INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **44**:1317–1335 (DOI: 10.1002/fld.644)

A parallel cell-based DSMC method on unstructured adaptive meshes

Min Gyu Kim, Hyoung Soon Kim and Oh Joon Kwon*,†

Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea

SUMMARY

A parallel DSMC method based on a cell-based data structure is developed for the efficient simulation of rarefied gas flows on PC-clusters. Parallel computation is made by decomposing the computational domain into several subdomains. Dynamic load balancing between processors is achieved based on the number of simulation particles and the number of cells allocated in each subdomain. Adjustment of cell size is also made through mesh adaptation for the improvement of solution accuracy and the efficient usage of meshes. Applications were made for a two-dimensional supersonic leading-edge flow, the axi-symmetric Rothe's nozzle, and the open hollow cylinder flare flow for validation. It was found that the present method is an efficient tool for the simulation of rarefied gas flows on PC-based parallel machines. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: DSMC method; cell-based scheme; parallelization; unstructured mesh; mesh adaptation; load balancing

1. INTRODUCTION

The Direct Simulation Monte Carlo (DSMC) method, originally developed by Bird [1] in 1960s, is one of the most practical techniques for solving the non-linear Boltzmann equation. Finite number of representative particles generated inside the computer memory travels and collides with each other in the cyber space under the assumption that particle movement and collision phases can be decoupled. For the solution accuracy, cell size restriction is usually given such that the characteristic length of cells is smaller than the molecular mean free path. The time step size should also be smaller than the mean collision time.

The dependency of viscosity and thermal conductivity on cell size was studied analytically by Alexander *et al.* [2]. It was reported that significant error can occur when the cell size

*Correspondence to: Oh Joon Kwon, Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea.
†E-mail: ojkwon@kaist.ac.kr

Contract/grant sponsor: Agency for Defence Development; contract/grant number: ADD-98-5-5

Received 24 February 2003
Revised 18 August 2003

Copyright © 2004 John Wiley & Sons, Ltd.

becomes larger than the mean free path. Bird [3] showed that the vortex strength predicted by DSMC method is also affected by the cell size. Since large cells degrade the solution accuracy and small cells containing insufficient number of particles cannot ensure the statistical requirement, proper cell distribution over the solution domain becomes one of the fundamental issues for DSMC methods. This is particularly true for complex flows involving flow separation and shock waves since given computational mesh sometimes yields unacceptable results. To remedy this difficulty, a grid system with variable subcells was suggested by Bird [3] and Talbot-Stern and Auld [4]. However, the problem still remains since the cell-averaged macroscopic flow properties are assigned for each cell, not for its subcells. Other attempts were also made based on the mesh adaptation technique by refining cells on Cartesian grids [5] and unstructured meshes [6]. However, the effect of mesh adaptation has not been studied in detail, particularly under the parallel computation environment frequently adopted for solving practical large-scale problems.

During the past decade, significant efforts have been made to overcome the large computational resources required for DSMC simulations. One pertinent choice is to parallelize the code since simulation particles move independently from each other and their collision is made within each local cell. Recently, rapid improvement of computer hardware technologies made it possible to construct PC-based parallel clusters, which show compatible performance to that of the traditional supercomputers. However, these clusters are equipped with memory slower than the large shared memory of supercomputers for data access. The fast cache memory can be used to enhance the speed, but only to a limited extent. Thus, the system performance can be significantly improved by properly avoiding the cache-missing phenomena on these clusters. This cache-missing frequently occurs when accessing cross reference arrays from the particle-based data structure. To avoid this problem, a cell-based DSMC method was suggested by Dietrich and Boyd [6] using a modular data structure and a list-linked technique.

In order to obtain high parallel efficiency, it is important to maintain the load balancing between processors above a certain level. McDonald [7] distributed blocks containing group of cells in each processor to achieve the load balancing on structured grids. A similar concept of using 'slab' is also adopted for Cartesian grids by LeBeau [5]. For unstructured triangular meshes, Robinson and Harvey [8] used an adaptive domain decomposition technique by successively transferring cells attached to the domain boundary from subdomains with high computational loads to their neighbours. Since each cell can be treated independently, load balancing between processors can be handled easier by using the cell-based method than using blocks or 'slaps'. However, achievement of load balancing for multiple levels of mesh adaptation has not been previously studied.

In the present study, a parallel cell-based DSMC method is developed for the efficient simulation of rarefied gas flows on PC-based clusters by using unstructured meshes. A multi-level mesh adaptation is also incorporated to enhance the accuracy of the solution and to increase the grid efficiency. A dynamic load balancing procedure is simultaneously applied to maximize the parallel efficiency by accounting for varying number of particles and grid cells allocated on each processor. Calculation was made for flow in equilibrium state to compare the performance between the particle-based scheme and the present cell-based method. A two-dimensional supersonic flat-plate flow was also calculated to demonstrate the validity of the present mesh adaptation technique. The effect of load balancing was tested for the axisymmetric Rothe's nozzle flow. Finally, an open hollow cylinder flare flow was calculated

Copyright © 2004 John Wiley & Sons, Ltd.

and the results were compared with the experiment and those of a particle-based scheme. The effect of cell size with successive application of mesh adaptation was examined for this complex flow involving shock waves and flow separation. The parallel efficiency and the effect of dynamic load balancing were also tested to demonstrate the validity of the present method.

2. NUMERICAL METHOD

In the present study, the computational mesh is constructed by using unstructured triangular cells. The collision pairs are determined within each subcell, which is obtained by dividing the triangular cell into four. The Variable Hard Sphere model proposed by Bird [9] is used as the molecular model. The specular model is applied at the plane of symmetry and the solid boundary is treated by using the diffuse model. For simulating diatomic flows, the Larsen–Borgnakke phenomenological model is adopted to redistribute translational and internal energies [10]. The macroscopic properties are obtained by successively sampling the molecular properties within each cell and by averaging these properties over the cell volume. The collision sampling is made by using the No Time Counter method [11]. Tracing of particle movement on triangular cells is achieved by adopting the linear shape function at each node [8].

