Journal of Computational Physics **178,** 342–372 (2002)
doi:10.1006/jcph.2002.7030, available online at http://www.idealibrary.com on IDEAL®

A Combined Continuum/DSMC Technique for Multiscale Analysis of Microfluidic Filters

Ozgur Aktas and N. R. Aluru[1]

Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, 405 N. Mathews Avenue, Urbana, Illinois 61801

Received June 12, 2001; revised December 3, 2001

A multiscale method that combines continuum fluid models with the direct simulation Monte Carlo (DSMC) method is presented. The continuum regions are treated by Stokes equations and a scattered point based finite cloud method is employed to solve the Stokes equations. The continuum and DSMC regions are combined by an overlapped Schwarz alternating method with Dirichlet–Dirichlet type boundary conditions. A scattered point interpolation scheme is developed to interpolate the solution between subdomains. The convergence characteristics of the multiscale approach are investigated in detail. Specifically, the dependence of convergence on the overlap size, the DSMC noise, and the number of time steps employed in the DSMC algorithm are studied. While the convergence depends weakly on the DSMC noise and the overlap size, the number of DSMC time steps simulated in each coupling iteration should be selected so that the total time steps simulated until convergence of the coupled process is close to the time constant of the DSMC subsystem. Steady-state analysis of microfluidic filters is studied in detail using the multiscale approach. The multiscale approach is also applied for the simulation of a membrane with an array of microfluidic filters and a dual-stage microfluidic device with an array of microfluidic filters for particle trapping and sorting.

Key Words: multiscale analysis; DSMC; finite cloud method; meshless methods; microfluidics.

1. INTRODUCTION

Prediction of gas flow through microfluidic devices is important to enable the design and development of complex microelectromechanical systems (MEMS) [1–4]. Simulation of gas flow through microfluidic devices is, however, very complicated because of the breakdown

[1]Assistant Professor, Beckman Institute for Advanced Science and Technology, http://www.staff.uiuc.edu/~aluru, aluru@uiuc.edu.

0021-9991/02 $35.00

of continuum theories. The small dimensions encountered in microfluidic devices result in rarefaction of the flow, which necessitates the use of molecular approaches such as a direct simulation Monte Carlo method (DSMC). In order to model MEMS devices it is often necessary to consider the complete flow path, including large regions where rarefaction is not observed. At the very minimum, the input and output regions of the device need to be simulated to avoid difficulties with far field boundary conditions. As a consequence, for most microfluidic devices of interest, rarefaction is limited to certain critical regions containing small length scales. The simulation of such multiscale problems by DSMC alone is difficult, since in regions that are not rarefied, the DSMC method has low computational efficiency. On the other hand, the use of continuum theories for the entire device can produce inaccurate results. An efficient approach for such multiple length scale problems is to develop a multiscale approach combining continuum theories with molecular approaches such as molecular dynamics or DSMC [5–8]. In this paper, we investigate the application of the alternating Schwarz method for combining continuum theories with a DSMC method for simulation of microfluidic devices.

Even though DSMC is a versatile method and has been used extensively for the simulation of rarefied flows [9, 10], its computational cost can be significant, limiting its usefulness. Despite the large computational cost, the DSMC technique has been applied successfully to simulate rarefied flows in microgeometries [10–15]. The use of DSMC alone for analysis and design of complex microfluidic systems can be daunting because of the enormous computational cost. Specifically, for microflow simulations, the computational cost can be enormous when (i) the system under study has large regions with low Knudsen number (Kn), (ii) the flow velocity is low, and (iii) when the time constant of the system is large. Multiscale methods are attractive alternatives for simulation of microfluidic systems as they significantly reduce the computational cost. For example, the low Kn regions can be simulated by continuum theories effciently and accurately there by necessitating the use of DSMC only for low velocity rarefied regions. If steady-state results are desired, then the continuum domain can be simulated by time-independent fluid models, which require significantly less time than solving the time-dependent fluid models. In this paper, we focus on predicting steady-state flow characteristics in microfluidic devices by developing an efficient multiscale approach.

