AIAA 97-0375

Role of Boundary Conditions in Monte Carlo Simulation of MEMS Devices

Robert P. Nance
North Carolina State University
Raleigh, NC

David B. Hash
National Research Council
NASA Ames Research Center, Moffett Field, CA

H. A. Hassan
North Carolina State University
Raleigh, NC

35th Aerospace Sciences Meeting and Exhibit
January 6-9, 1997 / Reno, NV

ROLE OF BOUNDARY CONDITIONS IN MONTE CARLO SIMULATION OF MEMS DEVICES

Robert P. Nance*
North Carolina State University, Raleigh, North Carolina
David B. Hash†
National Research Council, NASA Ames Research Center, Moffett Field, California
H. A. Hassan‡
North Carolina State University, Raleigh, North Carolina

Abstract

A study is made of the issues surrounding prediction of microchannel flows using the direct simulation Monte Carlo method. This investigation includes the introduction and use of new inflow and outflow boundary conditions suitable for subsonic flows. A series of test simulations for a moderate-size microchannel indicates that a high degree of grid under-resolution in the streamwise direction may be tolerated without loss of accuracy. In addition, the results demonstrate the importance of physically correct boundary conditions, as well as possibilities for reducing the time associated with the transient phase of a simulation. These results imply that simulations of longer ducts may be more feasible than previously envisioned.

Nomenclature

a	Local speed of sound
C_{mp}	Most probable thermal speed
Kn	Knudsen number
L	Channel length
n	Number density
$\dot{n}$	Particle flow rate
p	Pressure
R	Specific gas constant
T	Temperature
t	Channel half-height
V	Velocity magnitude
u, v	Velocity components in x and y directions
w	Channel half-width
λ	Fluid mean free path
μ	Fluid viscosity
ρ	Mass density
ξ	General coordinate direction

Subscripts:

e	Exit condition
i	Inlet condition
m	Computational cell index

Introduction

Microelectromechanical systems (MEMS) have recently become the focus of a great deal of attention in several research disciplines. These devices are manufactured using processes similar to those used in microprocessor fabrication and promise the opportunity to sense and control physical processes at length scales on the order of a micron.[1] Potential applications for such devices cover a broad spectrum, including adaptive optics, surgical instruments, and laminar flow control. Many of these proposed designs involve fluid flow through microchannels. One such possibility is the use of microchannels to dissipate heat in integrated circuits. Although such microscale flows have actually been studied for nearly a century,[2] the growth in MEMS research has prompted a resurgence in the investigation of these flows. This work includes experimental and analytical research.[3,4]

Because of the very small length scales associated with MEMS devices, the fluid mean free path can be on the order of the characteristic channel dimensions. This fact indicates that treating the flow in the microchannel

* Research Assistant, Mechanical and Aerospace Engineering, Student Member AIAA.

† Research Associate, Member AIAA.

‡ Professor, Mechanical and Aerospace Engineering, Associate Fellow AIAA.

as a continuum phenomenon may lead to inaccurate results. Additionally, if the mean free path is large compared to the channel dimensions, it will certainly be much larger than the gradient scales in the flow, leading to very large local Knudsen numbers. The use of continuum-based techniques in microchannel analysis and design may therefore lead to even larger errors than expected. However, microchannel flows appear to be well-suited to the application of Bird's direct simulation Monte Carlo (DSMC) method.[5] This technique models a gas flow as a myriad of discrete computational particles, with each simulated particle representing a large number of real particles. These simulated particles move through the computational domain and collide with one another as well as computational boundaries. The particle properties are then sampled to determine macroscopic flow quantities, such as velocity and density. DSMC has been used with great success in the prediction of rarefied hypersonic flows, and some researchers have already applied it to microchannel flows.[6,7]

However, microchannel simulations present a set of challenges not encountered in previous DSMC applications. First of all, the flow velocities in these systems are generally much less than the speed of sound. Consequently, the "stream" and "vacuum" boundary conditions typically employed in DSMC calculations are not physically appropriate. Instead, one must use inflow and outflow conditions that impose the correct propagation of information across the boundaries. This fact is now quite well-established, and alternative DSMC boundary conditions for low-speed flows have been described in a number of sources.[6,7,8]

Secondly, microchannels are characterized by very large aspect ratios (ratio of channel length to channel height). One of the fundamental assumptions in the DSMC method is that the cell size in the gradient direction must be less than a fluid mean free path if the solution is to be considered valid. Proper resolution of both streamwise and transverse flow directions will lead to a large number of cells, implying that a single timestep will be quite expensive to compute.

