pairs such as NTC (No Time Counter) method [1]. The conventional method based on coordinate geometry is not used as it becomes very cumbersome and complicated in the three-dimensional case. The MC method is easy in concept as well as practical implementation, as shown schematically in Fig. 3. Each particle with randomly assigned position is checked to see if it is located in the background cell. Once a particle is found to be located in the background cell, then the refined cell which contains that particle is easily identified by taking advantage of the Cartesian structure of the refined cells. Only those particles located inside the refined cell and background cell are counted for the area calculation. The area of the i^{th} refined cell inside the background cell is thus calculated as follows:

$$V_{vc_i} = V_c \times R_i \Big/ \sum_{i=1}^{N_{vc}} R_i \tag{1}$$

where R_i is the number of particles located inside the ith refined cell, V_c is the area of background cell and N_{vc} is the total number of re-

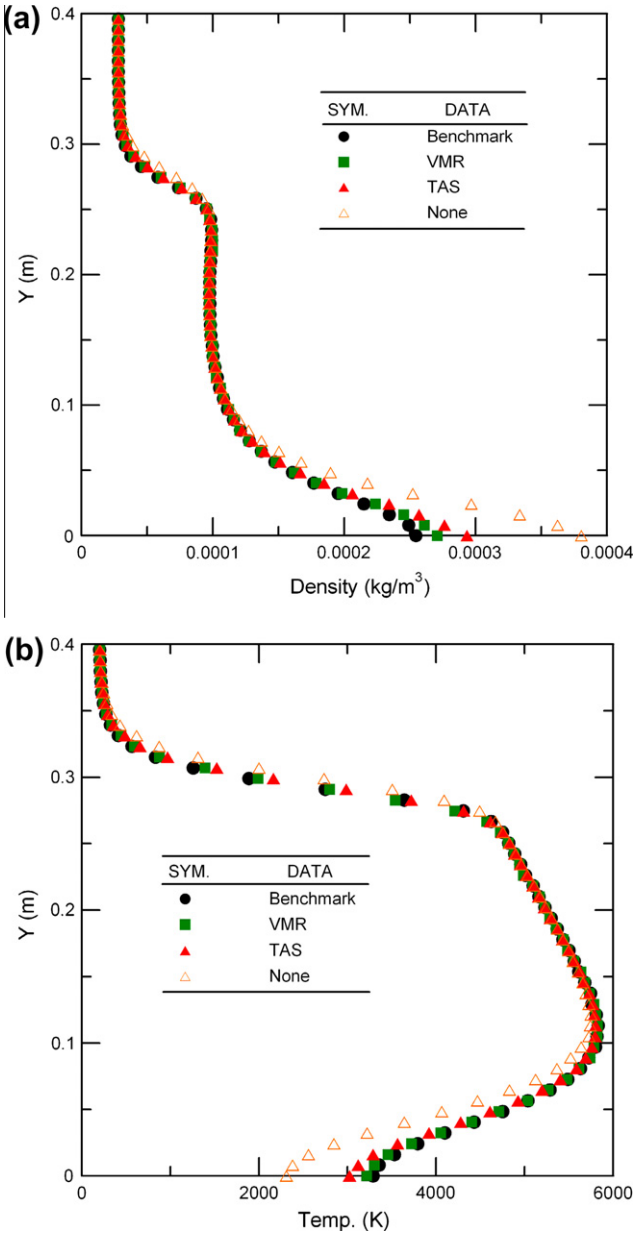

Fig. 6. Property distribution along a vertical line before the cylinder (x = 0.005 m). (a) Density. (b) Temperature.

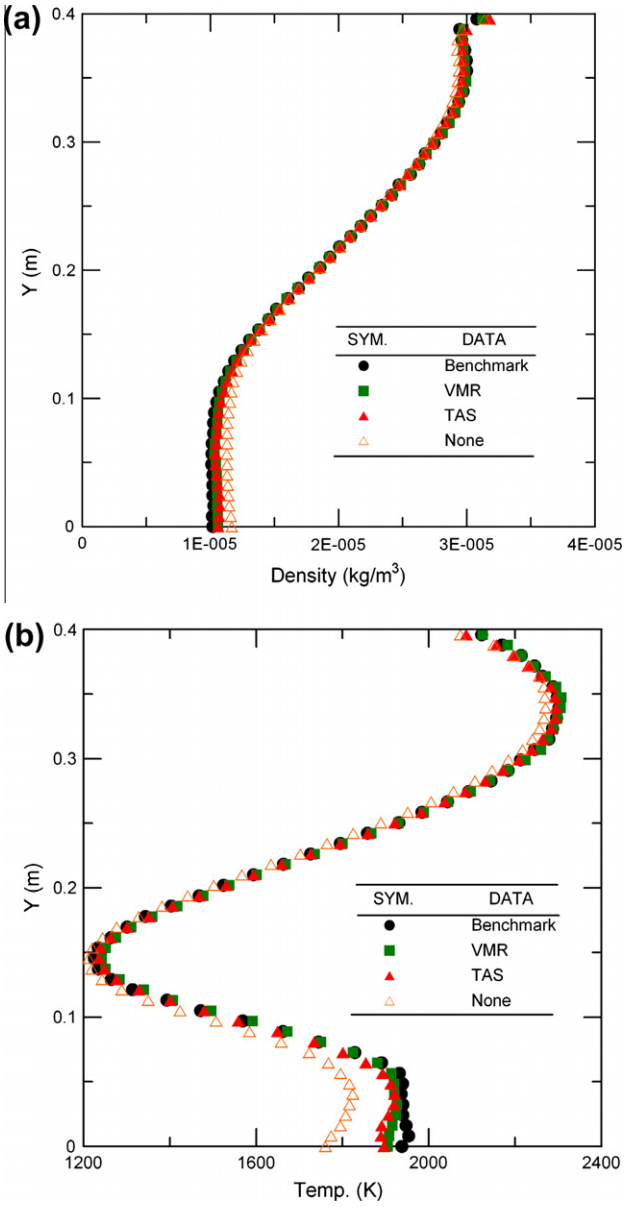

Fig. 7. Property distribution along a vertical line in the wake region (x = 0.4 m). (a) Density. (b) Temperature.

fined cells. Our experience shows that approximately $5000 * N_{vc}$ particles are required to reach 0.1% error for area calculations of all the refined cells, which takes about 12.5 min of computational time for ~300,000 virtual refined cells using 12 processors (AMD 2.4 GHz). This computational overhead is comparatively low compared to the total DSMC simulation in general. In addition, it is only necessary to do this once per simulation, if the number of refinements is limited to be one as will be the case in many steady flow problems.

