PARALLEL PARTICLE SIMULATIONS OF LOW-DENSITY FLUID FLOWS

Steve Plimpton and Tim Bartel
Parallel Computational Sciences and Thermophysics Departments
Sandia National Laboratories*
Albuquerque, NM 87185

sjplimp@cs.sandia.gov tjbarte@cfd.sandia.gov

ABSTRACT

Direct Simulation Monte Carlo (DSMC) is a simulation technique that uses particles to model fluids at low densities where continuum formulations break down. Monte Carlo rules are used to perform particle collisions and chemical reactions and a grid is used to locate nearby particles. We describe our implementation of a large-scale DSMC model on the thousand-processor nCUBE 2 and Intel Paragon machines. It includes body-fitted irregular grids and particle weighting to reduce the numbers of particles and grid cells that must be used. Our parallel methods for load-balancing and performing irregular communication on the distributed-memory machines are also discussed. The parallel machines have enabled us to run large simulations (10^7 particles, 10^4 grid cells, 10^4 timesteps) at speeds up to 40 times faster (on our nCUBE 2) than on a Cray Y-MP processor. For illustration purposes, we include results from some of these large-scale simulations.

INTRODUCTION

When modeling fluid flow at very low densities, the continuum Navier-Stokes equations can become inaccurate. This is because the linear transport terms for mass diffusion, viscosity, and thermal conductivity in the partial differential equations are no longer valid when the distance particles move between collisions is comparable to typical length scales in the flow geometry. This is a characteristic of flows with a Knudsen number (ratio of mean-free path to length scale) on the order of 0.2. A commonly used method for modeling such flows is via particle simulation on a grid, a technique known as Direct Simulation Monte Carlo (DSMC), pioneered by Graeme Bird (Bird, 1976) . It has utility in aerospace applications (flow around "space"-planes, re-entry vehicles, and rocket plumes) and in vacuum chamber technologies for the semiconductor industry (plasma etching and chemical vapor deposition).

The key concept in DSMC is that particle motion is decoupled from particle collisions. During a timestep, particles flow through the simulation geometry independent of each other. At the end of the timestep a grid is used to locate nearby particles, and collisions between pairs of particles are performed via Monte Carlo rules to produce a new velocity distribution. An attractive feature of DSMC is that gas-phase and surface chemistry reactions can be performed as particles of different species collide with each other and with objects in the simulation domain.

In practice, DSMC simulations are computationally intense because of the large numbers of particles and grid cells used, and the long timescales often necessary for statistical averaging. Large-scale DSMC computations on current-generation supercomputers can require on the order of 10^7 particles, 10^6 grid cells, and 10^5 timesteps (though not necessarily all at the same time), even for two-dimensional simulations. Implementing a model on a parallel computer poses additional load-balancing challenges since the computational work per grid cell is proportional to the particle density, which can vary by orders of magnitude both spatially and temporally. In this article, we describe a 2-D parallel DSMC simulation we have developed for our nCUBE 2 and Intel Paragon machines. Descriptions of other DSMC models on MIMD and SIMD parallel machines are given in (MacDonald, 1991) and (Dagum, 1991). They differ from the model described here in that they employ regular grids and uniform particle weighting. The latter work is also a 3-D implementation. Fuller descriptions of earlier versions of our model have been given elsewhere (Plimpton, 1992; Bartel, 1992); here we focus on results we have generated using the production-level code. It is currently being used extensively on Sandia's parallel machines by a growing community of Sandians and external partners. The reason for this interest is that the parallel implementation of the DSMC method has allowed us to model much more complex flows than we could previously. In the next two sections, we briefly describe the features of the parallel code and give three examples of the kinds of systems we are studying with it.

PARALLEL IMPLEMENTATION.