Much of the work published on coupling DSMC with fluid models such as Navier–Stokes equations was based on nonoverlapping decomposition of the simulation domain [5–7]. The half–fluxes method [6] has been used successfully to achieve coupling in a nonoverlapping decomposition for space vehicle reentry problems. However, the DSMC scatter in estimation of higher-order moments of the distribution function presents a big difficulty for the application of the half-fluxes method to MEMS devices, where the flow velocity is small and correspondingly the associated scatter can be large [5]. Another technique that was used to achieve coupling is the extrapolation of the flow properties to the interface. This method is affected to a lesser degree from DSMC scatter. However, this method is still critically dependent on the smoothing of the solutions obtained from DSMC and on the accuracy of the extrapolation [5]. As compared to earlier work on coupling of DSMC with continuum methods, the approach in the present paper considers coupling of DSMC with time-independent continuum models using an overlapped decomposition and emphasizes the use of Dirichlet boundary conditions in order to attain good efficiency with the coupled method. In comparison to the method presented in [8], which considers coupling of continuum methods with a molecular dynamics technique, the present paper considers the

coupling of DSMC with fluid flow equations, presents a detailed investigation of convergence characteristics, and introduces the use of scattered point methods for interpolation between domains.

An issue with multiscale methods is the treatment of the interface between continuum and molecular (or DSMC) regions. Mesh-based methods, such as finite element methods, can be complicated allowing little flexibility between a finite element mesh for the continuum and a cell structure for the molecular or the DSMC region. Scattered point methods or meshless methods [16, 17] for the solution of the continuum equations are attractive in a multiscale approach as they allow for an arbitrary treatment of the interface between continuum and DSMC regions. In particular, the continuum nodes and the DSMC cells can be placed randomly in the overlapped continuum and DSMC regions and a scattered point interpolation technique can be used to transfer data between domains. In this work, we use a scattered point finite cloud method (FCM) [16, 17] for the solution of the fluid models in the continuum regions. FCM uses a fixed kernel technique for the construction of interpolation functions and a collocation technique for the discretization of the governing equations. A fixed kernel technique is also used to construct a scattered point interpolation scheme between continuum and DSMC regions.

The rest of the paper is outlined as follows: In Section 2, microfluidic filters are introduced. In Section 3, the multiscale method is described in detail. In Section 4.1 the convergence properties of the coupled method are investigated, and in Section 4.2 several examples of the application of the coupled method are presented. Conclusions are presented in Section 5.

2. MICROFLUIDIC FILTERS

Several types of microfluidic filters were reported in the literature. In [2], microfluidic filters were developed to capture airborne particles for detailed chemical analysis. The filters were fabricated by opening an array of holes on thin silicon membranes. The membrane thickness was 1 μm and a typical filter or hole size, determined by the minimum size of the particles to be filtered, ranged from 5 to 10 μm. Figure 1 shows the cross-sectional images of the filters fabricated in [2]. When the filters shown in Fig. 1 were coated with a polymer layer, a burst pressure as high as 0.25 atm was obtained. Other filter-like devices, with much smaller hole sizes of 50 and 100 nm, were developed for immunologic isolation applications [3]. Even though the primary application area is in liquid flows, gas flow tests

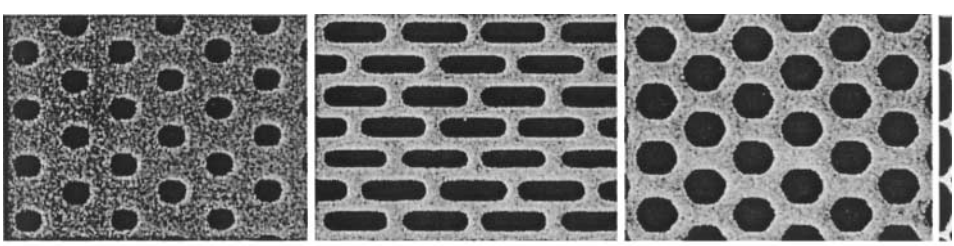

FIG. 1. The plain-view images of some of the filters developed in [2]. Reproduced with permission from [2] (© *IEEE*).

were performed by applying a pressure difference of 1.17 atm to verify that the holes were successfully manufactured. In much earlier work, filter-like structures that consist of small holes on a thin membrane were fabricated by irradiation of mica membranes followed by etching [4]. The mica membranes were 7.5 μm thick, had a diameter of 11/16 inch and were tested at 1 atm pressure difference. The pore radii varied from 2 nm to several micrometers.