The above problem is aggravated by the need to compute a very large number of timesteps before steady flow is attained. Generally, the length of the transient portion of a simulation is estimated by considering the amount of time necessary for a particle traveling at the mean flow velocity to traverse the computational domain. Since the domain size (*i.e.*, channel length) is relatively large and the flow velocity is relatively small, the flow "residence time" will be considerably larger than in typical reentry simulations. Moreover, operating conditions for such channels are typically at or around standard atmosphere and pressure, leading to a high collision frequency. A high collision frequency corresponds to a low mean collision time. Since DSMC also requires that the simulation timestep be less than the mean collision time, the total number of timesteps associated with the transient phase will be very large.

Finally, microchannel simulations require much larger run times than traditional problems even after steady flow has been established. It has been pointed out[6] that many microchannel flows involve mean flow velocities on the order of 0.1 m/s. Thus, at ambient temperatures around 300 K, the ratio of the mean flow velocity to the average thermal speed can be on the order of 0.01% or even less. This extremely low signal-to-noise ratio has serious implications for DSMC since the method is probabilistic in nature. Some researchers have estimated[6] that Monte Carlo simulation of realistic microchannels will require over 100 million steady-state samples in order to eliminate scatter from the results.

These characteristics indicate that realistic microchannel simulations will push the limits of existing computer technology, if they are even currently feasible. The purpose of this article is to examine the above issues in the context of a moderately-sized test case and determine what, if any, approaches can be taken to improve solution accuracy and turnaround time.

DSMC Algorithm

This investigation utilizes a parallel version of Bird's DSMC3 demonstration code.[9] Although the original algorithm was developed for three-dimensional simulations, only two-dimensional cases are considered here. A uniform Cartesian grid is used to minimize the time associated with particle movement, and parallelization is achieved through use of the CHAOS runtime library developed at the University of Maryland.[10] This library provides the programmer with simple procedure calls for data migration, domain decomposition, and dynamic load balance. CHAOS has already been shown to yield excellent parallel performance when coupled with the DSMC3 algorithm.[11]

In order to perform accurate microchannel solutions, new inflow and outflow boundary conditions were developed and incorporated into the basic algorithm. These new boundary conditions are described below, as well as some further modifications to improve convergence to steady flow.

Inflow Boundary Conditions

At the inflow boundary, we elect to specify the pressure, temperature, and transverse velocity (assumed to be zero for the present application). The streamwise velocity is determined for each boundary cell through

consideration of the fluxes across the cell's boundary face and enforcing conservation of particles.

For a given mean speed and temperature, the particle flux across a boundary in a particular direction can be determined as follows, assuming a Maxwellian distribution:[9]

$$\dot{n} = \frac{nC_{mp}\left[\exp\left(-q^2\right) + \sqrt{\pi}q\left(1 + \mathrm{erf}q\right)\right]}{2\sqrt{\pi}} \qquad (1)$$

$$q = s\cos\theta$$

where s is the speed ratio V/C_{mp}, and θ is the angle between the velocity vector and normal to the boundary element. Considering the boundary face of inflow cell m, we can apply Eq. (1) to determine the flux crossing the face in either direction. Then, if particles are conserved,

$$\left(\dot{n}_+ - \dot{n}_-\right)_m = n_i\left(u_i\right)_m A \qquad (2)$$

where the subscripts + and - refer to particle fluxes in the positive and negative x directions, respectively. A is the area of the boundary face. Note that the inflow velocity depends on the incoming number flux, which in turn depends on the inflow velocity. Hence, Eq. (2) is actually a nonlinear function of the inflow velocity. Instead of solving for $\left(u_i\right)_m$ numerically, though, we use the last computed value to evaluate the positive number flux. As a result, we can easily rearrange Eq. (2) to determine the new inflow velocity:

$$\left(u_i\right)_m = \frac{\left(\dot{n}_+ - \dot{n}_-\right)_m}{n_i A} \qquad (3)$$

The value of $\left(u_i\right)_m$ will vary during the simulation. However, the velocity should eventually attain a nearly constant value in each inflow cell. The known inflow properties are then used to determine the entering particle distributions.

This inflow condition is quite similar to the subsonic boundary conditions proposed by Ikegawa and Kobayashi.[8] One important difference is that Ikegawa and Kobayashi use the particle-conservation concept to come up with a constant inflow velocity, whereas in this work particle conservation is applied on a per-cell basis. Additionally, the particle fluxes in this implementation are computed from the Maxwellian distribution. In contrast, Ikegawa and Kobayashi determine the particle fluxes by actually counting the number of particles crossing the computational boundary.