The cell-based scheme originally suggested by Dietirch and Boyd [6] has a modular structure, which can be extended or inserted easily into other programs without extensive modification. Each module contains information about the cell, its nodes, cell faces and the connectivity to the neighbouring cells, while long individual array for each of them is used in the particle-based scheme. In Figure 1, the difference in memory usage between the two schemes is presented schematically. It is shown that the data required for flow simulation is more closely located in the computer memory for the cell-based scheme so that cache missing can be avoided effectively.

Figure 2 shows the schematic of the data structure and its linkage of the two schemes. For the particle-based scheme, information about cell data, cell volume, connectivity and node points is obtained from the cell array. Then particle data can be obtained by connecting the particle index from the cell array to the one in the particle array through the cross-reference array. Thus, the particle-based scheme requires more time and memory for data loading and is more vulnerable to cache missing. However, the cell-based scheme on the modular structure easily access the information about simulation particles, which is directly connected to each cell by the list-linked technique. When a particle leaves the current cell and moves into the target cell, the linkage of the particle is disconnected from the current cell and is reattached to the new cell. For the list-linked data structure of the present cell-based scheme, additional index routine is not required by using the dynamic memory allocation of FORTRAN 90 or C language. Since each cell is treated independently, redistribution of cells between parallel processors for dynamic load balancing can also be achieved easily without large computational overhead.

Parallel processing is made by dividing the computational domain into several subdomains by using the MeTiS library, which is based on the k-way, n-partitioning technique by Karypis [12]. After each time step advancement, information about the particles and their properties is exchanged through the subdomain boundary by using the Message Passing

Copyright © 2004 John Wiley & Sons, Ltd.

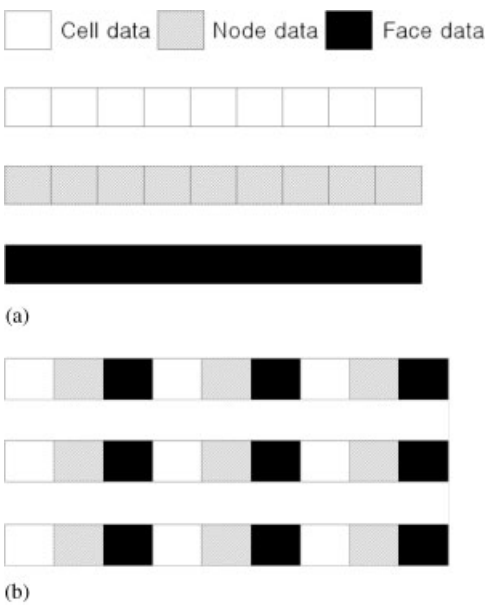

Figure 1. Comparison of the memory usage between particle-based and cell-based schemes. (a) Particle-based scheme and (b) cell-based scheme.

Interface (MPI) library. In order to achieve high parallel efficiency, computational load assigned to each processor should be maintained evenly throughout the calculation. In DSMC simulations, computational time is strongly dependent not only on the number of cells allocated in each processor but also on the number of particles contained within each subdomain. At the beginning of calculation, equal number of cells is distributed for each processor. As the calculation proceeds and the simulation particles develop, load balancing between processors is continuously monitored by checking the parallel efficiency. The parallel efficiency is defined such that pure calculation time of each processor can be tested:

$$\text{Parallel efficiency: } P = \frac{T_t - T_c - T_i}{T_t}$$

where T_t, T_c and T_i represent the total calculation time, the communication time and the idle time, respectively. When the parallel efficiency decreases or as its difference between processors becomes large, domain decomposition is dynamically updated by accounting for the number of particles and the number of cells simultaneously. This is done by assigning different weighting factor for each grid cell, which is determined based on the number of particles contained within the cell. However, at the early stage of the simulation, frequent application of dynamic load balancing was restricted since the number of particles contained in each processor rapidly changes.

In order to improve the solution accuracy and the grid efficiency, a mesh adaptation technique is also adopted in the present study. Initially, calculation is made on the coarse mesh

Copyright © 2004 John Wiley & Sons, Ltd.

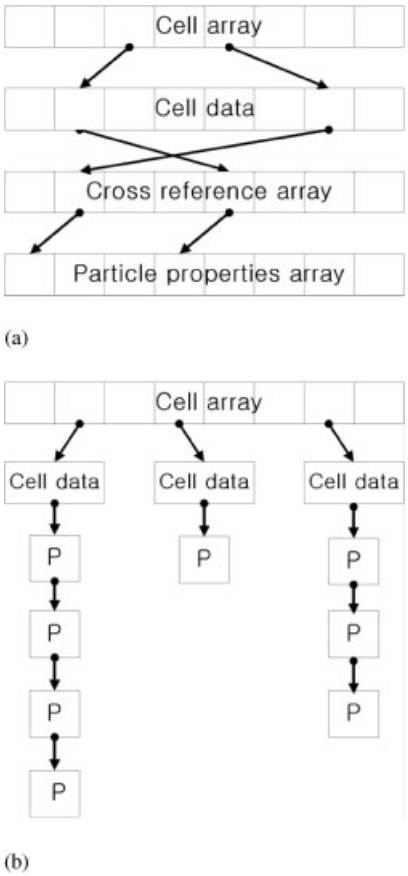

Figure 2. Comparison of the path for memory accessing between particle-based and cell-based schemes [6]. (a) Particle-based scheme and (b) cell-based scheme.

until a statistically converged solution is obtained. Then, the value of adaptation indicators is examined for each cell based on the flow properties such as the ratio of the mean free path to the characteristic length of cell, the velocity gradient or the density gradient. The tagged cells are divided into either four or two child cells depending on the connectivity with surrounding cells. Linkage of the particles to the refined cells is also taken care of. Then, new domain decomposition is made for load balancing and calculation restarts on the refined mesh. This procedure is repeated until a satisfactory result is obtained. Application of mesh adaptation stops when the refined small cells do not possess sufficient number of particles for statistical mean flow sampling. When the cell refinement is made at the communication boundary, information about the cell division is also exchanged between neighbouring processors. Data collection and redistribution for mesh adaptation and domain decomposition can also be achieved easily by using the modular data structure of the present cell-based scheme.

Copyright © 2004 John Wiley & Sons, Ltd.