In Step (3), adjustment of time step and particle weighting for refined background cell based on total number of refined cells as follows:

$$\Delta t_{adjust,i} = \Delta t_i \Big/ \sqrt{N_{vc}} \tag{2}$$

and

$$F_{adjust,i} = F_i \Big/ \sqrt{N_{vc}} \tag{3}$$

where Δt_i and F_i are time step and particle weighting for i^{th} background cell. Here the time step and particle weighting for each background cells are different to base on variable time-step scheme [4].

Adjustment of time step and particle weighting after refinement causes the increase of number of simulation particles when the flow field reaches a steady state. In Step (4), we re-generate new randomly distributed particles in the refined cells using the Maxwellian distribution based on the sampled macroscopic properties. Here the macroscopic properties are obtained from the initial DSMC simulation on background cells.

In Step (5), a final DSMC simulation is conducted based on the refined cells and unrefined background cells. The refined cells are used for collisions and sampling. And the TAS function [7] is used to identify the collision partners in each refined cell and each unrefined background cell. Finally, only the sampled data of those refined cells, which contain the centroid of the background

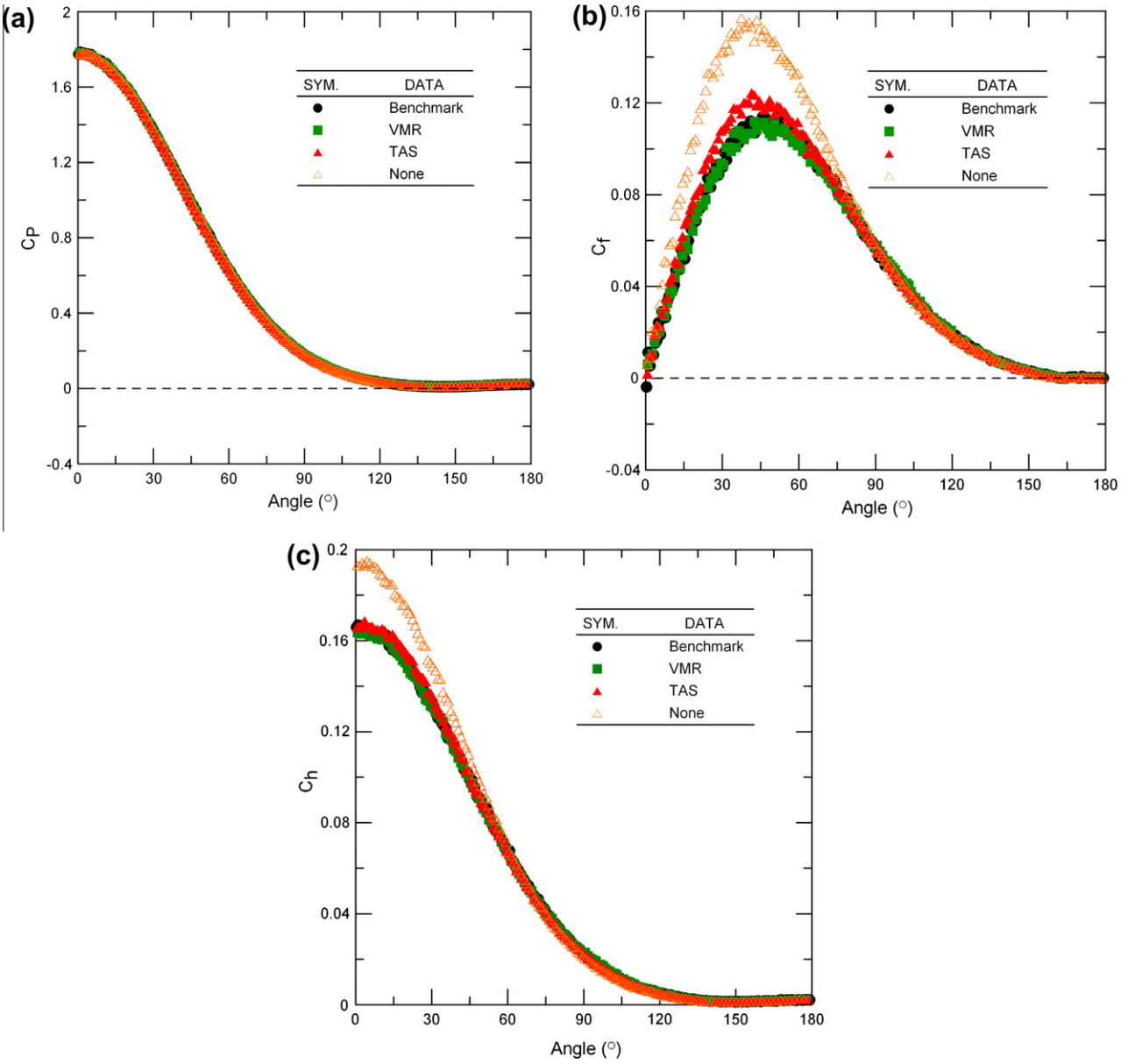

Fig. 8. Surface property distribution as a function of distance from the front stagnation point. (a) Local pressure coefficient. (b) Local friction coefficient. (c) Local heat transfer coefficient.

cells, are used as the macroscopic properties on the background cells. Indeed, after refinement we need to store all the coordinates and sampled data of refined cells for later processing before output.