Outflow Boundary Conditions

For the outflow, we appeal to the theory of characteristics, which is frequently used in continuum calculations to derive boundary conditions for subsonic flows. The use of the theory implicitly assumes the flow to be locally inviscid, adiabatic, and close to a perfect gas. Note that we can apply the theory of characteristics even though the flow is rarefied, since the conservation equations themselves still hold. In this case, we have employed Whitfield's characteristic formulation,[12] since it allows the specification of a constant exit pressure. The resultant equations (for a Cartesian grid) are:

$$\left(\rho_e\right)_m = \rho_m + \frac{p_e - p_m}{a_m^2} \qquad (4)$$

$$\left(u_e\right)_m = u_m + \frac{p_m - p_e}{\rho_m a_m} \qquad (5)$$

$$\left(v_e\right)_m = v_m \qquad (6)$$

In the above relations, the subscript e signifies exit quantities. Since the exit pressure is known, the new value of the exit temperature can then be determined using the perfect-gas law:

$$\left(T_e\right)_m = p_e / \left[\left(\rho_e\right)_m R\right] \qquad (7)$$

As with the inflow conditions, the outflow properties are expected to vary during the simulation before settling out to steady values.

Convergence Enhancement

As discussed in the Introduction, simulation of low-Mach-number microchannel flows can be very time-consuming, partly because of the very long transient period associated with such problems. In an effort to improve convergence for low-speed microchannel flows, two further modifications were made to the present DSMC algorithm.

First, a closed-form continuum solution is used for flowfield initialization instead of the traditional uniform- stream condition. The analytic solution is obtained through integration of the Navier-Stokes equations for isothermal, steady, two-dimensional flow with velocity slip.[4] This approach yields the following result for the velocity in the x direction:

$$u = \frac{1}{2\mu}\left(\frac{dp}{dx}\right)\left(y^2 - Ct^2\right) \qquad (8)$$

$$C = 1 + 2\mathrm{Kn}$$

where the y coordinate is measured from the channel centerline and fully diffuse surface reflection is assumed.

Eq. (8) shows that the velocity is a function of the pressure gradient as well as the local Knudsen number

based on channel height. This quantity can be determined using

$$\mathrm{Kn} = \mathrm{Kn}_e \frac{p_e}{p(x)}$$

The pressure distribution is found to be only a function of x:

$$p = p_e \sqrt{(C_1 + 1)^2 + C_2\left(1 - \frac{x}{L}\right)} - C_1 p_e \qquad (9)$$

$$C_1 = 3\mathrm{Kn}_e$$

$$C_2 = \frac{3\mu RTLQ_m}{2wt^3 p_e}$$

Q_m is the volume flow rate, given by

$$Q_m = \frac{2wt^3 p_e^2}{3\mu RTL}\left[\left(\frac{p_i}{p_e}\right)^2 - 1 + 6\mathrm{Kn}_e\left(\frac{p_i}{p_e} - 1\right)\right] \qquad (10)$$

These expressions are used to determine the streamwise velocity and number density at the center of each computational cell during the initialization phase. The cell is then populated with simulated particles generated according to these conditions.

Second, it was observed that the computed inlet and exit mass fluxes can become negative during the transient phase of the simulation. Since negative mass fluxes are to be avoided, we simply choose not to update the boundary conditions at the inflow (or outflow) if a negative inlet (or exit) mass-flow rate is calculated.

Results and Discussion

The algorithm described above was applied to the microchannel geometry and flow conditions shown in Fig. 1. The given dimensions correspond to a channel aspect ratio of 60, and O_2 was specified as the working gas. Additionally, the Knudsen number based on channel height ranges from about 0.02 at the channel inlet to about 0.05 at the channel exit. This range of Knudsen numbers suggests that the channel operates in the slip-flow regime.

Several runs were conducted in order to ascertain the impact of various simulation parameters on the quality of the solution. In each run, an initial population of 30 simulated particles per cell was specified. This value is larger than what is often specified in simulations, but was used in order to help reduce scatter. All computations were carried out on an IBM SP2, with 8 to 25 processors used for each run.

Comparison to Navier-Stokes Solution

Fig. 2 compares a grid-resolved solution obtained using the DSMC algorithm to the analytic Navier-Stokes solution described above. Both solutions return the correct inflow and outflow pressures. Additionally, both solutions show a fairly significant departure from the linear pressure distribution predicted by traditional (no-slip) continuum theory. Although not shown here, it was found that the DSMC pressure distribution displays little variation in the transverse direction. This characteristic is in agreement with the continuum result for the pressure distribution.

This plot also shows differences between the DSMC and Navier-Stokes pressure distributions away from the inflow and outflow boundaries. Somewhat larger differences were observed in the velocity distributions, particularly near the channel wall, as may be observed in Fig. 3. This discrepancy is a consequence of a higher local degree of rarefaction. As we move to the centerline, the difference between the two results decreases, as shown in Fig. 4.

