48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2010, Orlando, Florida

Parallelization of a Modular Particle-Continuum Method for Hypersonic, Near Equilibrium Flows

Timothy R. Deschenes* and Iain D. Boyd†
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109

Extension of a modular particle-continuum (MPC) method is outlined to take advantage of available cluster computer technology and allow future extension to simulation of full three dimensional flow. This method loosely couples an existing direct simulation Monte Carlo (DSMC) code to a Navier-Stokes solver (CFD) while allowing both time-step and cell size to be completely decoupled between each method. This approach allows the solver to maintain the physical accuracy of DSMC in regions where the Navier-Stokes equations break down, while achieving the computational efficiency of CFD in regions that are considered fully continuum. Parallelization techniques that take advantage of the modular implementation are outlined and evaluated using a set of flow problems with various free stream conditions as test cases. In general, linear speedup is realized for small problems but with a slope less than unity, with no effect on the final solution. Overall, the scaled efficiency of larger computations on more processors remains above 80% for the test case examined. Comparisons between solution time requirements for full DSMC and the MPC are made. A moderate speedup of about five is attainable for the higher Knudsen number case, while a speedup of over thirty compared to the full DSMC time is realized for the near equilibrium case.

Nomenclature

d	Diameter $[m]$
k	Boltzmann constant $\left[1.38 \times 10^{-23} \frac{J}{K}\right]$
m	Molecular mass $[kg]$
n	Computational load
p	Number of processors
Q	Flow quantity
t	Wall clock time
T	Temperature $[K]$
u	Velocity $\left[\frac{m}{s}\right]$
η	Scaled efficiency
λ	Mean free path $[m]$
μ	Coefficient of viscosity $\left[\frac{kg}{m\,s}\right]$
Φ	Subrelaxation weight
ρ	Mass density $\left[\frac{kg}{m^3}\right]$
σ	Reference cross section $\left[m^2\right]$
τ	Relaxation time $[s]$
ω	VHS viscosity temperature exponent

Subscripts

CFD	Computational Fluid Dynamics

*Graduate Student, Student Member AIAA. Email: thytimo@umich.edu.
†Professor, Associate Fellow AIAA. Email: iainboyd@umich.edu.

$DSMC$	direct simulation Monte Carlo
ref	Reference
rot	Rotational
tra	Translational
∞	Free stream

I. Introduction

As a hypersonic vehicle traverses a planetary atmosphere, the variation of characteristic flow length and time scales about the vehicle can vary from very large to extremely small. One parameter used to compare the variation in length scales is the Knudsen number. At sufficiently low Knudsen numbers, many gas particle collisions occur around the body and the flow can be considered in collisional equilibrium. When the flow is considered near collisional equilibrium, continuum flow formulations such as the Navier-Stokes equations provide a physically accurate description of the flow field. Computational Fluid Dynamics (CFD) provides accurate and efficient numerical solutions of the Navier-Stokes equations. However, at high Knudsen numbers, the flow is rarefied, or in collisional nonequilibrium. In this regime, the approximations used to derive the Navier-Stokes equations break down and the flow can only be accurately described using a kinetic description. The direct simulation Monte Carlo (DSMC) method[1] first proposed by Bird provides an approach that can accurately model dilute gas flows described by the Boltzmann equation. Though physically accurate over all flow conditions, DSMC becomes prohibitively expensive at higher densities due to time and length scale restrictions set by separating the move and collision processes in the method. At these higher densities, the approximations used to form the Navier-Stokes equations are valid, and the flow can be computed using CFD. For many hypersonic flows of interest, some regions of the flow field, such as the shock, boundary layer, or wake, can be sufficiently rarefied so that CFD can not be used over the entire flow field while simultaneously having regions that can be considered well within the continuum regime making full DSMC simulations prohibitively expensive. Instead, a hybrid method can be employed that takes advantage of the computational efficiency of CFD in regions that are near collisional equilibrium, while maintaining the physical accuracy of DSMC in regions that are rarefied.

Previous work has been performed using a "zonally decoupled" DSMC-CFD simulation.[2–4] For these hybrid methods, a CFD solution is calculated to a predetermined interface, then this information is used as the boundary condition for the DSMC simulation over the remainder of the domain. This method is only valid when the rarefied region is completely down stream of the continuum region with no recirculation or subsonic flow occurring across the interface. This is because the CFD solution is completely decoupled from the DSMC solution. Often, rarefied regions are highly localized and two way coupling is required to obtain an efficient, physically accurate simulation. Various methods have been proposed that adaptively re-position the interface between DSMC and CFD throughout the simulation using different coupling methods between the two domains.[5–8] Reference 9 presents a discussion of the major considerations involved in coupling a kinetic scheme with a continuum method as well as a summary of published work on past methods.[9]

The present article extends the modular particle-continuum (MPC) method that was first developed for 1-D shock waves[10] and later extended to 2-D and axi-symmetric flows.[11,12] In previous work, the MPC method was able to reproduce the full DSMC results. By limiting the DSMC method to only regions that are rarefied, the MPC method can achieve speed-up factors exceeding 3 for transitional flows[13] and 13 for near continuum flows.[12] In addition, previous work was performed to increase the physical accuracy of the MPC method at higher enthalpies by including consistent models for rotational and vibrational nonequilibrium in both flow modules.[14,15]

All past work with the MPC method has been limited to 2-D and axi-symmetric flows due to the serial implementation of the method. In order to simulate larger, more complicated flows and extend the method to 3-D, the code must first be parallelized to reduce the processor memory requirements and take advantage of networked computing clusters. This article outlines the recent progress of parallelizing the MPC method with domain decomposition. First, the modular implementation of the MPC is outlined including details relevant to parallel implementation. Then details of preliminary parallel timing comparisons are outlined. Finally, the parallel MPC method is applied to a low Knudsen number hypersonic flow over a 2-D cylinder and compared with full DSMC and CFD flow field results.