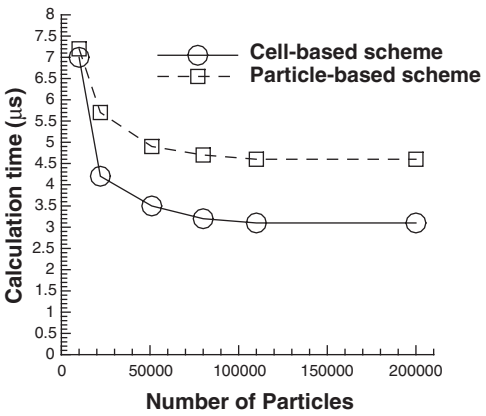

Figure 3. Comparison of calculation time per particle for uniform flow.

3. RESULTS AND DISCUSSION

3.1. 2-D Equilibrium box flow

A uniform flow inside a square domain having a size of 2×2 and 100 cells was initially studied to compare performance between the particle-based and cell-based schemes. The temperature was fixed at all boundaries and the mean flow velocity was set to zero. Calculation was made on a single processor. The results confirmed that the number of particles and the temperature distribution were maintained nearly unchanged for both methods. Six tests were conducted with various number of particles ranging from 10 000 to 200 000. Figure 3 shows the required calculation time per particle as a function of the number of particles. It is shown that the simulation time is not significantly different between the two schemes when small number of particles is used. However, as the number of simulation particles increases, the cell-based scheme requires much less computational time than that of the particle-based scheme, which demonstrates the efficiency of the cell-based approach. The time required per particle was approximately 4.51 μs for the particle-based scheme while 3.35 μs was spent using the cell-based scheme, representing about 25% reduction. Also, approximately 45% reduction of the computer memory was observed for the cell-based scheme, requiring 197 bytes compared to 356 bytes of the particle-based scheme.

3.2. 2-D Supersonic flat-plate flow

Next, supersonic flow over a flat plate was calculated to demonstrate the mesh adaptation procedure. The initial uniform mesh consisted of 3181 nodes and 6145 triangles. The plate is located along the bottom boundary from $x = 0.1$ to 1. The Knudsen number based on the length of the plate is 0.0143, and the uniform flow is entering the left boundary at a freestream Mach number of 4. The temperature on the flat plate was fixed to 500 K, and the vacuum condition was imposed at the right boundary of the computational domain. Calculation

Copyright © 2004 John Wiley & Sons, Ltd.

was made on five parallel processors using approximately 530 000 particles. This number is enough for excluding the dependency of the solution on the number of simulation particles, even for the refined cells after mesh adaptation. Since the main purpose of the present calculation is to demonstrate the mesh adaptation procedure, application of load balancing was not made.

Figure 4 shows the mesh adaptation sequence and the refined cells. In order to capture high gradient flow regions such as shock wave and boundary layer, density and velocity gradients were adopted as the error indicators for mesh adaptation. The refined meshes resulted in 10 751 cells and 5719 nodes after the first adaptation, and 22 769 cells and 11 994 nodes with two levels. It is shown that refined small cells are distributed mostly along the oblique shock wave and inside the boundary layer as expected. The subdomain boundaries are also shown as bold lines in the figure. For a comparison purpose, calculation was also made on a fine mesh with uniformly distributed small cells having a characteristic size smaller than the mean free path. This size is approximately equivalent to that of the cells obtained after two level mesh adaptation. The fine mesh contains 98 336 cells and 50 224 nodes as shown in Figure 4(d), which are approximately 4 times more than those of the mesh after the second adaptation.

In Figure 5, density contours corresponding to the meshes in Figure 4 are presented. It is shown that the oblique shock wave is resolved better on the refined mesh having a tight shock structure and less wiggles on the contours. Development of the boundary layer is also well captured after the mesh adaptation. It can be observed that the result after two levels of mesh adaptation shows a comparable resolution to that of the uniform fine mesh using cells less than 25%. At local regions of low density, particularly inside the boundary layer or behind the shock wave, the result with mesh adaptation shows less fluctuation than that of the fine mesh since adequate number of particles was maintained for each cell for better statistical sampling.

3.3. Axi-symmetric Rothe's nozzle flow

In order to validate the implementation of load balancing, axi-symmetric flow through the Rothe's nozzle [13] was studied on a mesh containing 19 263 cells and 9 901 nodes. The Knudsen number based on the throat diameter is 0.023. The inlet boundary of the calculation was set at the nozzle throat where the flow becomes choked and the mean flow variables were obtained by assuming an isentropic expansion from the stagnation. Since the main purpose of this validation is to demonstrate the domain decomposition, load balancing was not applied dynamically, but forced at every 50 iterations (in which each iteration is equivalent to 20 time step advancement). Ten applications were made at the early stage of the simulation and the last one after 3000 iterations.

In Figure 6, the result of sequential load balancing is shown as domain decomposition for 6 processors. Initially, equal number of cells is allocated for all processors as shown in Figure 6 (a). Since large number of small size cells is distributed near the throat, the subdomains are mostly packed at this region. As simulation particles flow into the computational domain through the nozzle throat and propagate downstream, the subdomain boundaries were readjusted such that load balancing between processors is satisfied. The result is presented at the subsequent iteration levels as shown in Figures 6(b)–6(d), confirming the flow development through the nozzle.

Copyright © 2004 John Wiley & Sons, Ltd.