We may also compare the surface shear-stress distributions predicted by the two methods, as shown in Fig. 5. A considerable amount of scatter is present in the DSMC solution, but it agrees fairly well with the Navier-Stokes result across most of the channel wall.

Influence of Grid Resolution

One of the requirements for a successful DSMC application is that the cell size in the gradient direction be less than a mean free path, as stated earlier. However, this requirement does not give us any insight into how well-resolved the grid must be normal to the primary gradient direction. In channel flow, there are gradients in both the streamwise and transverse directions, but the gradients in the latter tend to be much greater. If high resolution is required in both directions for an accurate solution, simulation of very high-aspect-ratio microchannels will be virtually impossible. Thus, one important consideration is the degree of under-resolution which can be tolerated in the streamwise direction.

In order to make such a determination, the test channel was first simulated using a fully-resolved grid (cell size in each direction less than a mean free path). This solution required 2400 cells in the streamwise direction and 40 cells normal to the channel wall, and yielded a total of approximately 2.1 million simulated particles at steady state. Once this solution was complete, the grid in the x direction was coarsened by successively doubling the cell length. This procedure was repeated several times, and the results were compared to those for the baseline, fully-resolved solution.

Figures 6 and 7 show some representative results from this procedure. Property distributions along the channel centerline are compared for three different levels of refinement in the streamwise direction. Note that there is a factor-of-32 difference between the resolution on the finest grid and on the coarsest. Even so, both the pressure (Fig. 6) and velocity (Fig.7) distributions agree very well on all three grids. In order to gain some insight on why such under-resolution is permissible, consider the variation of the local Knudsen number, defined as

$$\mathrm{Kn}(Q,\xi) = \left(\frac{\lambda}{Q}\right)\frac{\partial Q}{\partial \xi}$$

where Q is the flow property used to define the gradient scale. Note that the value of this quantity depends both on the direction under consideration as well as the flow property being examined. In regions of strong gradients, this Knudsen number will be large. Fig. 8 shows values of this local Knudsen number for differing flow properties and directions. For this plot, the local Knudsen numbers are shown at the surface. Based on these results, it is clear that the velocity gradient in the y direction is much stronger than the pressure gradient in the x direction. (The velocity gradient in y is also much greater than the velocity gradient in x; for clarity, however, Kn(u,x) is not shown in this figure.) Since the streamwise direction is not the direction of the primary flow gradient, it makes sense that a loss of resolution in this direction will not have a significant impact on the accuracy of the solution.

This plot also helps make another important point. As indicated at the beginning of this section, the maximum value of Knudsen number based on channel dimension is about 0.05. However, Fig. 8 indicates that the maximum value of Kn based on gradient length scales is around 0.5—ten times greater than the value based on channel dimension. Since the local gradient scale is the proper length scale to use in Knudsen-number computation, this difference implies that the degree of rarefaction present in a microchannel flow can be considerably greater than that predicted by using gross overall dimensions.

Of course, reduction in grid resolution will have a significant impact on computation time. Table 1 lists measured computation times for completion of 100 steady-state print cycles (10,000 simulation time steps). Since each of these cases was run on varying numbers of processors, the figures listed are in terms of node-hours (CPU hours required multiplied by number of processors used). There is a factor-of-36 difference between the execution time on the fully-resolved grid and the CPU time on the coarsest grid used. Each of the solutions shown here was obtained using a total of 1,000 print cycles. Thus, the coarse-grid solution requires less than 1/30th the overall time required for the fine-grid solution.

Influence of Boundary Conditions

In order to illustrate the significance of correct boundary conditions on the solution, the coarsest-grid case was rerun using standard DSMC stream conditions for the inflow and outflow planes. In order to apply these conditions, it is necessary to specify an inflow velocity. Based on the results shown above, a value of 20 m/s was selected. Fig. 9 compares computed pressure distributions along the duct centerline for the subsonic boundary conditions and the conventional boundary conditions. It is obvious from this graph that use of the physically incorrect conditions results in an erroneous solution.

Fig. 9 also shows results for this case computed using the subsonic boundary conditions proposed by Piekos and Breuer.[7] This plot shows that these results are virtually identical to those obtained with the boundary conditions proposed here. The agreement for other flow properties was found to be equally good.

Influence of Initial Conditions

It was mentioned earlier that the analytic Navier-Stokes solution was used to initialize the flowfield for the DSMC computations. We would like to determine if this initialization method biases the final result in any way. Additionally, it would be instructive to determine the relative benefit, if any, of starting the DSMC computation in this manner.