3. Results and discussion

In this paper, we implement and verify the proposed VMR module in the PDSC by simulating a two-dimensional hypersonic flow over a circular cylinder ($M = 10$) using various types of grids, including purely triangular, purely quadrilateral and mixed triangular and quadrilateral. Finally, we also present the results of a hypersonic scramjet flow to demonstrate the applicability of VMR in the DSMC method. Approximately 40 particles per cell for all the cases are maintained throughout the simulations. In addition, VHS collision model [1] and variable time step (VTS) scheme [4] are applied in the simulations unless otherwise specified. Note no chemical reactions are considered in the simulations.

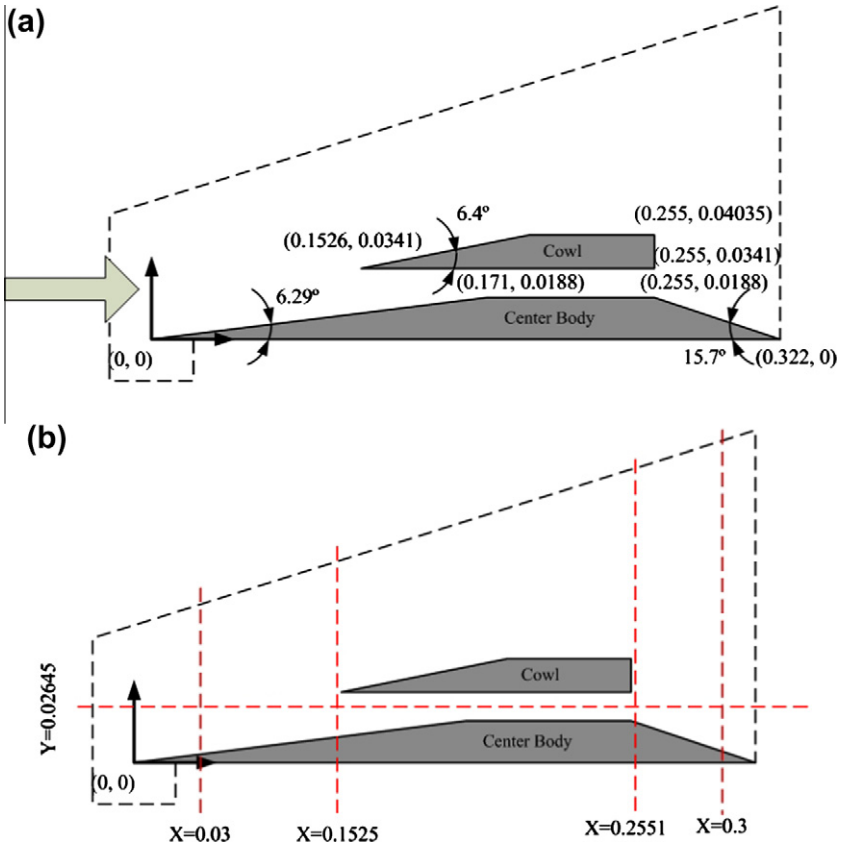

Fig. 9. Sketch of a M-15 hypersonic scramjet flow. (a) Detailed geometrical configuration. (b) Vertical lines where the flow properties are shown later.

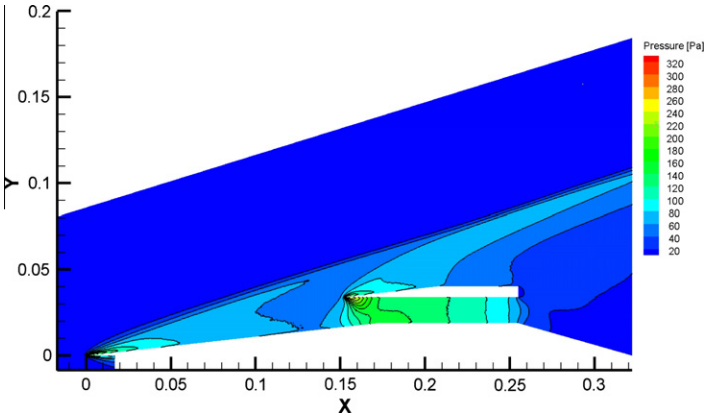

Fig. 10. Pressure contours in an M-15 scramjet flow.

3.1. M-10 hypersonic flow past a circular cylinder

Fig. 4 shows the sketch of the Mach-10 hypersonic flow past a cylinder and related mixed quadrilateral–triangular grids. Note this problem has been adopted by Bird [8] as the benchmark for DSMC recently. Important free-stream flow conditions include: argon gas, velocity of 2634.1 m/s, temperature of 200 K, number density of 4.274E20 m^{-3}, and Mach number of 10. The corresponding free-stream Knudsen number is 0.0091 based on the free-stream mean free path ($\lambda_\infty = 0.003$ m) and the diameter of the cylinder ($D = 0.3048$ m). There are four simulations, which are termed as Benchmark (only quadrilateral grids), VMR (mixed grids), TAS (mixed grids), and None (mixed grids). Note the cell size is 1/5–1/2 λ_∞ for the case of Benchmark using quadrilateral mesh, while those are only approximately 1–2 λ_∞ for the other three cases using mixed quadrilateral–triangular mesh. The number of cells of the former case is 195,000, while that of the latter cases is 12,825. Approximately 40 particles per cell are maintained for all the four simulation cases. Using 16 processors of a cluster, the computational time is ~18 h for the benchmark case, while it is reduced to 5 h for the Case-VMR, which can still faithfully reproduce the results of Case-Benchmark.