II. DSMC and Navier-Stokes Modules

The modular particle-continuum (MPC) method is capable of simulating planar shock waves[10] and two dimensional and axi-symmetric flows.[11,12,16] For the continuum regions, it uses a CFD code called LeMANS, while a DSMC code called MONACO is used for rarefied regions. Very few modifications are made to each source code which allows the use of state of the art simulation methods that have been previously developed and verified. Instead, the focus is on the hybrid methodology, particularly where the interface between CFD and DSMC is located and when and how information is transferred between the two modules. In order to use full DSMC results as verification of the MPC method, care must be taken in order to ensure that the physical models used in the two flow solvers are consistent and provide the same results in regions where both solvers are valid. The rest of this section will outline the important physical models for the flow of interest employed in each flow solver, while the proceeding section summarizes the methodology used in the framework of the MPC method.

CFD Module

LeMANS is a laminar, hypersonic code that uses CFD methods to solve the Navier-Stokes equations that are modified to account for rotational and vibrational nonequilibrium. The Variable Hard Sphere (VHS) viscosity model, as seen in Eqs. 1 and 2, is used to be consistent with the DSMC module. For all simulations presented in this paper, diatomic nitrogen is used with a reference temperature, T_{ref}, of $273\,K$ and a reference diameter, d_{ref}, of $4.17 \times 10^{-10}\,m$. The power law exponent used is $\omega = 0.75$ while m is the molecular mass, k is the Boltzmann constant, and T_{tra} is the local translational temperature. The inviscid fluxes are solved in LeMANS using a modified version of the Steger-Warming flux vector splitting method. This method is less dissipative outside of the shock which is required to resolve the boundary layer, but switches back to the original form of the Steger-Warming fluxes within the shock. Viscous fluxes are calculated using both values at the cell centers and nodes. For this work, the method uses a point-implicit time-integration method. More details about the numerical implementation can be seen in Ref. 17.

$$\mu = \mu_{ref} \left(\frac{T_{tra}}{T_{ref}} \right)^{\omega} \tag{1}$$

$$\mu_{ref} = \frac{15\sqrt{\pi m k T_{ref}}}{2\pi d_{ref}^2 \left(5 - 2\omega\right)\left(7 - 2\omega\right)} \tag{2}$$

LeMANS is capable of simulating separate rotational or vibrational energy equations. The translation-rotation relaxation time is modeled using the product of Parker's model for rotational collision number and VHS equilibrium mean collision time. The translation-vibration relaxation time is modeled using the sum of the of Landau-Teller approximation with coefficients from Millikan and White[18] and Park's phenomenological high temperature correction.[19] For the cases presented in this paper, the free stream enthalpy is sufficiently small such that vibrational activation can be ignored with little effect on flow results, so only rotational nonequilibrium is activated in the simulations presented in this paper.

DSMC Module

MONACO is a general, cell-based implementation of the DSMC method capable of simulating 2-D, 3-D, or axially symmetric flows. Modules that model rotational, vibrational, or chemical nonequilibrium of an arbitrary number of species are also available. The VHS collision model that replicates the macroscopic viscosity-temperature dependence under equilibrium conditions which is shown in Eq. 1 is used.

Rotational relaxation is modeled using the phenomenological variable rotational energy exchange probability model of Boyd,[20] that is derived from Parker.[21] The Lumpkin correction parameter is applied to the instantaneous probability to resolve the differences in definition of continuum and particle relaxation times.[22]

Vibration-translation relaxation is modeled using a phenomenological, cell-temperature based probability model that is consistent with the equilibrium vibrational relaxation time. This method applies an average probability of a vibrational relaxation process for each collision across the cell. More information can be found in Ref. 15. A correction parameter suggested by Gimelshein, et al. is applied to resolve the differences in the definition of the relaxation time between DSMC and CFD for quantized energy distributions, such as

the harmonic oscillator approximation used in MONACO.[23] Again, for the proceeding simulations, activation of vibrational energy is ignored.

III. Modular and Parallel Implementation

In order to parallelize the MPC method in an efficient manner, the modular implementation and data structure must be taken into account. Both modules used in the MPC method are parallelized using message passing interface (MPI), so work on parallelizing the MPC method is restricted to providing cell load information consistent with the module being called next so that dynamic domain decomposition can provide each processor with comparable segments of computational load.

Modular Implementation

An outline of the MPC method is as follows:

1. Load grid-independent full Navier-Stokes solution on a structured mesh. Use continuum breakdown parameter (Eq. 3) to set up initial interface locations. Create overlap region into initial continuum domain. Create DSMC grid by refining CFD grid to meet local cell size restrictions using the initial continuum result. Generate particles in the DSMC domain.

2. In particle boundary cells, destroy all old particles and create new DSMC particles based on Navier-Stokes information in corresponding cells. Sample particle velocities from Chapman-Enskog velocity distribution and particle internal energies from Boltzmann energy distribution functions. Cycle through DSMC solver for one time-step. Update hybrid macroscopic quantities using subrelaxation parameter (Eq. 4). Repeat prescribed number of times.

3. Re-evaluate breakdown parameter; if needed, move interfaces, create particles in new DSMC cells, and destroy particles in newly tagged pure continuum cells.
   IF interfaces have significantly changed go to step 2.
   ELSE update the Navier-Stokes boundary cells with subrelaxation average and continue.

4. Cycle through Navier-Stokes solver. Repeat until significantly converged.

5. Re-evaluate breakdown parameter; if needed, move interfaces, create particles in new DSMC cells, and destroy particles in newly tagged pure continuum cells.
   IF interfaces have significantly changed or steady-state has not been reached, go to step 2.
   ELSE continue.

6. Remove overlap regions and delete particles in overlap region.

7. In particle boundary cells, destroy all old particles and create new DSMC particles based on Navier-Stokes information in cells. Cycle through DSMC solver for a prescribed number of iterations and collect samples.

8. Update Navier-Stokes boundary cells and further converge continuum region.
   IF DSMC statistical scatter and Navier-Stokes residual are acceptable, end.
   ELSE return to step 7.