Microfluidic filters are used as test beds to investigate the accuracy, efficiency, and convergence characteristics of the multiscale approach described in this paper. The flow in microfilter systems is characterized by the presence of different length scales [15]. Within the filter channel, the characteristic length is that of the channel height, whereas outside the channel the characteristic length is much larger. Within the filter channel and close to the filter membrane the flow is rarefied and needs to be simulated with DSMC. However, farther away from the channel the Kn is low and a DSMC treatment is not necessary, and the flow can be simulated by employing an appropriate continuum model. The use of DSMC in the low Kn regions increases the computational cost significantly. For this reason, an approach involving the coupling of continuum models with DSMC seems optimal for microfluidic filter applications.

3. MULTISCALE APPROACH

The multiscale approach discussed in this paper uses an overlapped Schwarz method with Dirichlet–Dirichlet type boundary conditions for solving the steady-state flow problems encountered in microfluidic filters. It is assumed that the validity regions for continuum and atomistic models have already been identified. The continuum region, where continuum models hold good, is simulated by Stokes equations using the finite cloud method, and the atomistic region, where continuum models fail, is simulated by DSMC. It is clear that, in general, the flow in the continuum region can show compressibility effects. Thus, the use of Stokes equations means that a smaller region, where compressibility effects are negligible, can be simulated by continuum models. If Navier-Stokes equations are used for continuum simulation, a larger region can be simulated by continuum models resulting in a larger speed-up. However, the use of Stokes equations is comparatively easier and enables the investigation of fundamental properties of the multiscale approach.

In the following, Section 3.1 presents the finite cloud method for scattered point analysis of Stokes equations, Section 3.2 discusses the DSMC issues that relate to the multiscale method, and Section 3.3 presents the details of the coupling technique.

3.1. Scattered Point Analysis of Stokes Equations

A scattered point approach for analysis of Stokes equations involves two steps. The first step is the construction of interpolation functions given a scattered set of points and the second step is the discretization of the Stokes equations. A fixed kernel technique is employed for the construction of interpolation functions and a collocation technique is employed for the discretization of the Stokes equations.

In a fixed kernel technique [17] for the construction of interpolation functions, an approximation (u^a) to an unknown (u) is given by

$$u^a(x) = \int_\Omega C(x, s)\varphi(x_K - s)u(s)\, ds. \tag{1}$$

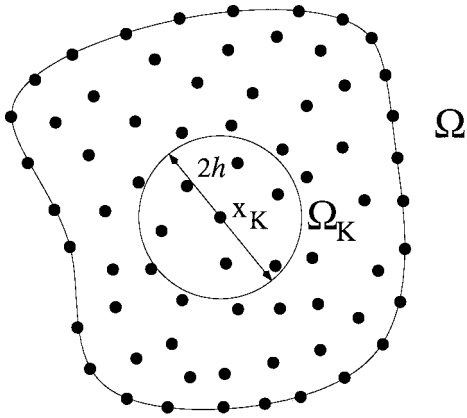

FIG. 2. A sample point distribution and the cloud defined by a kernel centered at one of the nodes is shown.