Fig. 5 shows the contours of density and temperature for all the cases. Results clearly show that the results of the cases of VMR and TAS are very close to the benchmark in general, while the results of Case-None shows large discrepancy with the Case-Benchmark, especially in the wake region. Note in the wake region the case-VMR is closer to the Case-Benchmark than the other two cases. Figs. 6 and 7 shows the distributions of density and temperature along a vertical line just before the cylinder ($x = 0.005$ m) and in the wake region ($x = 0.4$ m), respectively. In general, the results of the Case-None show large discrepancies

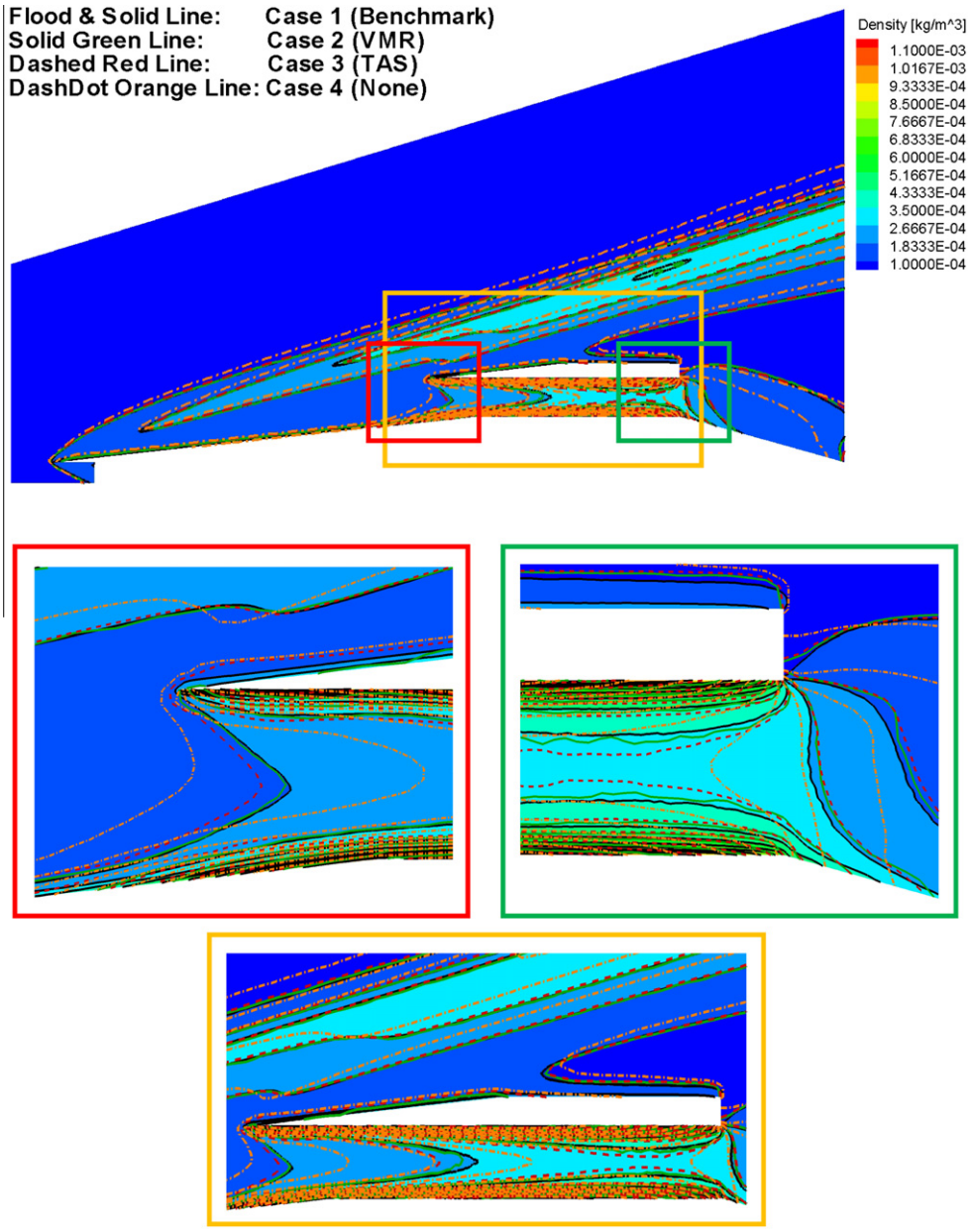

Fig. 11. Property contours in an M-15 hypersonic scramjet flow. (a) Density. (b) Temperature.