Equation 3 shows the breakdown parameter used in the MPC method where λ is the local mean free path and Q is some flow quantity of interest, such as temperature, density, or velocity magnitude. Equation 4 shows the subrelaxation average used to reduce the statistical scatter in the flow variables applied to Navier-Stokes boundary cells from DSMC information where $\bar{Q}_{j-1}$ is the average value at the previous iteration, Q_j is the value at the current iteration, and Φ is the subrelaxation parameter. The MPC code uses a subrelaxation value of $\Phi = 1. \times 10^{-3}$ for simulations presented in this paper.

$$Kn_{GL-Q} = \lambda \left| \frac{\nabla Q}{Q} \right| \tag{3}$$

$$\bar{Q}_j = (1 - \Phi)\, \bar{Q}_{j-1} + \Phi Q_j \tag{4}$$

A separate, structured DSMC mesh is created by refining the CFD grid to meet DSMC cell size restrictions. Due to the nature of creating the DSMC mesh by refinement of the CFD mesh, every DSMC cell is entirely physically located in one CFD cell. In addition, every CFD cell corresponds to a unique number of DSMC cells that cover the same area for the 2-D grid. Both flow modules are virtually unmodified and use their native data structure. Instead, hybrid functions and data structures are used to pass information between modules and perform miscellaneous hybrid routines, such as moving the rarefied-continuum interface location. Figure 1 shows a schematic of the data structure and modular implementation of the MPC method. The LeMANS data structure still operates over cells with each cell containing geometry and flow property data. In a similar manner, the MONACO data structure based on cells that contain geometry and pointers to each particle data structure. The MPC data structure is built to link to two existing data structures. Each CFD array element has a corresponding hybrid array element that contains all information (pointers) to the DSMC cells that are physically contained within the CFD cell as well as required hybrid information such as the cell type (particle, continuum or both in overlap). In addition, there is a look up array (not shown) to cross reference the CFD (and hybrid) cell index given a DSMC cell index. These data structures allow complete coupling between the two methods. During a typical hybrid loop, both CFD and DSMC information is needed by the hybrid functions that generate and destroy particles at the boundaries, evaluate the breakdown parameter and move interface locations, and track macroscopic changes in the rarefied region.

Because of the modular implementation, updates to either the CFD or DSMC modules can be performed and changes in code are confined to just the 18 hybrid functions used to couple the two flow modules. This significantly reduces the time required to include new developments in the hybrid code, such as the recent inclusion of a separate rotational energy equation in the continuum solver.

Parallel Implementation

Both MONACO and LeMANS are parallelized as separate codes using message passing interface (MPI). Due to the modular nature of the MPC method, all parallelized routines that apply to the module data structure can be used without modification. Instead, separate functions are provided to perform dynamic domain decomposition using load balance information consistent with the hybrid framework. Once the graph is partitioned, the hybrid functions provide each module with the partition information so that each module can perform its normal partitioning routine and renumber its local data on each node to be consistent with its parallel routines. Finally, hybrid functions reorder the local hybrid data structure to be consistent with the local MONACO and LeMANS data structures.

As earlier mentioned, during each time-step hybrid functions access data from both the DSMC and CFD data structures. In order to reduce inter-processor communication at each time-step, all data for one CFD cell is be kept on the same local processor. This leads to partition cuts of data along horizontal lines in the schematic of the data structure shown in Fig. 1. In turn, this leads to all DSMC cells located in one CFD cell to be located on the same processor. Partitioning can then be applied directly to the hybrid data structure (which is mapped one to one to the CFD data structure) and both modules can partition to remain consistent with the partitioned hybrid information. In addition, due to the loosely coupled nature, each module is called for multiple iterations, before calling the next module. This allows for dynamic domain decomposition to be performed using load information consistent only with the next solver that is called. This leads to a change in the following MPC method steps to perform dynamic domain decomposition:

3. Re-evalute breakdown parameter; if needed, move interfaces, create particles in new DSMC cells, and destroy particles in newly tagged pure continuum cells.
   IF interfaces have significantly changed, **repartition using DSMC cell load information** and go to step 2.
   ELSE **repartition using CFD cell load information** and update the Navier-Stokes boundary cells with subrelaxation average and continue.

5. Re-evalute breakdown parameter; if needed, move interfaces, create particles in new DSMC cells, and destroy particles in newly tagged pure continuum cells.
   IF interfaces have significantly changed or steady-state has not been reached, **repartition using DSMC cell load information** and go to step 2.
   ELSE continue.