To enable modeling of complex geometries, we use body-fitted, multi-block grids as illustrated in the top portion of Figure 1. The sub-grid in each block the grid is topologi-

*This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

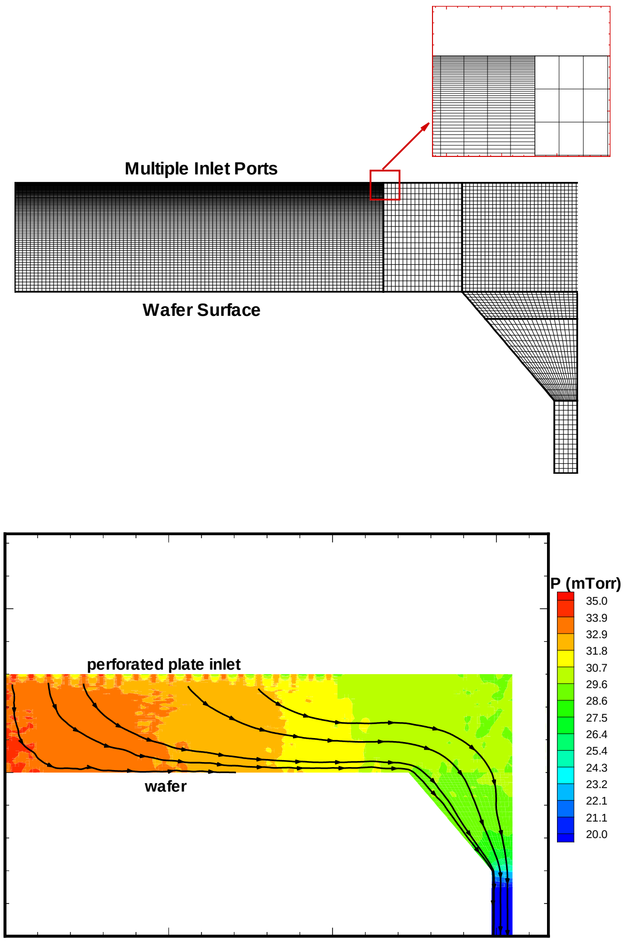

Figure 1: (Top) A multi-block grid for a plasma etch reactor. Within sub-blocks the grid is topologically uniform. (Bottom) Gas flow in the reactor chamber as it enters throught sonic nozzles at the top, flows across the silicon wafer surface, and exits at the bottom right. The pressure profile of the gas is shown.

cally regular and described by analytic equations for arcs and lines. Because the grid is only used to group nearby particles, there is no requirement for continuity between sub-grids in adjacent blocks. There are two principal advantages of this formulation. First, every processor can store the description of the entire grid in a compact form. This allows any processor to track the motion of a particle wherever it moves in the domain. More importantly, we can tailor the grid cells to be very small or large in different areas of the simulation domain to match the density profile anticipated in a particular flow. This provides resolution where it is needed and dramatically reduces the numbers of particles required in the simulation. Another feature of the code, cell-wise particle weighting, also reduces the computational load. Particles are created or destroyed as they move between cells with different weights. This method keeps the simulation from having too few particles in some cells (inadequate collision statistics) or too many in others (too much computational work). The code adjusts the weights as the flow evolves

Within this framework, a natural decomposition of the computational work on a distributed-memory parallel machine is to have each processor own a group of grid cells along with the particles inside them. At each timestep all processors can perform the collision and chemistry computations within their cells and adjacent surfaces in parallel. The MIMD nature of our parallel machines allows complex physics models for gas and surface chemistry and non-equilibrium thermodynamics to be implemented straightforwardly since the parallelism is at the grid cell level. This decomposition also allows all the particles to be moved independently and allows processors to find the new grid cell each particle ends up in. A general example of the particle movement a processor might track in a timestep, complete with particle creation and destruction, is shown in Figure 2. At the end of the timestep interprocessor communication is performed to pass particles to processors that own the new cells. The communication pattern changes from step to step because of the irregular nature of the grid and the long distances particles can potentially move. This irregularity is accounted for in the parallel code by a pre-communication step where each processor decides who it needs to send particles to, shares this information with those processors, and acquires similar information from the processors is will be receiving particles from. It then sends all its particles out (as buffered messages) and waits to receive all its incoming messages (particles).