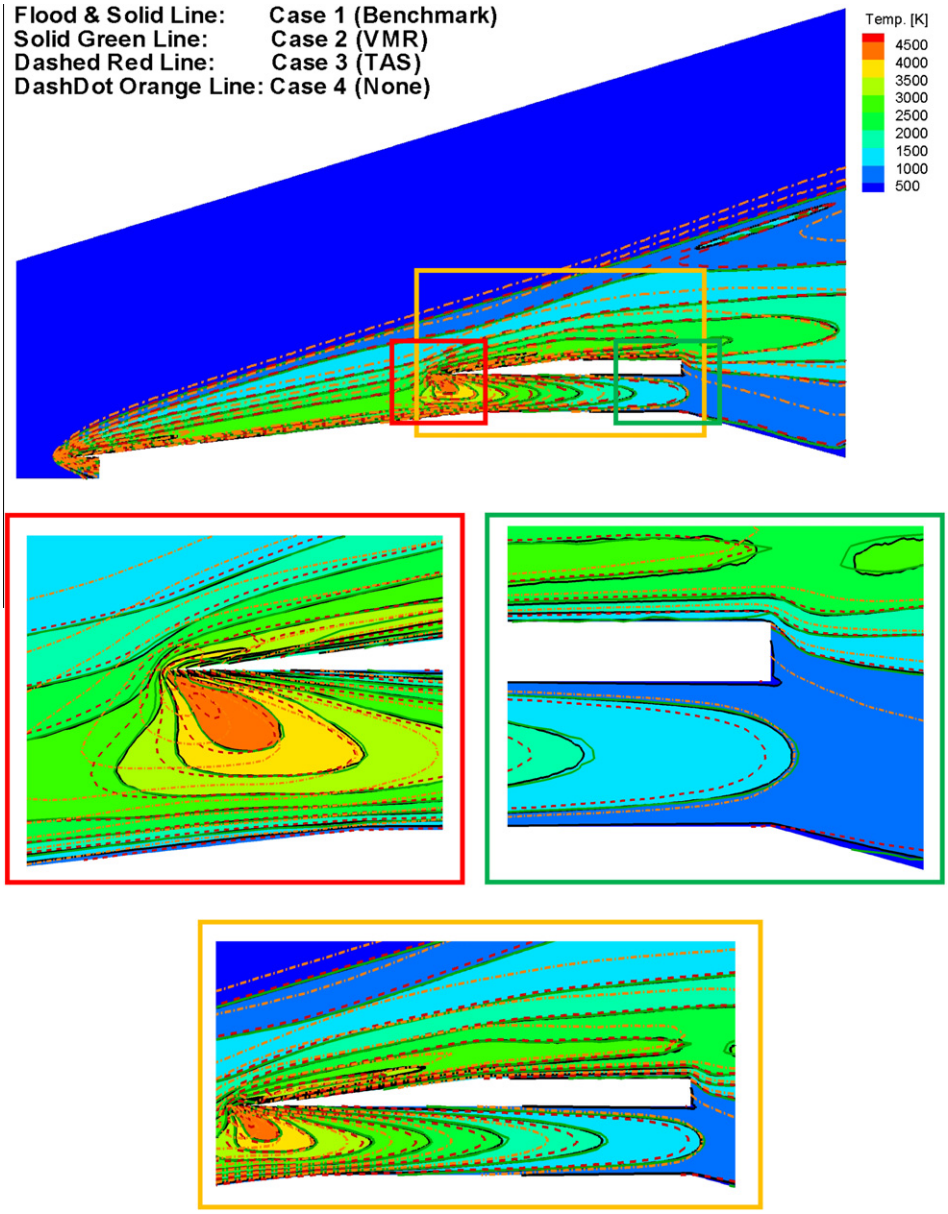

Fig. 11 (*continued*)

with the Case-Benchmark across the shock ($y = 0.25$–0.34 m), near the stagnation region and in the wake region, while those of the case-TAS show appreciable discrepancy with the Case-VMR and Case-Benchmark in the front stagnation region because of under-resolution of the high density (or small mean free path). Although the cell size in the wake region for all cases is small enough to resolve the local mean free path due to very low density there, only the results of case-VMR agree with those of the Case-Benchmark, probably because of influences mentioned above. Fig. 8 shows the surface properties as a function of angle measured from the front stagnation point. It shows that the local pressure coefficients are generally one order of magnitude larger than the local friction coefficients in most regions, except in the wake region. Results of pressure coefficient of all the test cases are in excellent agreement with those of the benchmark case. Case-None and Case-TAS clearly overpredict the peak friction coefficients, while those of Case-VMR agree very well with those of Case-Benchmark. Similarly, the heat transfer coefficients of Case-None and Case-TAS deviate significantly from those of Case-VMR and Case-Benchmark.

3.2. M-15 hypersonic scramjet flow

Fig. 9 shows the sketch of the Mach-15 hypersonic scramjet flow and the positions of the profiles for which data plotted. The free-stream flow conditions are: argon gas, velocity 3246 m/s, temperature 135 K, number density 1.41E21 m^{-3}, Mach number 15, and wall temperature of 300 K. The corresponding free-stream Knudsen number is 0.06 based on the free-stream mean free path ($\lambda_\infty = 0.000981$ m). Similarly, there are four simulations, which are termed as Benchmark, VMR, TAS, and None. Note the cell size is 1/3–1/2 λ_∞ for the case of Benchmark using a quadrilateral mesh, but only approximately 1–2 λ_∞ for the other three cases using mixed quadrilateral–triangular mesh. The number of cells

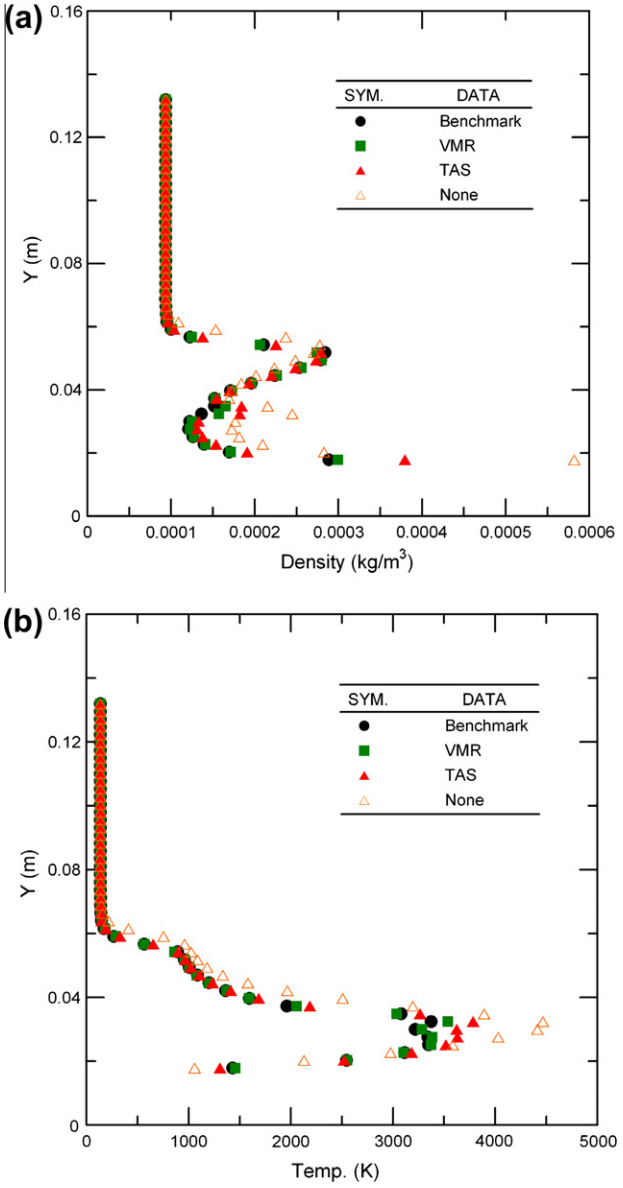

Fig. 12. Property distributions near the front end of the upper wedge along a vertical line ($x = 0.1525$ m) (a) Density. (b) Temperature.