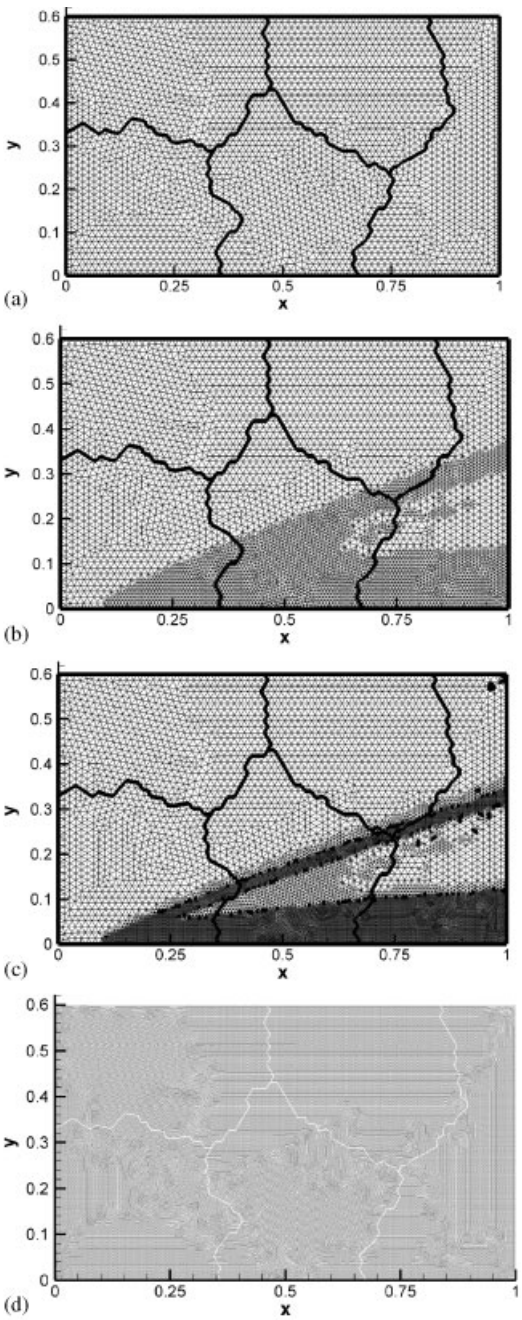

Figure 4. Mesh adaptation sequence and fine mesh for flat-plate supersonic flow. (a) Initial mesh; (b) one level mesh adaptation; (c) two level mesh adaptation and (d) fine mesh with uniformly distributed small cells.

Copyright © 2004 John Wiley & Sons, Ltd.

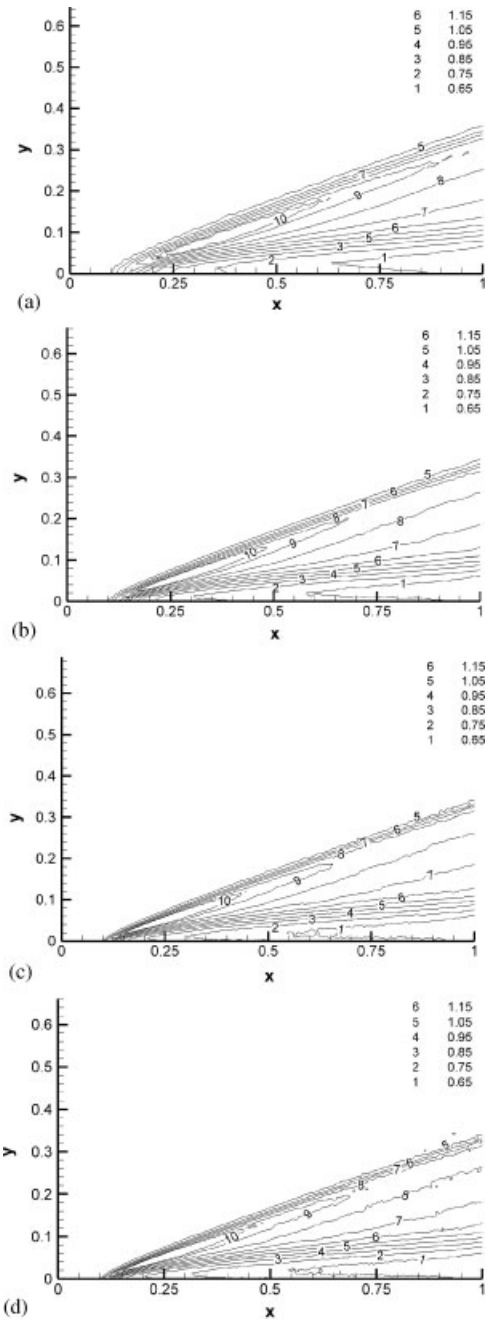

Figure 5. Comparison of density contours for flat-plate supersonic flow at $Kn = 0.0143$ and $M = 4$. (a) Initial mesh; (b) one level mesh adaptation; (c) two level mesh adaptation and (d) fine mesh with uniformly distributed small cells.

Copyright © 2004 John Wiley & Sons, Ltd.

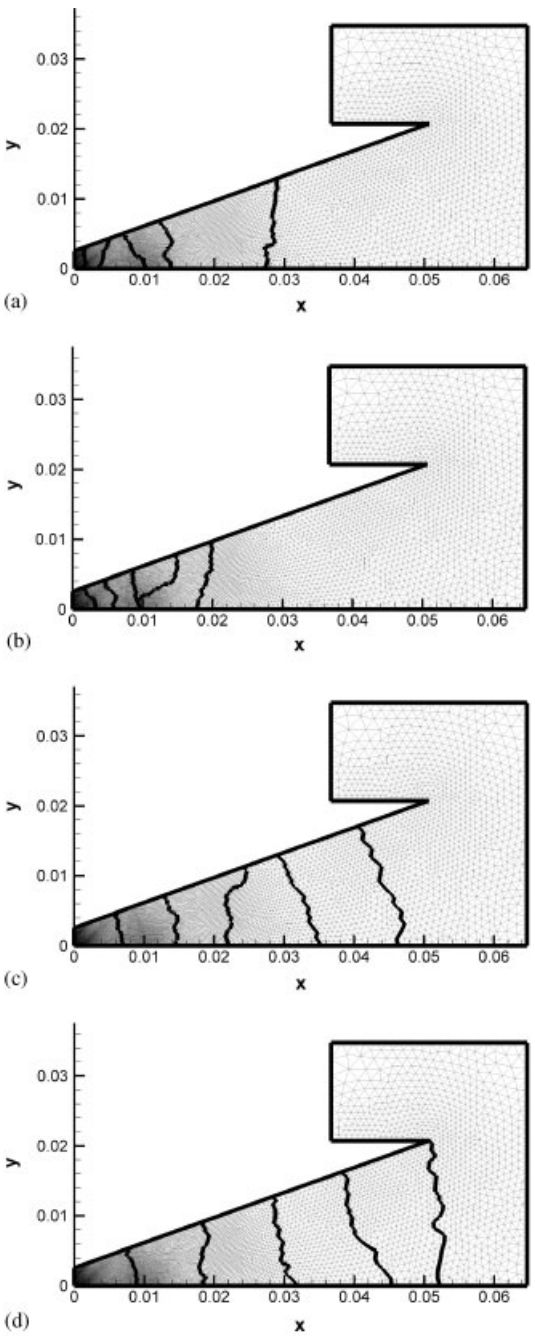

Figure 6. Sequence of load balancing and parallel domain decomposition for Rothe's nozzle flow. (a) Initial subdomain partition; (b) subdomain partition after 100 iterations; (c) subdomain partition after 300 iterations and (d) subdomain partition after 500 iterations.