To help answer the first question, the coarsest-grid solution was rerun beginning with a uniform initial condition. Such an initial state is commonly used in DSMC computations. Fig. 10 compares this result with the result obtained with the Navier-Stokes initialization. Clearly, the two solutions are very similar; examination of other flow properties indicated that the differences between two results are well within the statistical scatter.

Both of the solutions shown in Fig. 10 were obtained over the same number of steady-state samples. The number of print cycles to reach the assumed steady-flow state was also the same for each run. Therefore, the only savings afforded by the Navier-Stokes initialization lies in the use of fewer initial particles (when initializing with a uniform flow, no account is made of the decreasing density through the channel). However, it may be possible to start steady-flow sampling earlier than suggested by the transient-flow analysis described in the Introduction.

If we examine the time histories of the measured inlet and exit mass-flow rates for the uniform-initialization case (Fig. 11), we see that the inlet and exit values "converge" long before the transient period is assumed to be over. In addition, the mass-flow histories for the Navier-Stokes initialization case (Fig. 12) show no real variation prior to the beginning of steady flow. These graphs indicate that we may be able to begin steady sampling after about 200 print cycles for the uniform-initialization case, and only 100 cycles for the Navier-Stokes initialization case.

To validate the above hypothesis (and also to help answer the second question posed at the beginning of this section), each of the cases was rerun with the number of prints to steady flow reset to the values listed above. Figs. 13 and 14 compare results obtained using the Navier-Stokes initialization and the original transient phase of 635 print cycles to results obtained with the shortened transient phases. The results for all three solutions show excellent agreement. It therefore appears that shortening the transient phase is permissible, both when uniform and Navier-Stokes initialization techniques are used. However, it appears that steady-flow sampling can be started sooner by initializing with the Navier-Stokes solution instead of a uniform stream.

It is important to note that the effectiveness of the Navier-Stokes initialization is most likely dependent on the overall degree of rarefaction, as well as the magnitude of the flow velocity. If the final DSMC solution departs more drastically from the continuum solution than in the case considered here, there will almost certainly be a more visible transient region before the inlet and exit mass fluxes converge to nearly equal values. However, based on the results shown here, it appears that monitoring of the mass-flow history to determine when steady flow is established is a valid procedure.

For lower mean velocities, we expect to find general convergence trends similar to those reported above. The time required to establish convergence of the mass fluxes may increase substantially, though, because of the even lower signal-to-noise ratio.

Overall Performance

Table 2 summarizes the performance benefits realized by reducing the streamwise grid refinement and decreasing the length of the transient phase of the simulation. These figures indicate that the original case run (fully-resolved, original transient length) would require more than 65 days of continuous run time on a single SP2 processor. For this particular run, 25 nodes were used, resulting in an actual overall CPU time of about 2.6 days. Of course, the total wallclock time required was considerably greater (a week or more) because of queue waiting time.

At the other extreme, the solution utilizing only 75 cells in the streamwise direction and 100 prints to steady flow would need less than a single day on one node. Since 8 processors were used for this simulation, less than 3 CPU hours were actually required to obtain the results. The total wallclock time for the run is not much greater, since this smaller job was usually able to run with little or no waiting. The CPU time could be further reduced for any of these cases by increasing the number of processors, but the relative benefit would be decreased by dropoff in parallel efficiency and increased queue time.

Concluding Remarks

In this work, we have proposed new DSMC boundary conditions suitable for use in low-speed MEMS applications. These conditions respect the proper directions of signal propagation for subsonic flows, and allow specification of inlet and exit pressures. Low-speed microchannel computations using these new conditions yield much better results than when traditional DSMC boundary conditions are used. However, the solutions are essentially the same as those obtained with the boundary conditions proposed by Piekos and Breuer.

The results obtained from this work also indicate that it is possible to effect a substantial reduction in the CPU time required to obtain an accurate DSMC solution of a microchannel flow. This decrease in compute time is obtained by increasing the cell size in the x direction and using the Navier-Stokes initialization to reduce the time to steady flow. These modifications were shown to yield a solution approximately 70 times faster (on a per-node basis) than on a fully-resolved grid using residence time to gauge the onset of steady flow.

It remains to be seen whether this approach will be equally effective for higher degrees of rarefaction. In addition, a factor-of-70 decrease in solution time may still not be great enough to permit simulation of longer microchannels in a reasonable amount of CPU time. Future work should thus be focused on the effectiveness of this approach for varying Knudsen number, as well as application to progressively longer channels.

Acknowledgments

This work is supported in part by NASA Cooperative Agreement NCC1-112 and a National Defense Science and Engineering Graduate Fellowship. Computer resources were provided by the Numerical Aerody-

namic Simulation facilities at NASA Ames Research Center.