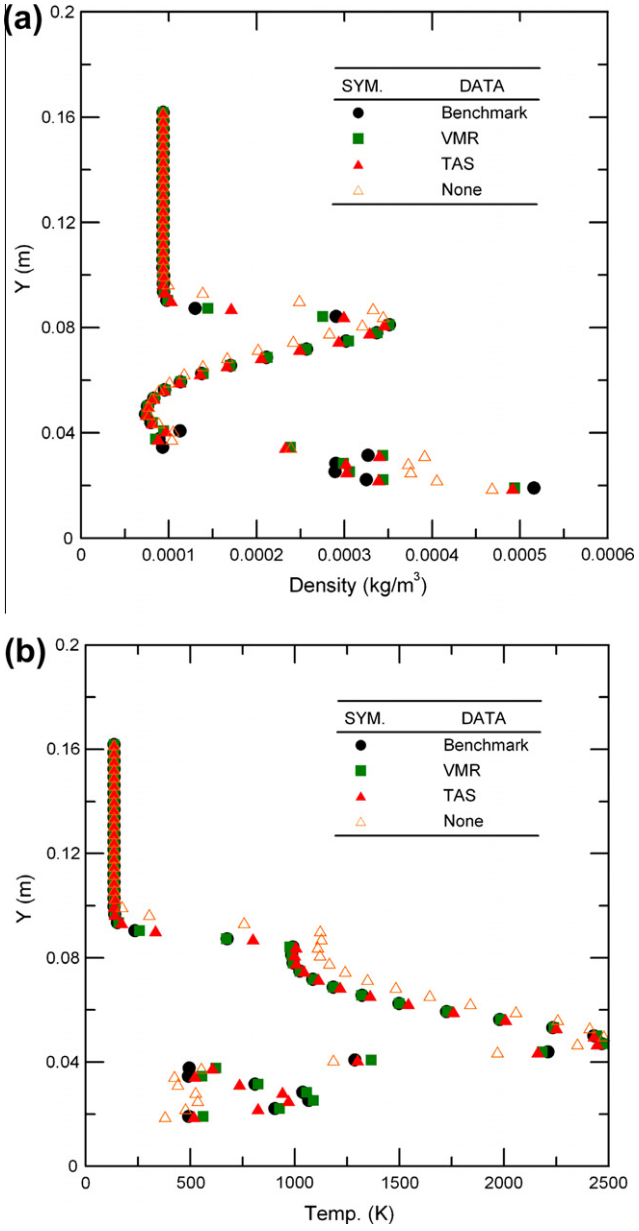

Fig. 13. Property distributions near the rear end of the upper wedge along a vertical line ($x = 0.1525$ m) (a) Density. (b) Temperature.

in the former case is 223,800, while in the latter cases it is 14,950. Approximately 35–40 particles per cell are maintained for all the four simulation cases. Using 16 processors of a cluster, the computational time is ~18 h for the benchmark case, while it is reduced to 7.5 h for the Case-VMR, yet the results of Case-Benchmark are faithfully reproduced.

Fig. 10 shows the pressure distribution around the scramjet. The flow after the first oblique shock impinges on the upper leading edge, which causes reflected shock to interact with the boundary layer flow along the bottom ramp. Finally, the flow inside the horizontal channel becomes a typical channel flow and further expands to the ambient at the exit. The flow just behind the reflected shock near the upper channel wall becomes stagnant and thus high-pressure. The above flow phenomena are reflected from Figs. 15 and 16, which show the surface properties along the upper and lower channel wall, respectively, and are described later.

Fig. 11 shows the contours of density and temperature along with corresponding exploded views in the region of horizontal channel for all test cases. They clearly show that only the results of Case-VMR agree faithfully with those of Case-Benchmark. Large deviations from Case-Benchmark can be seen in Case-None and Case-TAS, especially in the region of horizontal channel, in which the density becomes very high. This explains why Case-VMR performs much better than the other two cases without mesh refinement.

Figs. 12–14 show a series of density and temperature distributions along various vertical lines or horizontal line in the region near the horizontal channel. It is clearly seen that the results of Case-None deviate from those of Case-Benchmark significantly, while those of Case-VMR are in excellent agreement with those of Case-Benchmark. The results of Case-TAS agree reasonably well with those of Case-Benchmark, however clear discrepancy between them still exists.

Figs. 15 and 16 shows the surface property distribution along upper and lower channel wall, respectively. Results show that the pressure and shear stress along the upper channel wall are about the same order of magnitude, while the former is much