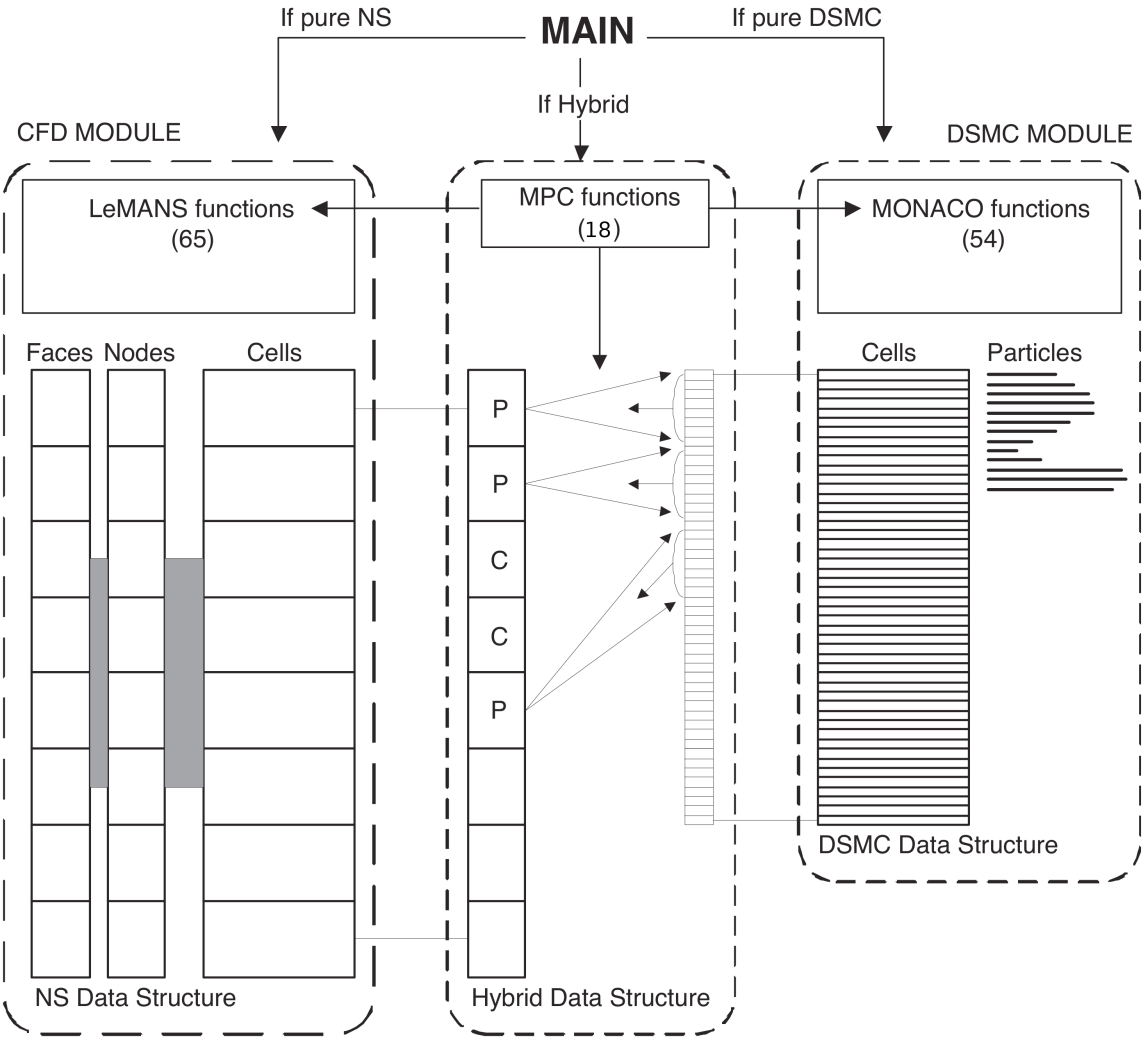


Figure 1: Modular implementation of the source code and data structures.[13]

6. Remove overlap regions, delete particles in overlap region, and **repartition using DSMC cell load information.**

8. **Repartition using CFD cell load information,** update Navier-Stokes boundary cells, and further converge continuum region.
   IF DSMC statistical scatter and Navier-Stokes residual are acceptable, end.
   ELSE **repartition using DSMC cell load information** and return to step 7.

The domain decomposition is performed using METIS[24] with cell load information provided by the hybrid and module data structures. Figure 2 shows a schematic of the mesh structures used in the two modules during the initial unsteady portion of the MPC loop at the *IF* statement of step 3. Since MONACO will be the next solver called, DSMC cell load information will be provided to METIS along with the hybrid data structure which is the same as the CFD data structure. In this case, the red cells will have a cell load proportional to the total number of DSMC particles in the cell, while blue cells have a cell load of zero. In contrast, Fig. 3 shows a schematic of the cell load information used for dynamic domain decomposition within the *ELSE* statement of step 3. Now CFD cells in the continuum and overlap region (red) are given a cell load of unity, while cells located in the rarefied region are given a cell load of zero since they are skipped in the CFD computation.

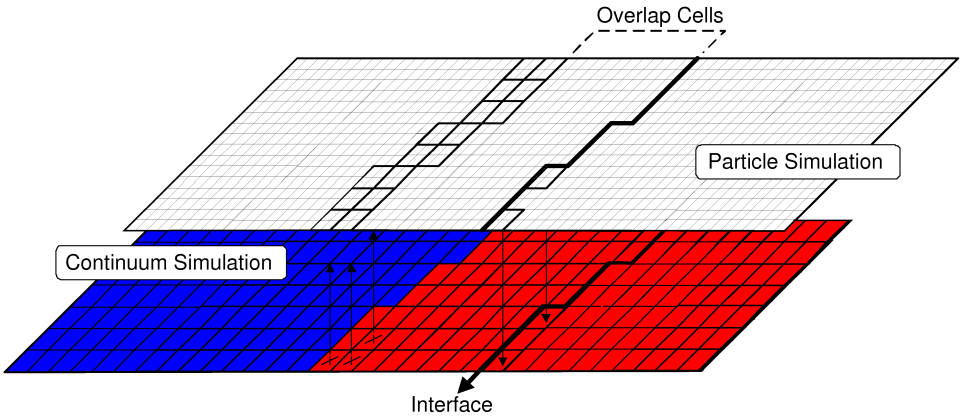


Figure 2: Schematic of cell load estimation supplied to METIS before the DSMC module is called. Red cells have a computational load proportional to the DSMC particles located in the cell while blue cells are given zero computational load.

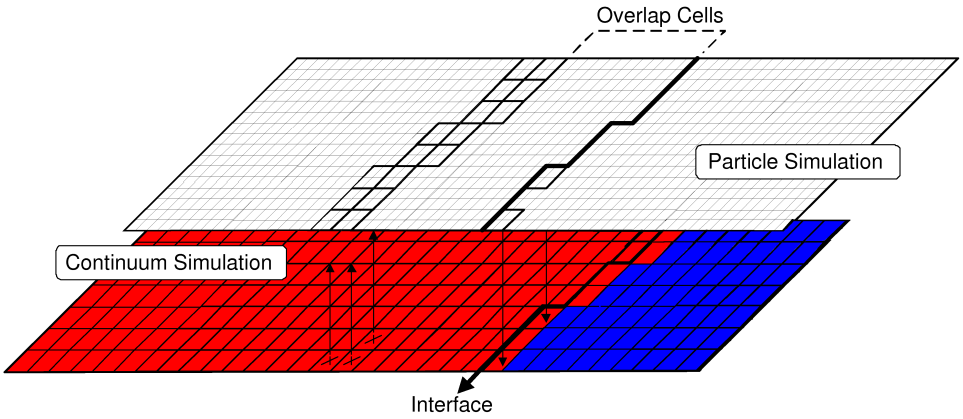


Figure 3: Schematic of cell load estimation supplied to METIS before the CFD module is called. Red cells have a computational load proportional to unity while blue cells are given zero computational load.