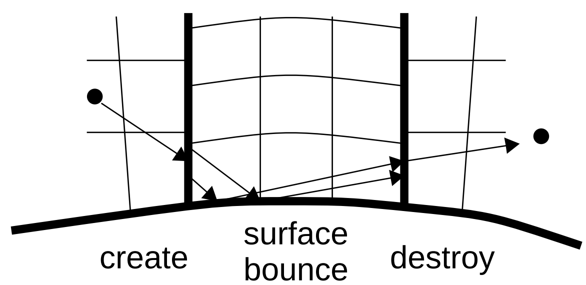

Figure 2: Particle movement through the multi-block grid. The particle is cloned as it enters a region with lower weighting factors and bounces off a surface. One of the particles is destroyed as it moves into a third region of cells with higher weighting factors.

To minimize these communication costs, a decomposition that assigned each processor a cluster of adjoining grid cells would be optimal. However, the particle density variations mentioned above make the computations highly unbalanced in such a scheme. Without a method for shifting work rapidly to idle processors as the simulation evolves (which incurs its own overhead), we have found clustered partitioning does not yield high performance on our DSMC problems. While we have not implemented such a dynamic load balancer, we have had good success with a static scattered decomposition of grid cells to processors; each processor is assigned an equal number of cells at the beginning of the simulation which are scattered across the simulation domain. On average each processor owns some cells in high-density regions and some in low-density regions so that each has roughly equal work to do. Such a scheme is simple to implement; it's only penalty is that communication costs are high since each processor has many neighbors. In large-scale simulations we typically pay a 20-40% overhead cost for this extra communication.

Many of the computations in a DSMC simulation do not vectorize easily on conventional supercomputers, primarily because of the multiple levels of branching inherent in many of the physics and Monte Carlo routines. This is not an obstacle to exploiting parallelism on machines like the nCUBE 2 and the Intel Paragon, however. On large problems, our DSMC code runs from 20-40 times faster on our nCUBE 2 than it does on a Cray Y-MP processor, where it runs only a few times faster than on a RISC workstation. Preliminary results on our new Intel Paragon indicate it is about 4-5 times faster on a per processor basis than the nCUBE for our DSMC simulation. A second benefit of the parallel machines for the problems we have run has been their large memory. For example, the nCUBE 2 with its 4 Gbytes of memory can easily run simulations with many tens of millions of particles, where each particle has a dozen or more variables associated with it.

The results from three of our DSMC simulations are shown in the next few figures. Figure 3 shows a computation performed with NASA collaborators to simulate flow around the Wake Shield Facility. This is a NASA experiment which is scheduled to fly on the space shuttle later this year. A 3.6-meter diameter disk will be positioned in low earth orbit by the shuttle arm so as to create a wake region of very low density behind it where high-vacuum materials processing or manufacturing processes could potentially be carried out. We used the DSMC model to compute the flow and density variations around an idealized shield. This was a large simulation (a few million particles, 10,000 grid cells) because of the six order-of-magnitude density variation from the front of the shield to the back. Even with a large variation in grid cell size and weighting factors it required 50,000 timesteps of statistical averaging to accurately capture the density profile.

Figure 4 shows a computation performed with US Air Force Phillips Lab collaborators investigating plume expansion at orbital conditions from small nozzles used as maneuvering devices. A pulsed piston is used to regulate 25-torr flow from the inlet orifice into the nozzle where the gas rapidly expands into vacuum. The temperature profile of the gas is shown in the figure. A low-temperature expansion region can be seen near the orifice with higher-temperature compression regions around the walls of the nozzle. The DSMC model is an ideal tool for these simulations because it can capture the gas thermodynamics from continuum conditions near the orifice to near-vacuum densities at the outflow boundaries.

Finally, Figure 1 illustrates simulations we have performed in conjunction with the SEMATECH consortium of semiconductor companies to study low-density gas flows in plasma etch and chemical vapor deposition reactors used for processing silicon wafers. The DSMC model can examine both the transient and steady-state regimes of the flow as gas is introduced into the vacuum chamber and compute the chemical reactions of interest that occur in the gas and on the wafer surface. The top portion of the figure shows the grid used for the simulation; the bottom portion shows the pressure profile and streamlines for gas entering through the sonic flow inlet ports at the top, crossing over the surface of the wafer, and being pumped out the bottom. The general gridding capabilities of our DSMC model are allowing the SEMATECH partners to study new designs on the computer; one company has begun to build an experimental chamber based on the DSMC results. We are currently working on extending the DSMC model to include electromagnetic field effects that are present in plasma reactors driven at radio frequencies. This requires adding the Lorentz equations for the motion of charged particles and solving Poisson's equation for the fields to couple the plasma to the boundary conditions.