Copyright © 2004 John Wiley & Sons, Ltd.

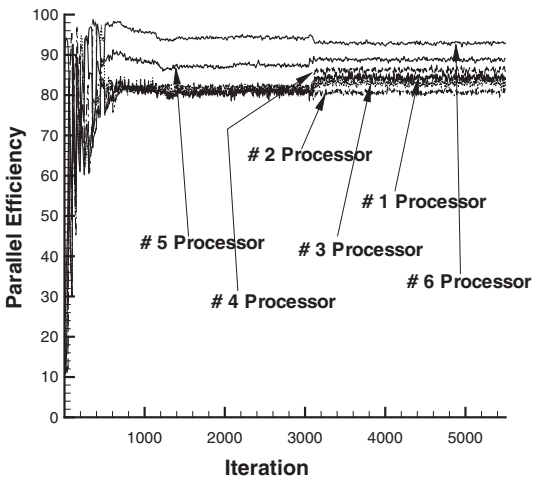

Figure 7. History of parallel efficiency of each processor for successive load balancing and domain decomposition.

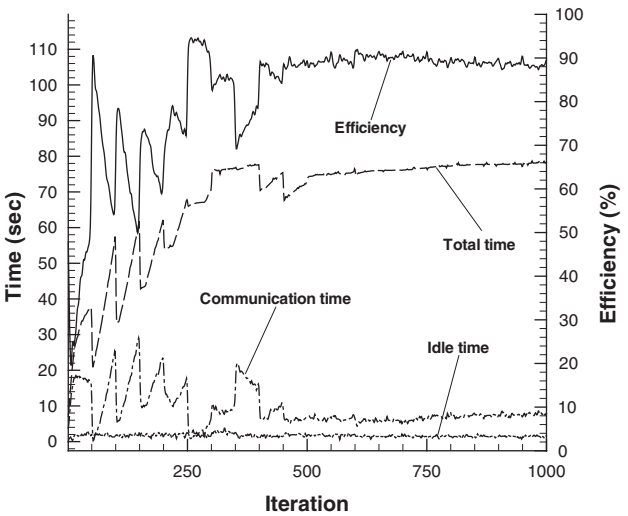

Figure 8. History of parallel efficiency and required computer time for a typical processor.

In Figure 7, the parallel efficiency of each processor is presented during the simulation. Once the flow establishes after highly transient early stage of the iteration, the parallel efficiency is maintained relatively high over 80% for all processors throughout the calculation as a result of successive load balancing and domain decomposition. The history of parallel efficiency and required computer time of a typical processor is presented in Figure 8 for the early stage of the simulation. Initially, the total calculation time increases rapidly due to the increasing number of particles migrating into the subdomain of this processor. Rapid improvement of

Copyright © 2004 John Wiley & Sons, Ltd.

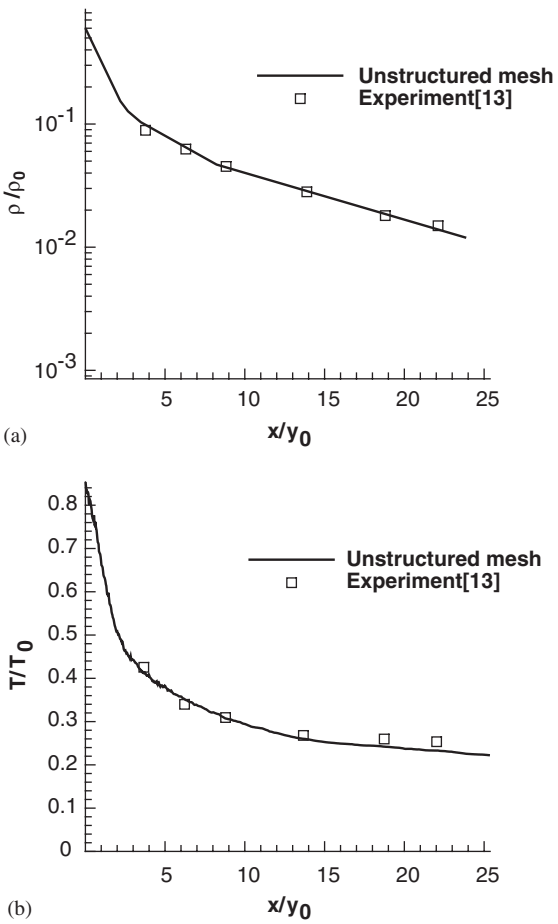

Figure 9. Density and temperature distributions along the centre line of the Rothe's nozzle. (a) Density distribution and (b) temperature distribution.

the parallel efficiency is also observed since portion of the pure calculation time compared to the other two elements becomes large. When new domain decomposition is applied and the simulation particles spread evenly for all processors, the parallel efficiency becomes higher while the total calculation time and the time for communication drop. This pattern is repeated at every application of the domain decomposition until the parallel efficiency and the total time settle down. The idle time remains unchanged throughout the calculation.

Figure 9 shows the comparisons of the density and temperature distributions along the centre line of the nozzle with the experiment [13]. Good agreement is obtained between the two results, confirming the validity of the present parallel DSMC method.

3.4. Axi-symmetric open hollow cylinder flare flow

The last calculation was made for flow around a well-known open hollow cylinder flare [14–16] by applying successive mesh adaptation, parallelization and dynamic load balancing

Copyright © 2004 John Wiley & Sons, Ltd.