IV. Results

Parallel Performance

Parallel performance studies are performed using two previously published cases which are summarized in Table 1 along with references where more information on the flow conditions can be found. These two test cases are chosen to cover the wide range of applicable flow conditions relevant to the MPC method. Figure 4 shows the final DSMC-CFD interface locations for the two cases. The rarefied region for the higher Knudsen number case covers the majority of the domain, while the rarefied region for the lower Knudsen number case is isolated to the boundary layer and near wake. It should be noted that though the strong bow shock in the low Knudsen number case does demonstrate nonequilibrium effects, CFD can still accurately model the jump across the shock and incorrect modeling of the shock interior has a negligible effect on the flow field and surface quantities.[12,13] Despite the rarefied region being isolated to a small region in the boundary layer and near wake for the low Knudsen number case, the majority of the total computation is spent in the DSMC module.

Table 1: Summary of cases used for parallel performance study

Case	M_∞	Kn_∞
2D Cylinder $(M15Kn01)$[15]	15	0.01
2D Cylinder $(M12Kn002)$[13]	12	0.002

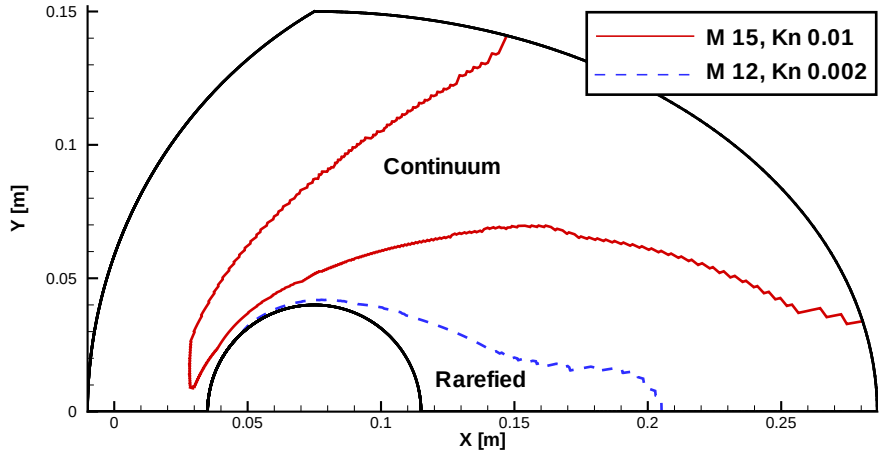


Figure 4: Final rarefied-continuum interface location for the two test cases.

Figure 5 shows a plot of the 32 processor domain boundaries optimized for load balancing for the DSMC module. All partitions are clustered in the near wake region where the DSMC particles are located while the entire continuum region is appended to one partition. In addition, translational temperature contours are displayed. No jumps are visible across processor or rarefied-continuum interface locations which demonstrate that the correct information transfer routines in both flow libraries are still working as intended.

Figure 6 shows a plot of the parallel speed up for the two cases examined for both the unsteady portion and the steady state portion of the MPC simulations. Both cases show a higher speedup during the steady portion compared to the unsteady portion. This is due to increased variation in the total number of simulation particles during the unsteady portion of the computation which necessitates the overhead of more dynamic domain decomposition calls. At steady state, the total number of particles on each processor remains nearly constant. The speedup for all cases begins to deviate from the ideal speedup as the number of processors increase. Despite this deviation, near linear speedup is still attained. In addition, the higher Knudsen number case achieves a higher speedup in nearly all tests due to the larger rarefied region which increases the total number of particles, and therefore computational load, required.

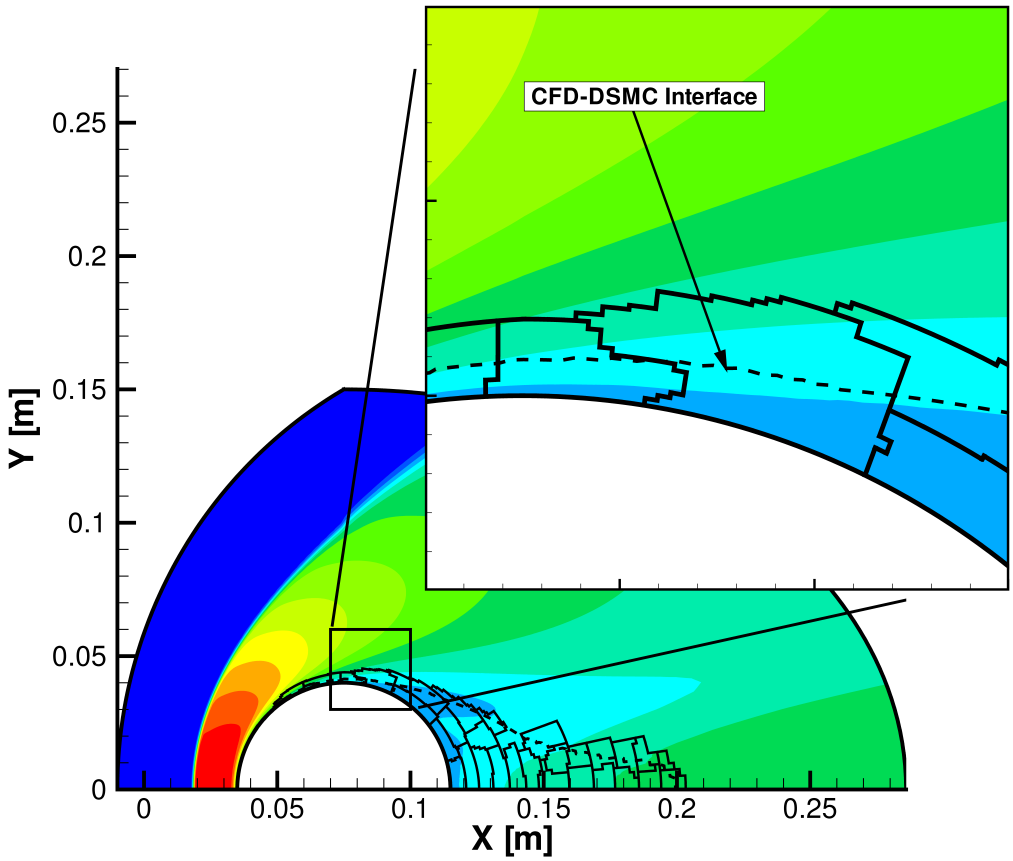


Figure 5: Temperature contours and processor domain boundaries for a DSMC module call during steady state.