In this equation, $\varphi(x)$, is the kernel function, Ω is the computational domain, and x_K is a selected point in Ω where the kernel is centered. The cloud Ω_K is defined as the subdomain of Ω where the kernel function, centered at point K, is nonzero, i.e., $\Omega_K \subset \Omega \mid \varphi(x_K - x) \neq 0 \, \forall x \in \Omega_K$. The discretization of the simulation domain into scattered points and the centering of a kernel at point K to define subdomain Ω_K are shown in Fig. 2. In Eq. (1), $\mathcal{C}(x, s)$ is the correction function, which is defined as

$$\mathcal{C}(x, s) = \boldsymbol{P}^T(x_K - s)\boldsymbol{C}(x). \tag{2}$$

$\boldsymbol{P}^T(x) = \{p_1, p_2, \ldots, p_m\}$ is the $1 \times m$ vector of basis function and $\boldsymbol{C}^T(x) = \{c_1, c_2, \ldots, c_m\}$ is the $1 \times m$ vector of correction function coefficients. A quadratic basis in two dimensions is given by

$$\boldsymbol{P}^T(x) = [1, x, y, x^2, xy, y^2], \quad m = 6. \tag{3}$$

The unknown correction function coefficients, $\boldsymbol{C}(x)$, are determined by satisfying the consistency conditions, i.e.,

$$\int_\Omega \boldsymbol{P}^T(x_K - s)\boldsymbol{C}(x)\varphi(x_K - s)p_i(s)\,ds = p_i(x) \quad i = 1, 2, \ldots, m. \tag{4}$$

A discrete approximation of the above consistency conditions is written as

$$\sum_{I=1}^{NP} \boldsymbol{P}^T(x_K - x_I)\boldsymbol{C}(x)\varphi(x_K - x_I)p_i(x_I)\Delta V_I = p_i(x) \quad i = 1, 2, \ldots, m, \tag{5}$$

where NP is the total number of points in the domain Ω, and ΔV_I is the nodal volume associated with node I (see [17] for a discussion on nodal volumes). The consistency conditions summarized in Eq. (5) can be written in a matrix form as

$$\mathcal{M}\boldsymbol{C}(x) = \boldsymbol{P}(x), \tag{6}$$

where $\mathcal{M}$ is the $m \times m$ moment matrix and is a constant, i.e., it is not a function of x. In discrete form, the entries in the moment matrix are given by

$$\mathcal{M}_{ij} = \sum_{I=1}^{NP} p_j(x_K - x_I)\varphi(x_K - x_I)p_i(x_I)\Delta V_I. \tag{7}$$

Substituting the correction function coefficients into the discrete approximation of Eq. (1), we obtain

$$u^a(x) = \sum_{I=1}^{NP} \boldsymbol{P}^T(x)\mathcal{M}^{-T}\boldsymbol{P}(x_K - x_I)\varphi(x_K - x_I)u_I\Delta V_I. \tag{8}$$

Equation (8) can be rewritten as

$$u^a(x) = \sum_{I=1}^{NP} N_I(x)u_I, \tag{9}$$

where u_I is a nodal unknown for node I, and $N_I(x)$ is the fixed kernel interpolation function defined as

$$N_I(x) = \boldsymbol{P}^T(x)\mathcal{M}^{-T}\boldsymbol{P}(x_K - x_I)\varphi(x_K - x_I)\Delta V_I. \tag{10}$$

Since x_K can be any point in the computational domain, the interpolation functions $N_I(x)$ are multivalued. This issue is resolved by evaluating the interpolation function $N_I(x)$ only at the point x_K where the kernel is fixed (see [17] for complete details). The required interpolation functions can be calculated by fixing the kernel at every point in the domain.

Finite cloud method combines collocation with the fixed kernel interpolation to achieve a truly meshless or a scattered point solution of the governing partial differential equations. Consider the following model problem of a system of equations with l unknown variables

$$\mathcal{L}[u(x)] = f(x) \quad \text{in } \Omega \tag{11}$$

$$u = g(x) \quad \text{on } \Gamma_g \tag{12}$$

$$\frac{\partial u}{\partial n} = h(x) \quad \text{on } \Gamma_h, \tag{13}$$

where $\boldsymbol{\mathcal{L}}$ is a vector of differential operators operating on the unknown vector u. In the point collocation approach, the system of differential equations is satisfied at every node for the approximate solution $u^a(x)$. A system of algebraic equations is obtained by satisfying the governing equations at every point in the computational domain, i.e.,