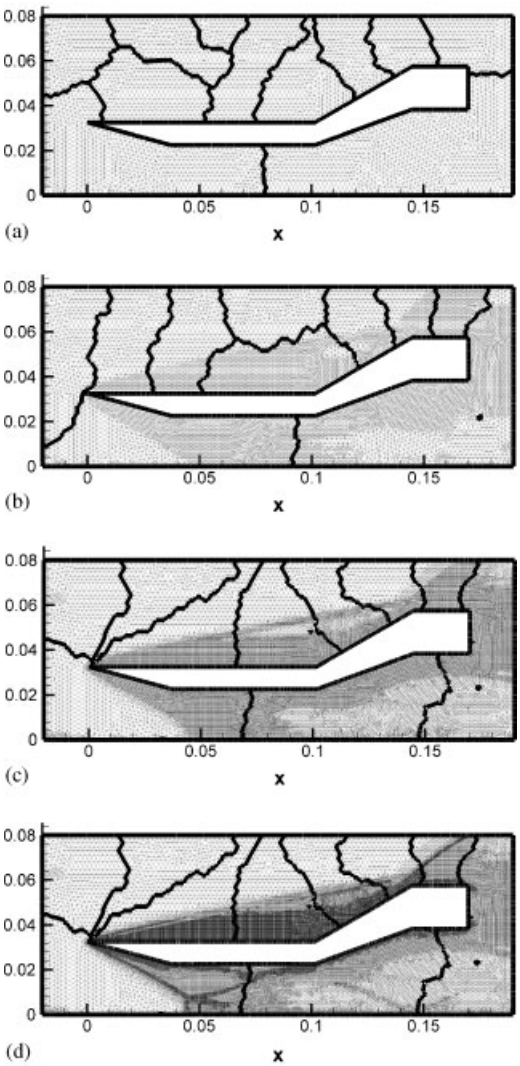

Figure 10. Mesh adaptation sequence for the open hollow cylinder flare at $Kn = 8.87 \times 10^{-4}$ and $M = 9.91$. (a) Initial mesh; (b) one level mesh adaptation; (c) two level mesh adaptation and (d) three level mesh adaptation.

simultaneously. The parallel efficiency and the effect of mesh adaptation on the solution accuracy were tested to validate the present method using 10 processors. The flow contains sophisticated features such as shock waves, shock-boundary interaction, local flow separation and reattachment. The hollow cylinder flare has a sharp leading-edge with a bevel angle of 15°. The compression flare is inclined at 30° with respect to the cylinder. Calculation was made for a Knudsen number of 8.87×10^{-4} based on the outer surface length and at a freestream Mach number of 9.91. The wall temperature was fixed to 293 K. The far-field boundary was

Copyright © 2004 John Wiley & Sons, Ltd.

Table I. Comparison of meshes and predicted results for the open hollow cylinder flare flow.

	Nodes	Cells	Separation location	Reattachment location
Initial mesh	8640	16 853	0.991	1.035
1st adaptation	16 529	33 536	0.964	1.15
2nd adaptation	38 524	74 580	0.942	1.2
3rd adaptation	63 634	123 912	0.799	1.31
Structured grid [14]	—	78 100	0.755	1.322
Experiment [17]	—	—	0.76	1.34

set at a distance away from the hollow cylinder such that the Maxwellian distribution can be safely applied.

Figure 10 shows the three level mesh adaptation sequence as cells are refined along the shock wave and inside the boundary layer. Mesh adaptation was made based on the density and velocity gradients. The density gradient was used mostly for capturing the shock waves, and the velocity gradient is more effective for flow inside the boundary layer and for the separated flow region. The cell refinement was made up to three levels such that sufficient number of simulation particles was maintained inside each cell to assure reasonable accuracy of the statistical solution. Calculation started on the initial mesh having cells with a uniform size over the whole computational domain as shown in the figure. For the first mesh adaptation, cells are mostly refined inside the region confined by the oblique shock waves attached to the leading edge of the cylinder. An additional level of mesh adaptation shows that more refinement was made along the shock wave, inside the boundary layer, and in the wake region. A similar process also continued for the third level of mesh adaptation, providing more refinement at those regions. The subdomain boundary between processors is also presented in the figure as shown with bold lines at each level of mesh adaptation. It was demonstrated that application of load balancing continuously updated the domain decomposition by redistributing cells for each processor to maintain high parallel efficiency.

Summary of the number of cells and nodes for each adaptation level is given in Table I. It is shown that the final mesh contains approximately 7 times more cells than that of the initial mesh. The mesh has over 50% more cells than the structured grid [14]. However, this is mostly due to the fact that the computational domain of the structured grid was confined only to the upper region of the cylinder filled with stretched, high-aspect ratio cells.

In Figure 11, streamlines around the hollow cylinder are compared at each level of mesh adaptation. For the initial coarse mesh, only general features of the flow such as flow deflection due to the leading-edge oblique shock waves and formation of the reflection shock are shown. After one level of mesh adaptation, capturing of more accurate flow physics becomes evident, showing local flow separation at the corner of the compression flare and on the bottom surface of the cylinder. This tendency continues as the cell refinement continues for additional levels, demonstrating the effect of mesh adaptation.

The density contours on the initial and final meshes are compared in Figure 12. It is shown that the oblique shock waves and the reflection shock are much more clearly defined on the final mesh than those of the coarse initial mesh. The shock structure at the symmetric plane

Copyright © 2004 John Wiley & Sons, Ltd.