It should be noted that the two cases used for this study do not require a large computational expense and the amount of time spent on performing the parallel portion of the computation significantly decreases as the number of processors increases. In addition, the main goal of the parallelization of the code is to solve larger problems in about the same amount of time. Larger computational expense jobs, such as three dimensional simulations, will display a speedup curve that stays closer to the ideal curve longer and may not even be feasible to perform on one processor due to memory constraints. This effect can be seen in Fig. 7 where the computational expense of the lower Knudsen number case is increased by increasing the number of DSMC simulation particles which proportionally increases the total computational load. Using this information, a scaled efficiency can be estimated. The scaled efficiency is defined in Eq. 5 where $\eta_S\,(pn, p)$ is the scaled efficiency of doing pn computational work on p processors and $t(pn, p)$ is the wall time required to do perform pn computational work on p processors. For these cases, the wall times are normalized with the wall time required for an 8 processor (1 full node) computational case. Doubling and quadrupling the computational load and resources require less than 7% and 17% more wall time, respectively.

$$\eta_S\,(n, p) = \frac{t(8n, 8)}{t(pn, p)} \tag{5}$$

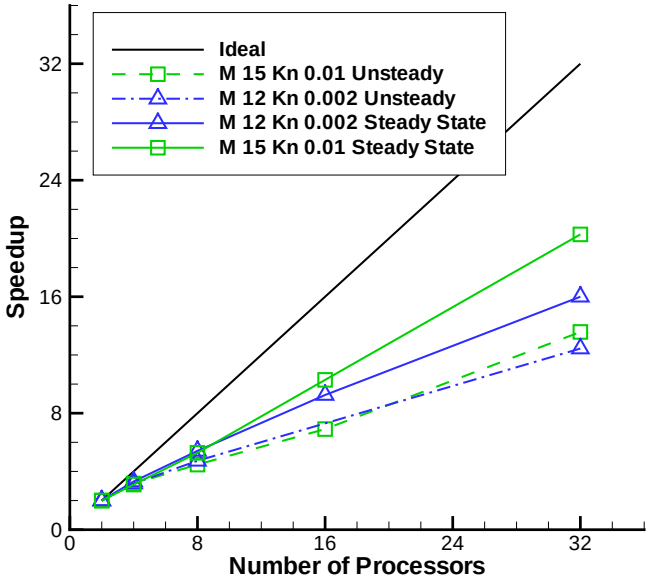


Figure 6: Parallel speedup for the two flow conditions examined (fixed problem size).

Flow field comparisons

Ultimately, the goal of the MPC method is to reproduce full DSMC results using a fraction of the CPU time. The next subsection will compare flow field results predicted by full DSMC, full CFD and the MPC method for the lower Knudsen number test case. Figures 8(a) and 8(b), respectively, compare the full CFD and MPC predictions of velocity magnitude contours and streamlines with full DSMC results. In general, the MPC provides little improvement over full CFD in the dense fore-body region and larger improvement in the near wake region where the flow is in continuum breakdown. The full CFD result shows slightly better agreement with full DSMC near the outflow domain, but all three simulation results are within 2% of each other in this region and the large difference in contour placement is mainly due to very small gradients. In addition, the MPC method shows a large improvement over full CFD results in predicting the size and shape of the recirculation zone in the near wake region. Again, this is directly due to continuum breakdown and the Navier-Stokes equations can not correctly model the physics of the flow in this region. In the MPC

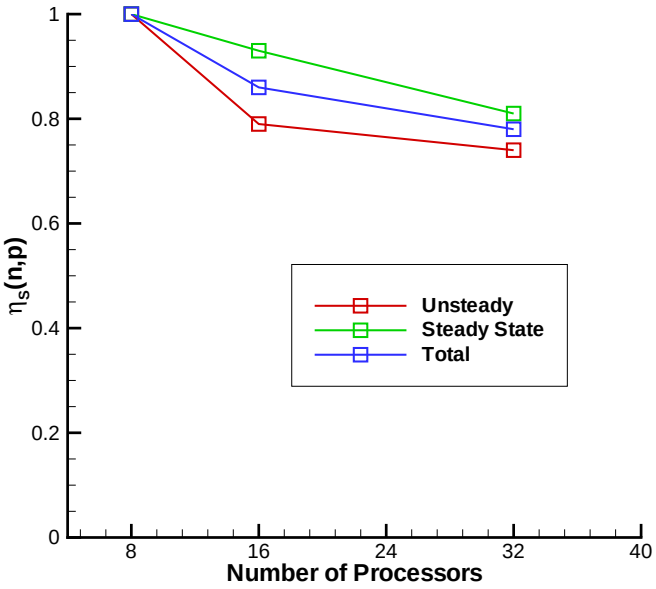


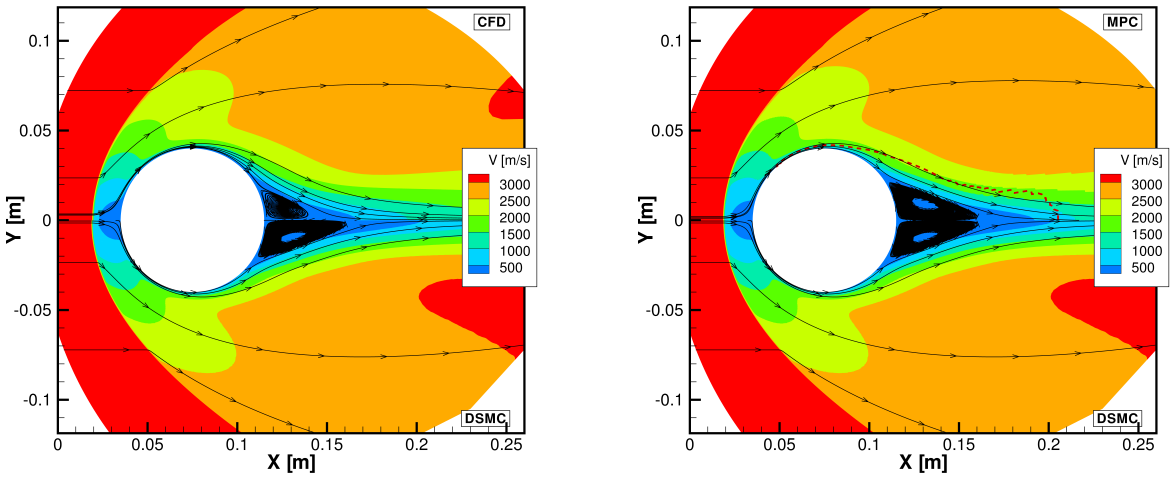
Figure 7: Scaled efficiency of the MPC method (problem size scaled with number of processors).

simulation, this region is marked as rarefied, so the DSMC module, which can accurately model the rarefied flow physics, is applied.