References

1. Ho, C.-M. and Tai, Y.-C., "MEMS: Science and Technology", *Application of Microfabrication to Fluid Mechanics*, ASME, 1994, pp. 39-49.
2. Knudsen, M., "Die Gesetze der Molecular Stromung und die Inneren Reibungstromung der Gase Durch Rohren", *Annalen der Physik*, Vol. 28, 1909, pp. 75-130.
3. Arkilic, E. B., Breuer, K. S., and Schmidt, M. A., "Gaseous Flow in Microchannels", *Application of Microfabrication to Fluid Mechanics*, ASME, 1994, pp. 57-66.
4. Liu, J., Tai, Y.-C., and Ho, C.-M., "MEMS for Pressure Distribution Studies of Gaseous Flows in Microchannels", *Micro Electro Mechanical Systems*, IEEE, 1995, pp. 209-215.
5. Bird, G. A., "Monte Carlo Simulation in an Engineering Context", *Rarefied Gas Dynamics*, Vol. 74, Pt. 1, Progress in Astronautics and Aeronautics, AIAA, New York, 1981, pp. 239-255.
6. Oh, C. K., Oran, E., and Cybyk, B. Z., "Microchannel Flow Computed with the DSMC-MLG", AIAA Paper 95-2090, June 1995.
7. Piekos, E. S. and Breuer, K. S., "DSMC Modeling of Micromechanical Devices", AIAA Paper 95-2089, June 1995.
8. Ikegawa, M. and Kobayashi, J., "Development of a Rarefied Flow Simulator Using the Direct-Simulation Monte Carlo Method," *JSME International Journal*, Ser. 2, Vol. 33, No. 3, 1990, pp. 463-467.
9. Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, England, UK, 1994.
10. Hwang, Y. S., Moon, B., Sharma, S., Ponnusamy, R., Das, R., and Saltz, J., "Runtime and Language Support for Compiling Adaptive Irregular Problems on Distributed Memory Machines", *Software Practice and Experience*, Vol. 25, No. 6, June 1995, pp. 597-621.
11. Nance, R. P., Wilmoth, R. G., Moon, B., Hassan, H. A., and Saltz, J., "Parallel Monte Carlo Simulation of Three-Dimensional Flow over a Flat Plate", *Journal of Thermophysics and Heat Transfer*, Vol. 9, No. 3, July-September 1995, pp. 471-477.
12. Whitfield, D. L., "Three-Dimensional Unsteady Euler Equation Solutions Using Flux Vector Splitting", December 1983.

Tables

Table 1. Timing comparisons for grid refinement.

Number of cells in x direction (μ m)	Time for 100 steady-state prints (node-hr)
2400	156.7
300	17.03
75	4.329

Table 2. Overall solution times.

Solution	Number of steady-state samples	CPU time (node-hr)
Baseline (fully resolved)	36,600	1567
Reduce number of cells in x direction	36,600	43.29
Shorten transient with uniform initialization	40,000	25.97
Shorten transient with N-S initialization	40,000	21.65

Figures

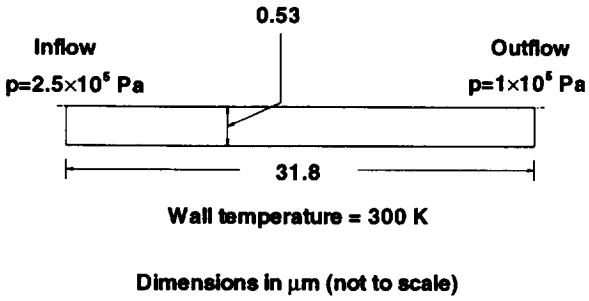

Figure 1. Microchannel geometry considered in this work.

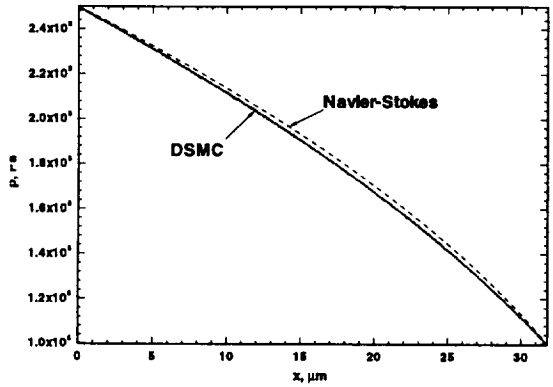

Figure 2. Comparison of Navier-Stokes and DSMC centerline pressure distributions.