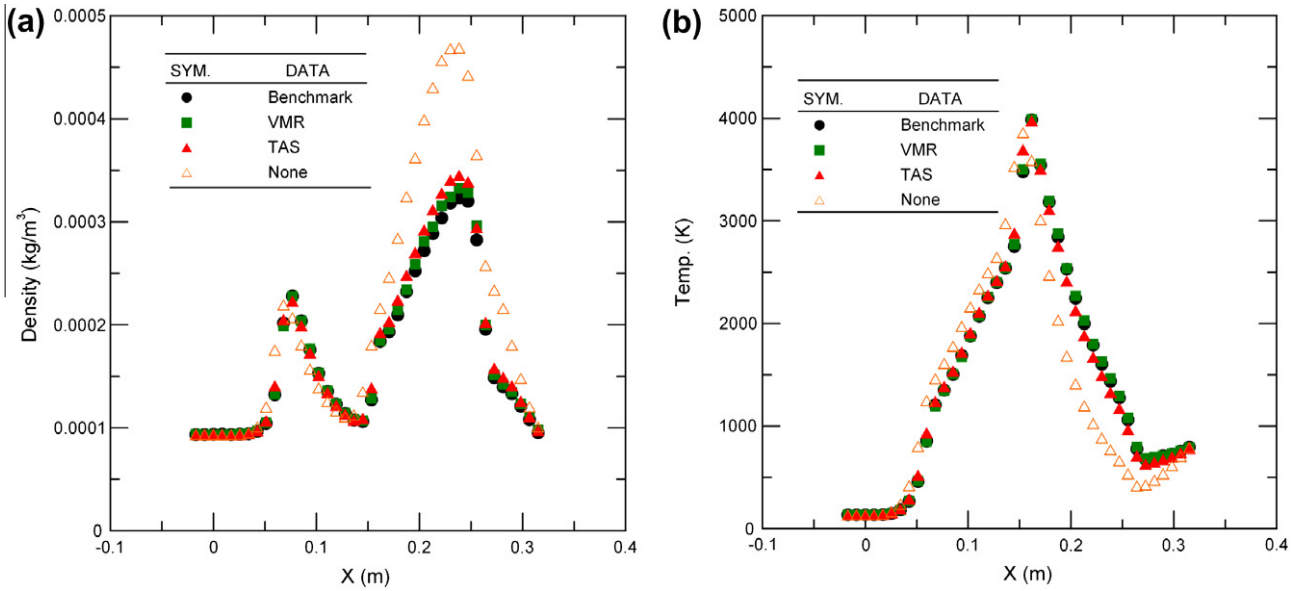

Fig. 14. Property distributions in the horizontal channel along a horizontal line (y = 0.02645 m). (a) Density. (b) Temperature.

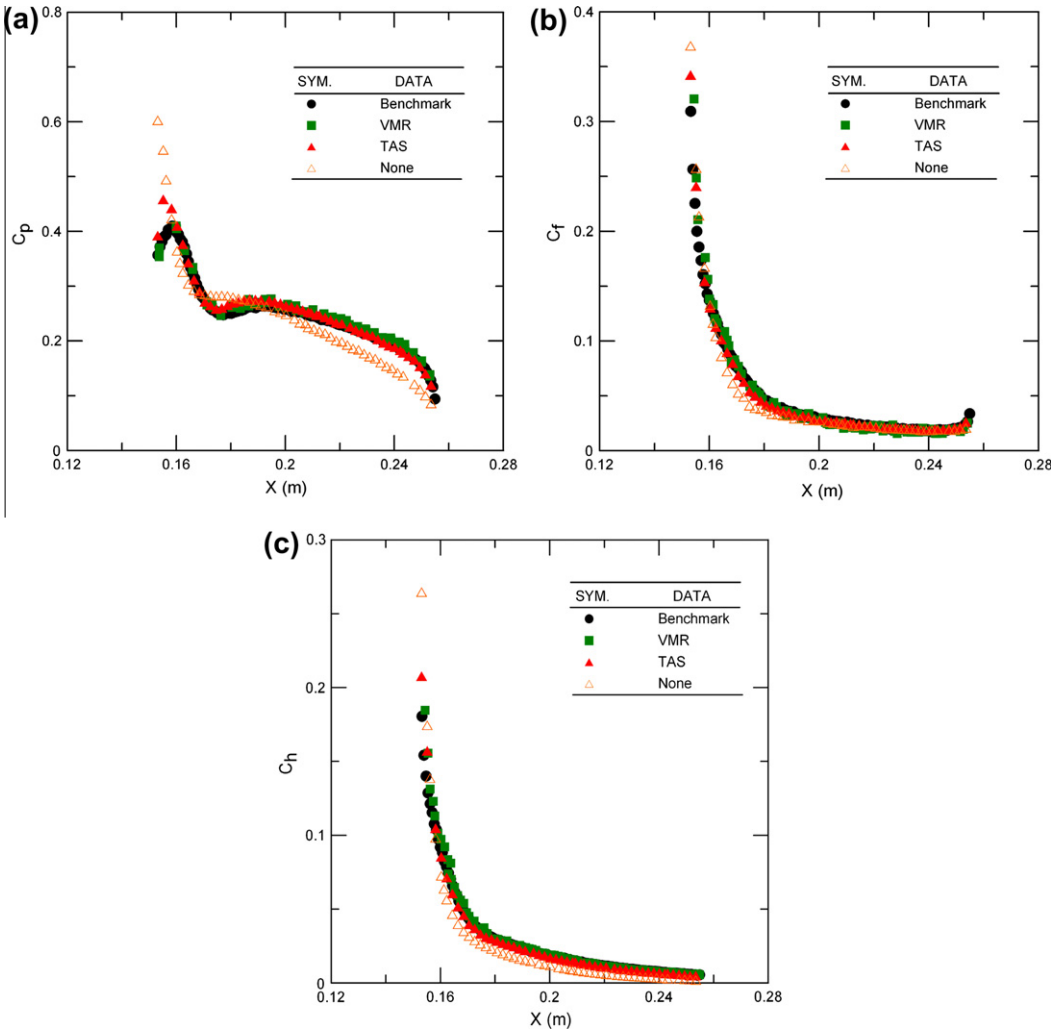

Fig. 15. Surface property distributions along the top channel wall. (a) Local pressure coefficient. (b) Local friction coefficient. (c) Local heat transfer coefficient.