Figures 9(a) and 9(b) compare full DSMC, full CFD, and the MPC method predictions of translational and rotational temperature, respectively. Again, the MPC method is able to improve the CFD prediction in the near wake region by applying the DSMC solver in regions where the Navier-Stokes equations break down with a maximum. The largest observed difference in flow field properties between the MPC and full DSMC results is about 4%. In addition, the MPC results show a small improvement in the agreement of contour placement in the continuum region as well. This is a direct effect of the hybrid coupling providing better boundary information to the CFD solver along the continuum-rarefied interface.

Computational Performance

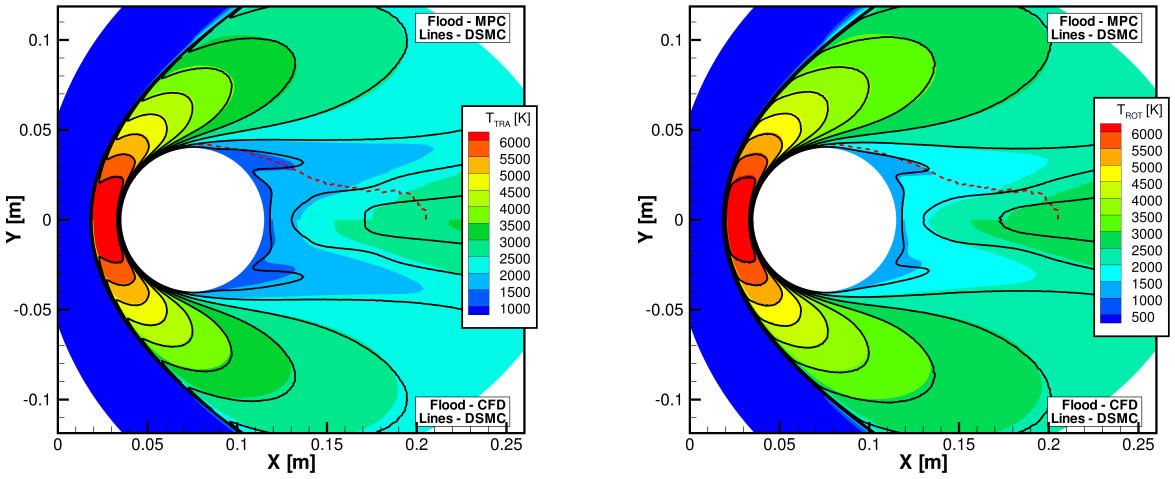
Table 2 summarizes the computational performance of the MPC method. The data for the higher Knudsen number case was taken from Ref. 15. The actual speedup is calculated by taking the ratio of the total CPU time to calculate the full DSMC result to total CPU time to calculate the MPC result. In addition, an estimate of the ideal speedup is included as well, where the ideal speedup is the final total particle ratio between full DSMC and MPC simulations. For fair comparison, both the full DSMC and MPC simulations are performed on the same computer architecture using the same compiler and compiler options. In addition, for the lower Knudsen number case, the parallel implementation of the MPC method is applied and the domain split so that the number of particles per processor is the same in the MPC and full DSMC simulation. This results in the MPC simulation being computed on 4 processors, while the full DSMC simulation is computed on 104 processors. In addition, constant numerical particle weighting, time-step, and number of sample steps are used in both DSMC modules. It is clear that the MPC method shows a much higher increase in performance for the near continuum simulation. This is a direct effect of significantly reducing the total number of simulation particles required. In addition, despite significant overhead for the parallel implementation, the MPC simulation still achieves near ideal speedup. This is due, in part, from a reduction in the number of time-steps required to reach steady state in the MPC method. Since the MPC method starts with a Navier-Stokes result that is nearly correct everywhere, the number of time-steps required to reach steady state is greatly reduced compared to the full DSMC simulation. A further increase in the MPC speedup could be realized for the low Knudsen number case if the DSMC time-step is increased to be consistent with the relevant mean collision time present in the MPC-DSMC domain. This time-step would



(a) CFD (top) and DSMC (bottom)

(b) MPC (top) and DSMC (bottom)

Figure 8: Comparison of density contours and streamlines predicted by full DSMC, full CFD, and the MPC method for Mach 12 flow over a cylinder with a global Knudsen number of 0.002.



(a) Translational Temperature

(b) Rotational Temperature

Figure 9: Comparison of translational and rotational temperature contours predicted by full CFD, full DSMC, and the MPC method for Mach 12 flow over a cylinder with a global Knudsen number of 0.002.

be larger than the corresponding full DSMC simulation due to the elimination of the stagnation region which contains the smallest mean collision time. Previous comparison of the required MPC and full DSMC run times for the same flow conditions of the low Knudsen number case showed a speedup of 10.6. The increase in performance of the current MPC simulation compared to full DSMC is due to a reduction of the size of the DSMC region that is used in the MPC simulation. This is possible with the larger range of applicability of the continuum solver due to the implementation of rotational nonequilibrium.

Table 2: Computational performance of the MPC method.

Case	Actual Speedup	Ideal Speedup	Memory Usage
M 15, Kn 0.01	4.5	2.6	22%
M 12, Kn 0.002	31.4	31.4	21%

V. Conclusion

A parallel implementation of the modular particle-continuum (MPC) method was described and tested on two cases of near-equilibrium, hypersonic flow over two dimensional cylinders. Although far from ideal, significant speedup was attained using the parallel implementation. In addition, it was shown that the parallel performance scales well as the problem size increases. Flow field and CPU time requirements were compared with full CFD and DSMC results. It was found that the MPC method can improve on the physical accuracy of the initial CFD solution to be in very good agreement with full DSMC results. This is directly due to applying the DSMC module in regions where the Navier-Stokes equations break down. Finally, it was found that higher speedup factors can be achieved for lower Knudsen number flows. This is due to significantly reducing the number of simulation particles required and the time required for the flow to reach steady state. It should be noted that both the full DSMC and MPC-DSMC module used constant cell weight and time-step. It is expected that variable cell time-step and particle weighting in the full DSMC simulation will reduce the required CPU time, but would have a similar, though smaller, effect on the required MPC CPU time.