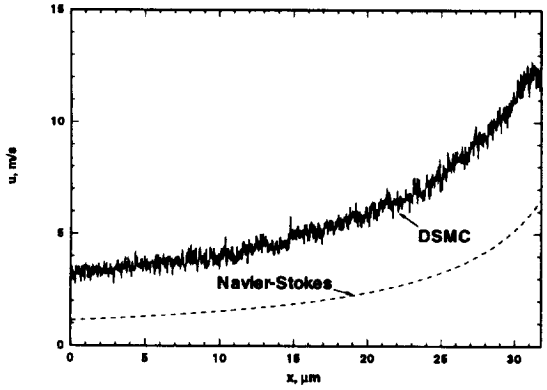

Figure 3. Comparison of Navier-Stokes and DSMC slip velocity distributions.

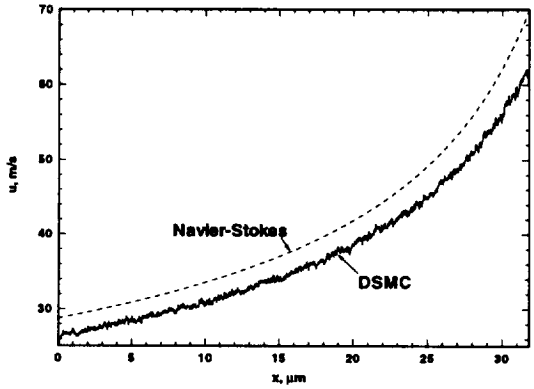

Figure 4. Comparison of Navier-Stokes and DSMC centerline velocity distributions.

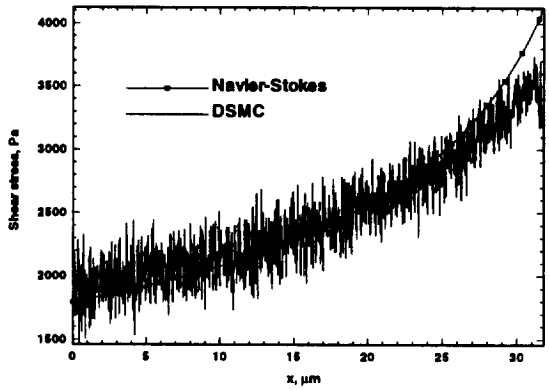

Figure 5. Comparison of Navier-Stokes and DSMC wall shear stresses.

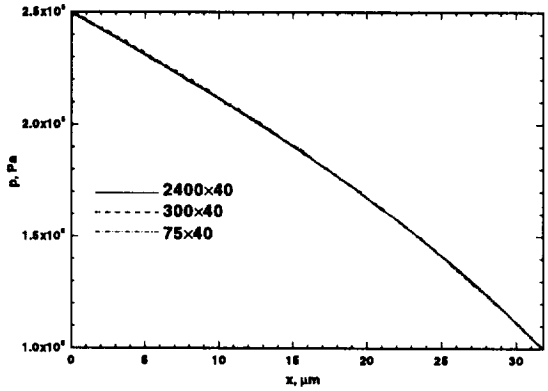

Figure 6. Influence of grid resolution on centerline pressure distribution.

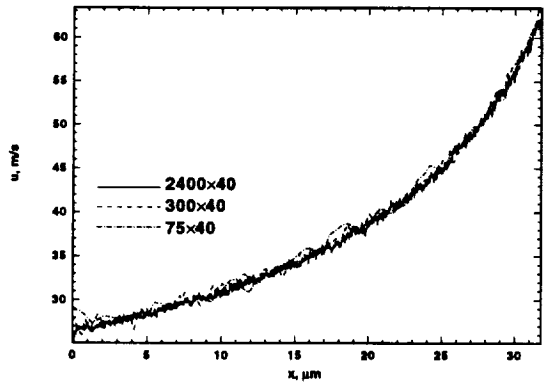

Figure 7. Influence of grid resolution on centerline velocity distribution.

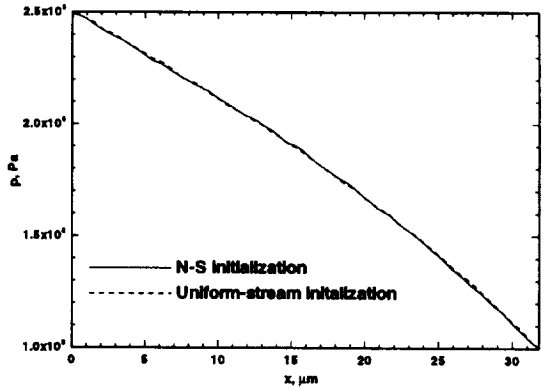

Figure 10. Comparison of pressure distributions for Navier-Stokes and uniform initialization.