REFERENCES

Bird, G. A. 1976. *Molecular Gas Dynamics*. Clarendon Press, Oxford.

Bartel, T. J. and Plimpton S. J. 1992. "DSMC Simulation of Rarefied Gas Dynamics on a Large Hypercube Supercomputer." In *Proceedings of the AIAA 27th Thermophysics Conference* (Nashville, TN, July 6-8), AIAA, Washington, DC, AIAA 92-2860.

Dagum, L. 1991. "Three-dimensional Direct Particle Simulation on the Connection Machine." in *Proceedings of the AIAA Thermophysics Conference* (Honolulu, HI), AIAA, Washington, DC, AIAA 91-1365.

MacDonald, J. D. 1991. "Particle Simulation in a Multiprocessor Environment." in *Proceedings of the AIAA Thermophysics Conference* (Honolulu, HI), AIAA, Washington, DC, AIAA 91-1366.

Plimpton, S. J. and Bartel, T. J. 1992. "Monte Carlo Particle Simulation of Low-Density Fluid Flow on MIMD Supercomputers." In *Proceedings of the Scalable High Performance Computing Conference (SHPCC-92)* (Williamsburg, VA, April 26-29), IEEE, Los Alamitos, CA, p. 212.

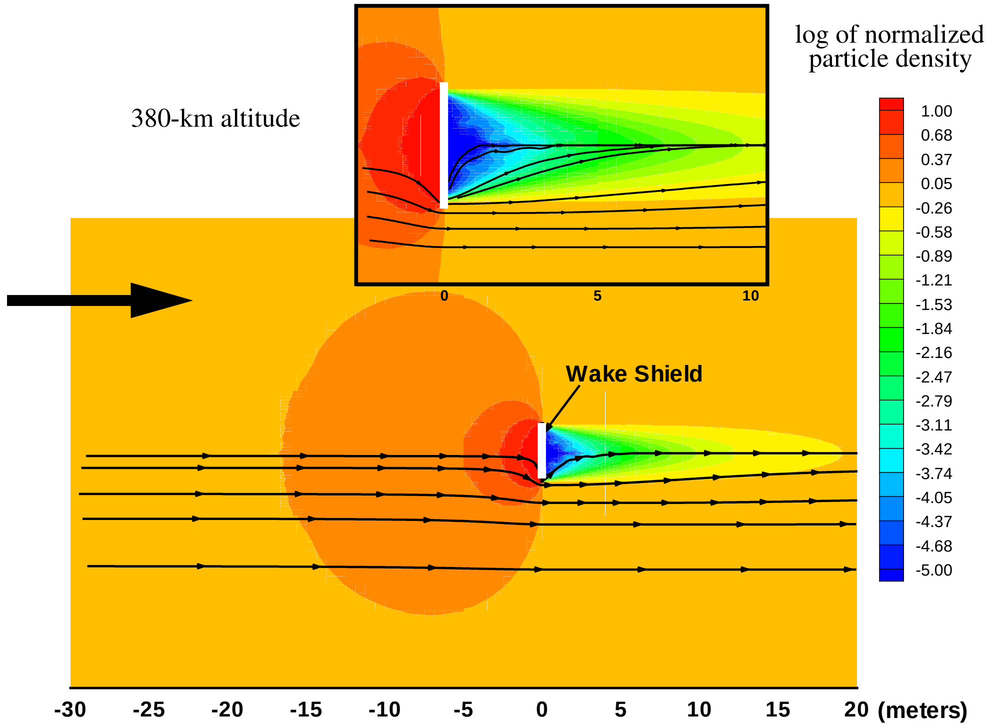

Figure 3: Flow around a 3.6-meter diameter shield moving at 7.8 km/s in low-earth orbit. The density profile and streamlines for individual particles are shown. There is a 10x density increase in front of the shield and a five order-of-magnitude density decrease in the wake behind the shield.

Figure 4: Gas expansion through a nozzle into vacuum. Gas enters the nozzle cavity from around a piston at the upper left corner of the diagram. The translational temperature profile of the gas is shown.