Future work is aimed at upgrading the DSMC module used in the MPC method. Though the DSMC module will remain nearly intact, change to the hybrid functions will be required to accommodate new capabilities. A newly upgraded DSMC module will allow varying cell weight and time-step in the DSMC region that will further reduce the computational requirements of the MPC method. In addition, extension to full 3-D simulation capability will further increase the applicability of the MPC method to solve more realistic near-continuum, hypersonic flow problems.

Acknowledgments

The authors gratefully acknowledge funding from NASA Grant NCC3-989 and the AFRL Collaborative Center in Aeronautical Sciences. Computational resources from NASA Advanced Supercomputing Division, where these simulations were performed, is greatly appreciated.

References

[1]Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, 1994.

[2]Glass, C. E. and Gnoffo, P. A., "A 3-D Coupled CFD-DSMC Solution Method With Application to the Mars Sample Return Orbiter," Tech. Rep. TM-2000-210322, NASA, July 2000.

[3]Glass, C. E. and Gnoffo, P. A., "Comparison of a 3-D CFD-DSMC Soultion Methodology With a Wind Tunnel Experiment," Tech. Rep. TM-2002-211777, NASA, August 2002.

[4]Wilmoth, R. G., Mitcheltree, R. A., Moss, J. N., and K., D. V., "Zonally Decoupled Direct Simulation Monte Carlo Solutions of Hypersonic Blunt-Body Flows," *Journal of Spacecraft and Rockets*, Vol. 31, No. 6, 1994, pp. 971–979.

[5]Hash, D. B. and Hassan, H. A., "Assessment of Schemes for Coupling Monte Carlo and Navier-Stokes Soultion Methods," *Journal of Thermophysics and Heat Transfer*, Vol. 10, No. 2, 1996, pp. 242–249.

[6]Hash, D. B. and Hassan, H. A., "A Decoupled DSMC/Navier-Stokes Analysis of a Transitional Flow Experiment," *Collection of Technical Papers - 34th AIAA Aerospace Sciences Meeting and Exhibit*, Vol. 1, Reno, NV, 1996.

[7]Roveda, R., B., G. D., and Varghese, P. L., "Hybrid Euler/Particle Approach for Continuum/Rarefied Flows," *Journal of Spacecraft and Rockets*, Vol. 35, No. 3, 1998, pp. 258–265.

[8]Roveda, R., B., G. D., and Varghese, P. L., "Hybrid Euler/Direct Simulation Monte Carlo of Unsteady Slit Flow," *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 753–760.

[9]Wijesinghe, H. S. and G., H. N., "A discussion of Hybrid Atomistic-Continuum Methods for Multiscale Hydrodynamics," *International Journal for Multiscale Computational Engineering*, Vol. 2, 2004.

[10]Schwartzentruber, T. E. and Boyd, I. D., "A hybrid particle-continuum method applied to shock waves," *Journal of Computational Physics*, Vol. 215, No. 2, July 2006, pp. 402–416.

[11]Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "Hybrid particle-continuum simulations of nonequilibrium hypersonic blunt-body flowfields," *Journal of Thermophysics and Heat Transfer*, Vol. 22, No. 1, 2008, pp. 29–37.

[12]Schwartzentruber, Thomas E.and Scalabrin, L. C. and Boyd, I. D., "Multiscale Particle-Continuum Simulations of Low Knudsen Number Hypersonic Flow Over a Planetary Probe," *Journal of Spacecraft and Rockets*, Vol. 45, No. 6, 2008, pp. 1196–1206.

[13]Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "Modular implementation of a hybrid DSMC-NS algorithm for hypersonic non-equilibrium flows," AIAA-2007-613.

[14]Deschenes, T. R., Boyd, I. D., and Schwartzentruber, T. E., "Incorporating Vibrational Excitation in a Hybrid Particle-Continuum Method," Vol. AIAA 2008-4106, 2008.

[15]Deschenes, T. R., Holman, T. D., Boyd, I. D., and Schwartzentruber, T. E., "Analysis of Internal Energy Transfer Withing a Modular Particle-Continuum Method," Vol. AIAA 2009-1213, 2009.

[16]Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "A modular particle-continuum numerical method for hypersonic non-equilibrium gas flows," *Journal of Computational Physics*, Vol. 225, No. 1, July 2007, pp. 1159–1174.

[17]Scalabrin, L. C. and Boyd, I. D., "Development of an Unstructured Navier-Stokes Solver for Hypersonic Nonequilibrium Aerothermodynamics," AIAA-2005-5203.

[18]Millikan, R. C. and White, D. R., "Systematics of Vibrational Relaxation," *The Journal of Chemical Physics*, Vol. 39, No. 12, December 1963, pp. 3209–3213.

[19]Park, C., *Nonequilibrium Hypersonic Aerothermodynamics*, John Wiley & Sons, 1990.

[20]Boyd, I. D., "Analysis of rotational nonequilibrium in standing shock waves of nitrogen," *AIAA Journal*, Vol. 28, No. 11, 1990, pp. 1997–1999.

[21]Parker, J. G., "Rotational and Vibrational Relaxation in Diatomic Gases," *Physics of Fluids*, Vol. 2, No. 4, July 1959, pp. 449–462.

[22]Lumpkin III, F. E., Haas, B. L., and Boyd, I. D., "Resolution of differences between collision number definitions in particle and continuum simulations," *Physics of Fluids A*, Vol. 3, No. 9, Sept. 1991, pp. 2282–2284.

[23]Gimelshein, N. E., Gimelshein, S. F., and Levin, D. A., "Vibrational Relaxation Rates in the Direct Simulation Monte Carlo Method," *Physics of Fluids*, Vol. 14, No. 12, 2002.

[24]Karypis, G. and Kumar, V., "METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices," *University of Minnesota, MN*, 1998.