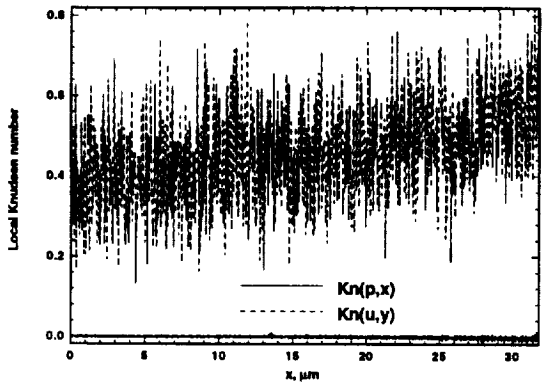

Figure 8. Comparison of local Knudsen-number values at the channel wall.

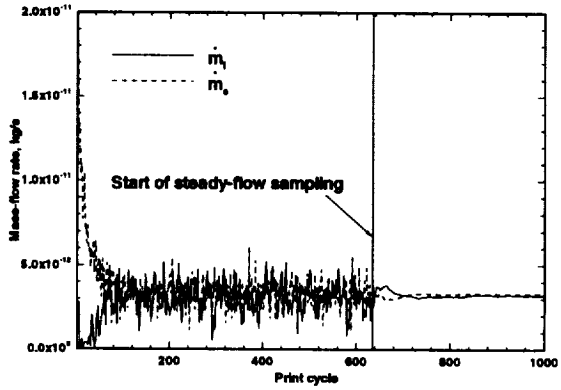

Figure 11. Evolution of measured mass-flow rates for uniform initialization.

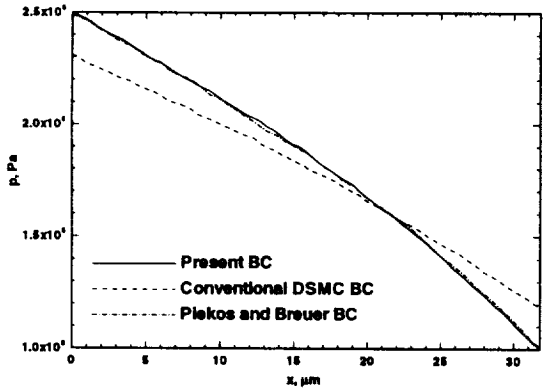

Figure 9. Comparison of pressure distributions for new boundary conditions and standard DSMC boundary conditions.

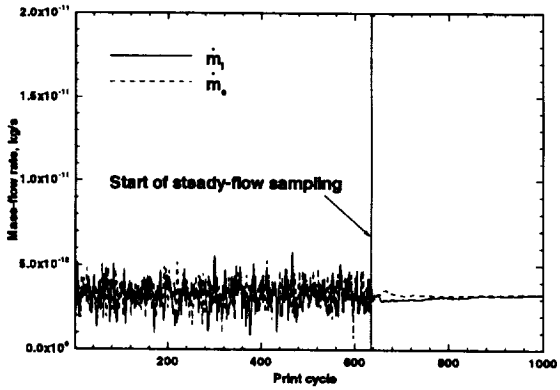

Figure 12. Evolution of measured mass-flow rates for Navier-Stokes initialization.

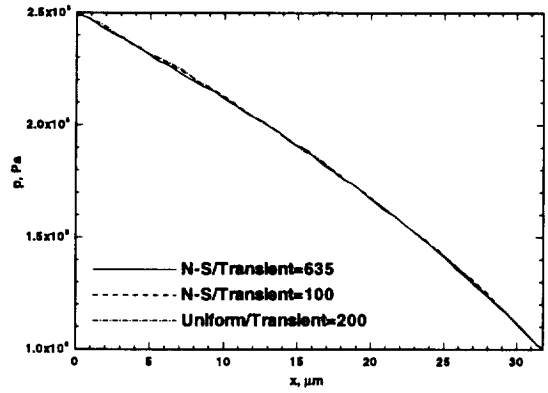

Figure 13. Comparison of pressure distributions for different initializations and transient lengths.

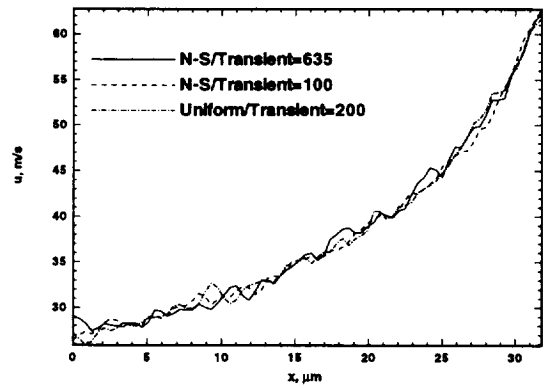

Figure 14. Comparison of centerline velocity distributions for different initializations and transient lengths.