$$\mathcal{L}_i[u^a(x_I)] = f_i(x_I) \quad x_I \in \Omega, i = 1, \ldots, l \tag{14}$$

$$u_i^a(x_I) = g_i(x_I) \quad x_I \in \Gamma_g, i = 1, \ldots, l \tag{15}$$

$$\frac{\partial u_i^a(x_I)}{\partial n} = h_i(x_I) \quad x_I \in \Gamma_h, i = 1, \ldots, l, \tag{16}$$

where $\mathcal{L}_i$ denotes the ith differential operator. u^a is approximated by meshless interpolation given in Eq. (9).

Time-independent Stokes equations are solved assuming that the nonlinear convection terms can be neglected. A stabilization term that modifies the continuity equation with the divergence of the momentum equation is considered (see [16] for details). Thus, the stabilized Stokes equations used in this study are written as

$$\mu \nabla^2 u - \nabla_p = 0 \tag{17}$$

$$\nabla \cdot u - \frac{h^2}{2} \nabla^2 p = 0, \tag{18}$$

where μ is the fluid viscosity, p is the pressure, u is the velocity vector, and h is referred to as the cloud size (see Fig. 2). More details on the solution of the fluid models with the finite cloud method can be found in [16].

3.2. DSMC Analysis of Gas Flow

DSMC is a statistical method in which a number of representative particles are traced in space and time. Using the simulated particles, the DSMC method samples the time evolution of the distribution function of a given system. The results are computed in the form of averages over the samples obtained. The simulation domain is discretized into cells for purposes of collision sampling and calculation of mean flow properties. Time is also discretized into small steps δT. During each time step, the particles first undergo free flight. The particles that hit a surface during the free flight are reflected according to a selected surface model. At the end of the time step, particle–particle collisions are sampled. Several models have been proposed [9] to determine the collision frequency and the distribution of the velocities after collision. We employ the variable-hard-spheres model for calculating particle cross sections and the no-time-counter method for sampling the collisions. More details on the implementation of the DSMC method for microfluidic filters can be found in [15].

The main steps in the DSMC algorithm are summarized in Algorithm 1. A description of the computations performed during each time step is provided in the second column. The algorithm is divided into two steps, showing that a number of DSMC time steps are performed to reach steady state. After steady state is established, further DSMC time steps are simulated to gather averages. The averages are estimated by summing the data from each time step into an array of variables (which are defined as *accumulators* in Algorithm 1) and then dividing by the number of time steps for which the data was collected. For the enforcement of self-consistent boundary conditions, the averages are collected for N_{step} DSMC time steps, and after each such period the boundary conditions are updated as discussed below. After the steady state is reached, the accumulators are saved to a file for postprocessing before they are reset.

The enforcement of boundary conditions in DSMC simulations is a topic that has received recent attention in the literature [18]. For the enforcement of boundary conditions in low-velocity flows, a recent work uses extrapolation of the variables at the input, and a flux corrected updating method at the output [19]. In our earlier work [15], we found that a simple extrapolation at the output is adequate, i.e., when it is necessary to self-consistently update the boundary conditions extrapolation of the following form

ALGORITHM 1. Key steps in DSMC technique for steady-state simulation:

initialize DSMC particle states
initialize DSMC boundary conditions
while not *converged*
  Simulate N_{step} DSMC time steps.
  update boundary conditions
  check for convergence
  reset accumulators
end while
while statistical noise > limit
  Simulate N_{step} DSMC time steps.
  update boundary conditions
  estimate the noise in averages
  save accumulators to file
  reset accumulators
end while