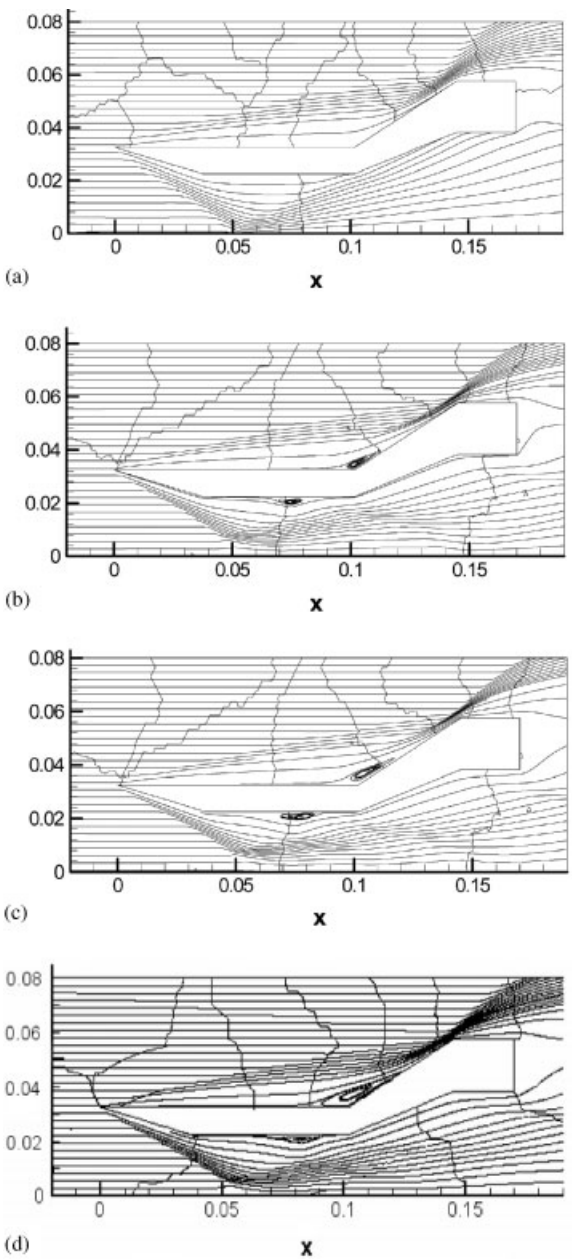

Figure 11. Streamlines around the open hollow cylinder flare at different mesh resolutions. (a) Initial mesh; (b) one level mesh adaptation; (c) two level mesh adaptation and (d) three level mesh adaptation.

Copyright © 2004 John Wiley & Sons, Ltd.

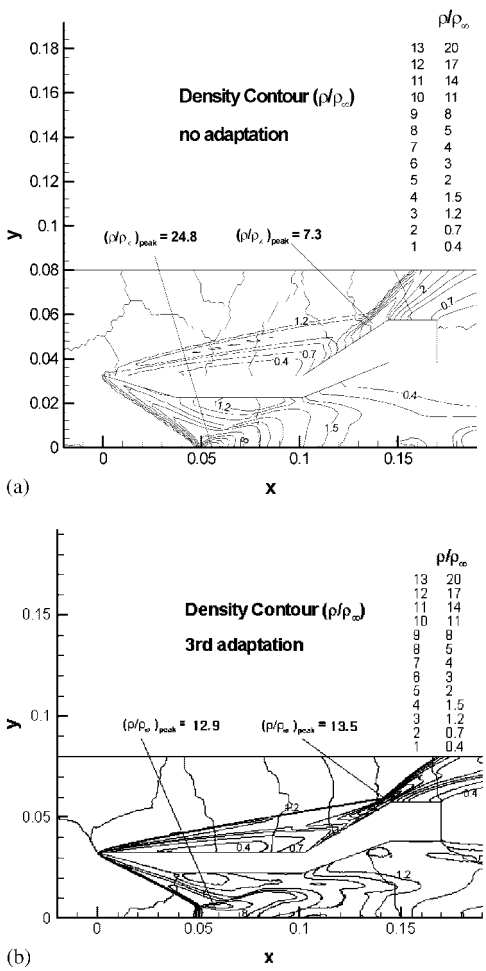

Figure 12. Comparison of density contours for the open hollow cylinder flare. (a) Initial mesh and (b) three level mesh adaptation.

is also clearly captured for the refined mesh, demonstrating the effect of mesh resolution on the solution accuracy. The size of the predicted separated flow region is 0.511 mm after three levels of mesh adaptation. The result compares well with the experiment [17] and that of the structured grid calculation by Moss [14] as shown in Table I. The predicted separated flow region on the initial coarse mesh is 0.044 mm, which is significantly smaller than is expected.

In Figure 13, the parallel efficiency of each processor is presented as a function of iteration. In the beginning of the calculation, wild fluctuation of the parallel efficiency is observed as the simulation particles enter the flow domain and the number of particles increases. As the flow developed and the mesh adaptations were applied, repeated domain decomposition for load balancing was also made by considering the number of particles and the number of cells

Copyright © 2004 John Wiley & Sons, Ltd.

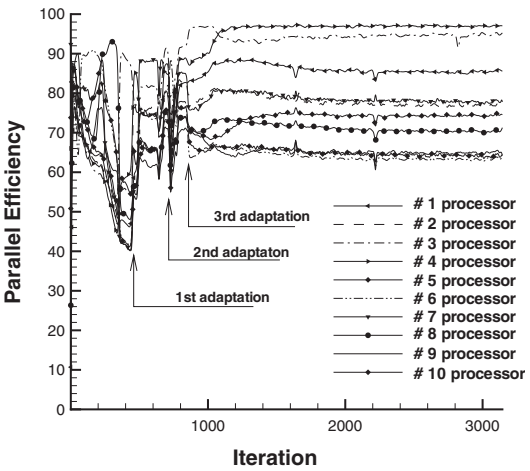

Figure 13. History of parallel efficiency of each processor for the open hollow cylinder flare flow calculation through mesh adaptation and load balancing.

contained in each subdomain. Once the flow established after the third mesh adaptation, it is shown that the parallel efficiency was maintained as high as 65% for all processors. The relatively low efficiency compared to that of the Rothe's nozzle case is mainly due to the lengthy flow boundary shared in the subdomain, which make precise load balancing more difficult. Further improvement is also possible by considering additional elements such as particle collision and sampling in load balancing.

In Figure 14, pressure coefficient, heat flux and shear stress distributions are presented along the outer surface of the hollow cylinder. It is observed that the results obtained for the refined mesh compare reasonably well with the experiment [17] and those of the structured grid [14]. It is shown that the coarse initial mesh significantly overpredicts the heat flux and the shear stress, confirming that the mesh resolution not only affects the flow field but also is very important for the accurate prediction of surface flow properties.