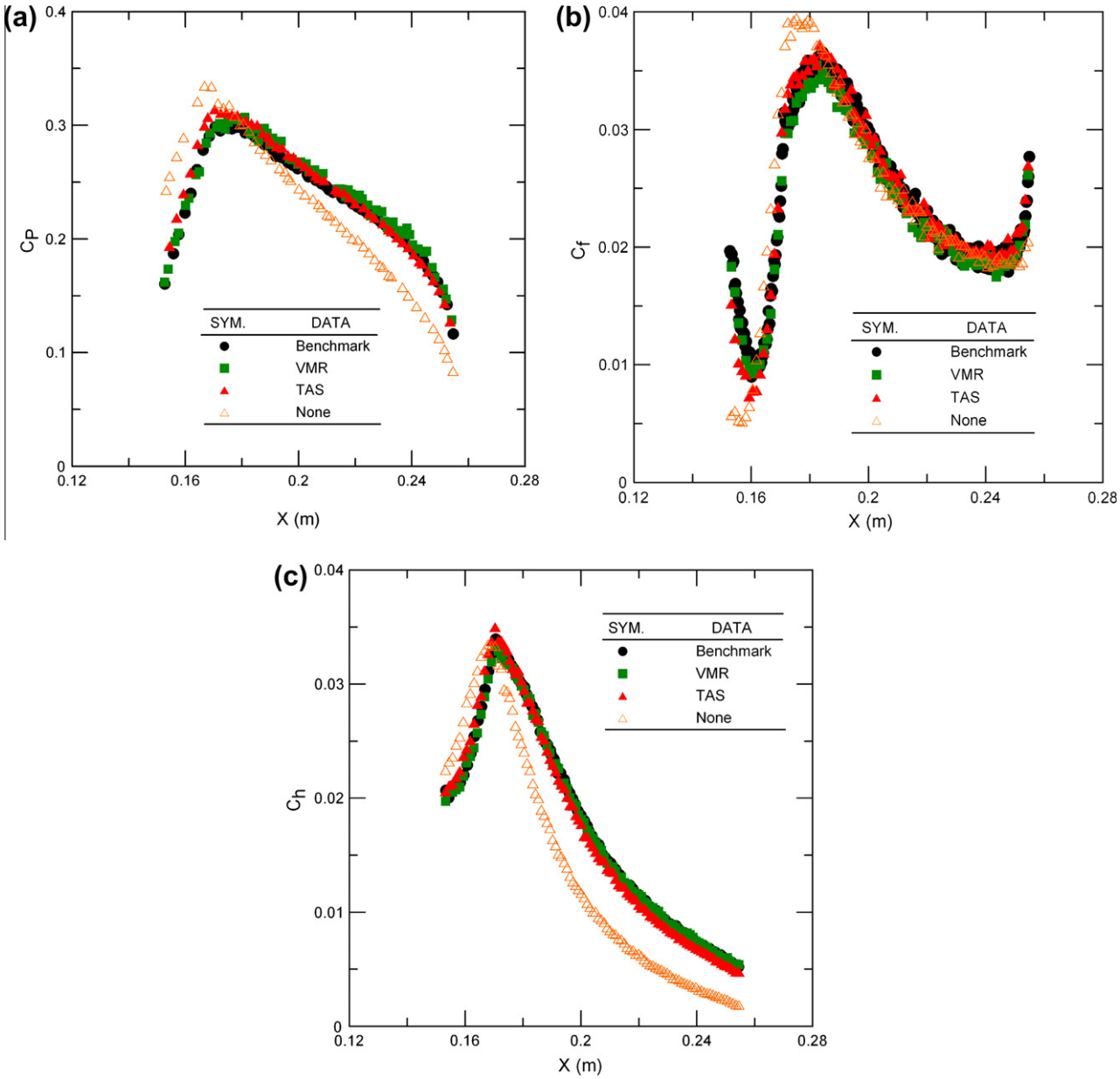

Fig. 16. Surface property distributions along the bottom channel wall. (a) Local pressure coefficient. (b) Local friction coefficient. (c) Local heat transfer coefficient.

larger than the latter along the bottom channel wall. They clearly show that only the results of Case-VMR agree very well with those of Case-Benchmark.

4. Conclusions

In this paper, a two-level virtual mesh refinement (VMR) algorithm is described and implemented in the parallel direct simulation Monte Carlo code PDSC, utilizing unstructured grids. Two test cases, an M-10 argon flow past a cylinder and an M-15 argon scramjet flow, were used to demonstrate that the use of VMR can reproduce the results of the benchmark quadrilateral mesh with cell size 1/2–1/3 of the mean free path, at greatly reduced mesh density and computational time. Future directions for research include: (1) testing the VMR algorithms in a gas system with excited rotational and vibrational degrees of freedom, and (2) extension to the three-dimensional case.

Acknowledgments

The authors would like to express their sincere thanks to the financial support through the Grant NSC-96-2628-E-009-136-MY3 by National Science Council of TAIWAN. The high-performance computing resources provided by the National Center for High-performance Computing are highly appreciated.

References

[1] Bird GA. Molecular gas dynamics and the direct simulation of gas flows. Oxford: Clarendon Press; 1994.
[2] Wagner W. A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation. J Stat Phys 1992;66:1011–44.
[3] LeBeau GJ. A parallel implementation of the direct simulation Monte Carlo method. Comput Methods Appl Mech Eng 1999;174:319.
[4] Wu JS, Tseng KC, Wu FY. Parallel three-dimensional DSMC method using mesh refinement and variable time-step scheme. Comput Phys Commun 2004;162:166–87.

[5] Wang L, Harvey JK. Rarefied gas dynamics. In: Harvey J, Lord G. editors. 19th international symposium on rarefied gas dynamics; 1994. p. 843–49.
[6] Wu JS, Tseng KC. Parallel DSMC method using dynamic domain decomposition. Int J Numer Meth Eng 2005;63:37–76.
[7] Su CC, Tseng KC, Cave HM, Wu JS, Lian YY, Kuo TC, et al. Implementation of a transient adaptive sub-cell module for the parallel-DSMC code using unstructured grids. Comput Fluids 2010;39:1136–45.
[8] Bird GA. Sophisticated versus simple DSMC. In: Proceedings of 25th international symposium on rarefied gas dynamics. St. Petersburg, Russia; 2006.