4. CONCLUSION

A parallel cell-based DSMC method is developed for the efficient simulation of rarefied gas flows. It was shown that simulation time and required memory can be significantly reduced by using the cell-based scheme. Sequential mesh adaptation and dynamic load balancing were used to enhance the solution accuracy and to maintain high parallel efficiency of the calculation. Applications were made to a 2-D supersonic flat-plate flow, axi-symmetric Rothe's nozzle flow, and the open hollow cylinder flare flow. The effect of mesh adaptation on the solution accuracy was validated and the improvement of parallel efficiency was demonstrated by applying the dynamic load balancing. It was shown that the unstructured mesh technique coupled with mesh adaptation and parallel computing is an efficient tool for the simulation of rarefied gas flows using DSMC method.

Copyright © 2004 John Wiley & Sons, Ltd.

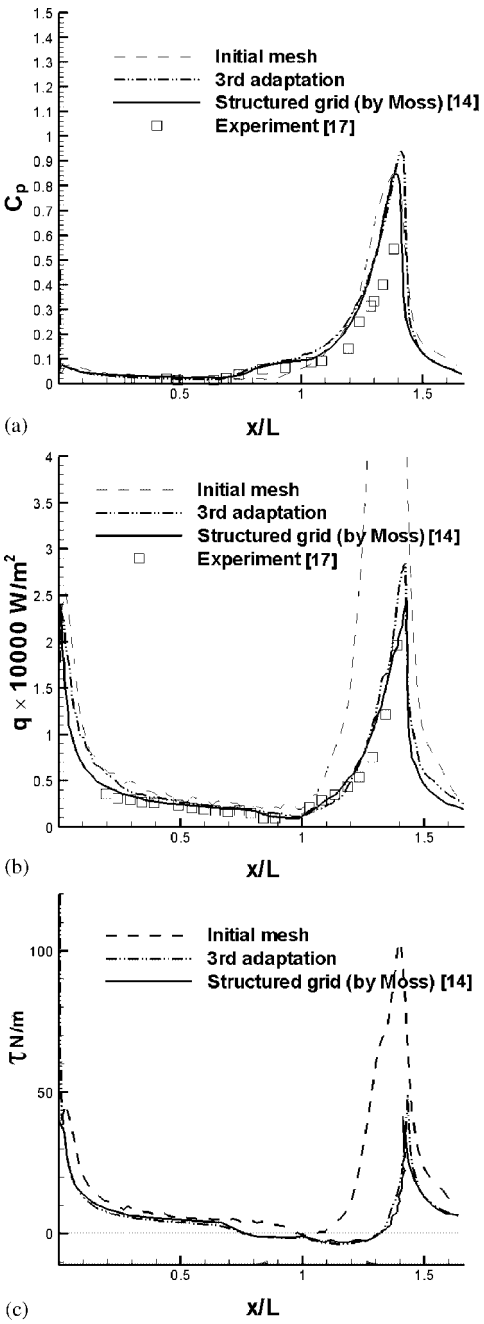

Figure 14. Predicted results along the outer surface of the hollow cylinder. (a) Pressure coefficient distribution; (b) heat flux distribution and (c) shear stress distribution.

Copyright © 2004 John Wiley & Sons, Ltd.

ACKNOWLEDGEMENTS

This work was supported by the Agency for Defence Development through Basic Research Fund ADD-98-5-5.

REFERENCES

1. Bird GA. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford University Press: London, 1994.
2. Alexander FJ, Garcia AL, Alder BJ. Cell size dependence of transport coefficient in stochastic particle algorithm. *Physics of Fluids* 1998; **10**(6):1540–1542.
3. Bird GA. Direct simulation of high-vorticity gas flows. *Physics of Fluids A* 1987; **30**(2):364–366.
4. Talbot-Stern JL, Auld DJ. Direct simulation (Monte Carlo) of two dimensional vortex streets. *AIAA Paper* 98-2671, 1998.
5. LeBeau GJ. A parallel implementation of the direct simulation Monte Carlo. *Computational Methods of Applied Mechanical Engineering* 1999; **174**(3):319–337.
6. Dietrich S, Boyd ID. Scalar and parallel optimized implementation of the direct simulation Monte Carlo method. *Journal of Computational Physics* 1996; **126**(2):328–342.
7. McDonald JD. Particle simulation in a multiprocessor environment. *AIAA Paper* 91-1366, 1991.
8. Robinson CD, Harvey JK. Dynamic load balancing applied to a direct simulation Monte Carlo implementation. *Proceedings of the 5th International Parallel Computing Workshop*, Kawasaki, Japan, 1996.
9. Bird GA. Monte-Carlo simulation in an engineering context. In *Progress in Astronautics and Aeronautics*: *Rarefied Gas Dynamics*, Fisher SS (ed.), vol. 74, Part I. AIAA: New York, 1981; 239–255.
10. Borgnakke C, Larsen PS. Statistical collision models for Monte-Carlo simulation of polyatomic gas mixtures. *Journal of Computational Physics* 1975; **18**(3):405–420.
11. Bird GA. The perception of numerical methods in rarefied gas dynamics: In *Progress in Astronautics and Aeronautics*: *Rarefied Gas Dynamics*, Muntz EP, Weaver DP, Campbell DH (eds), vol. 118. AIAA: Washington, DC, 1989; 211–226.
12. Karypis G, Kumar V. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing* 1998; **48**(1):96–129.
13. Rothe DE. Electron-beam studies of viscous flow in supersonic nozzle. *AIAA Journal* 1971; **9**(5):804–811.
14. Moss JN, Price JM, Dogra VK, Hash DB. Comparison of DSMC and experimental results for hypersonic external flows. *AIAA Paper* 95-2028, 1995.
15. Moss JN, Olejniczak J. Shock-wave/boundary-layer interactions in hypersonic low density flows. *AIAA Paper* 98-2668, 1998.
16. Markelov GN, Kudryavtsev AN, Ivanov MS. Continuum and kinetic simulation of laminar separated flow at hypersonic speeds. *Journal of Spacecraft and Rockets* 2000; **37**(4):499–506.
17. Chanetz B. Study of axi-symmetric shock wave/boundary layer interaction in hypersonic laminar flow. *ONERA Technical Report* RT 42/4362 AN, Februrary 1995.

Copyright © 2004 John Wiley & Sons, Ltd.