A DSMC time step includes:
inject particles
update particle positions
implement collisions
accumulate averages

is used:

$$u_i = u_c \tag{19}$$

$$p_i = p_c. \tag{20}$$

Here, u_i, p_i are the interface values and u_c and p_c are the values obtained from the cells neighboring the interface. For implementation of the boundary conditions a buffer zone approach is used. Figure 3 depicts an inflow boundary with a row of buffer cells that lie outside of the simulation domain. In the buffer cells, a Maxwellian distribution is created at each time step with the parameters p_i, u_i, v_i for pressure, x-velocity, and y-velocity, respectively. The created particles are moved for one time step. The particles that remain in the buffer cells, and those that cross into the buffer cells through Γ are deleted. The cells that lie within a given distance from the interface in the simulation domain are called the estimation cells. For enforcing self-consistent boundary conditions, the averages from the estimation cells, p_c, u_c, v_c, are extrapolated to the boundary. For the case of a coupled

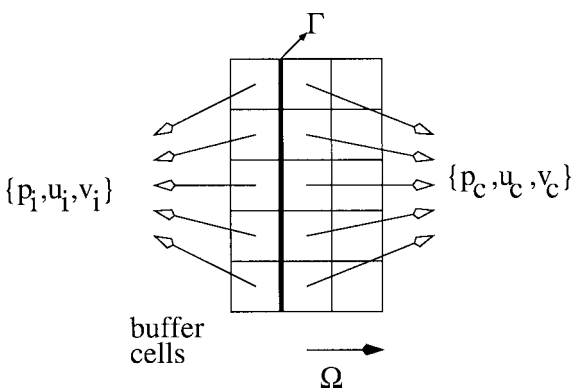

FIG. 3. Placement of buffers cells and the update cells to enforce boundary conditions in DSMC.

simulation the values p_c, u_c, v_c are passed on the continuum domain as discussed in the next section.

3.3. Coupled Stokes/DSMC

A high-level description of the coupled Stokes/DSMC approach is shown in Algorithm 2. Given an arbitrary initial state and a set of boundary conditions along the overlapping interfaces, a Schwarz technique is implemented to find a self-consistent solution to the Stokes and the DSMC subdomains. Self-consistency is determined by a convergence check which requires that the updates to the solution (for example, pressure and the velocities) be less than a specified tolerance value. Self-consistency also ensures that the boundary conditions at the interface have converged to the specified tolerance. After the convergence of the coupling iterations, several coupling iterations between Stokes and DSMC subdomains are performed and the DSMC results are saved to a file for a postprocessing step. The final results in the DSMC subdomain are obtained as an average of the samples collected. As a last step, the final results from the DSMC subdomains are used as boundary conditions

ALGORITHM 2. Description of Stokes/DSMC coupling in various overlapping Schwarz methods.

Main loop.
- initialize DSMC particle states
- initialize DSMC boundary conditions
- **while** coupling iterations not converged*
  - *Do coupling_iteration*
  - *check for convergence*
- **end while**
- *Start saving accumulators*
- **while** statistical noise > limit
  - *Do coupling_iteration*
  - *estimate the noise in averages*
- **end while**
- *Compute averages from saved accumulators*
- Interpolate DSMC to Stokes
- **for** each Stokes subdomain S_i **do**
  - Solve Stokes equations in S_i
- **end for**

* Convergence requires the convergence of the coupling iterations.

coupling_iteration

serial alternating Schwarz
- **for** each DSMC subdomain D_i **do**
  - Simulate N_{step} DSMC time steps in D_i
- **end for**
- Interpolate DSMC to Stokes
- Reset accumulators
- **for** each Stokes subdomain S_i **do**
  - Solve Stokes equations in S_i
- **end for**
- Interpolate Stokes to DSMC

coupling_iteration

colored alternating Schwarz
- Make N_{step} time steps in all D_i
- Interpolate DSMC to Stokes
- Reset accumulators
- Solve Stokes in all S_i
- Interpolate Stokes to DSMC

coupling_iteration:

parallel alternating Schwarz
- Make N_{step} time steps in all D_i
- Solve Stokes in all S_i
- Interpolate DSMC to Stokes
- Reset accumulators
- Interpolate Stokes to DSMC

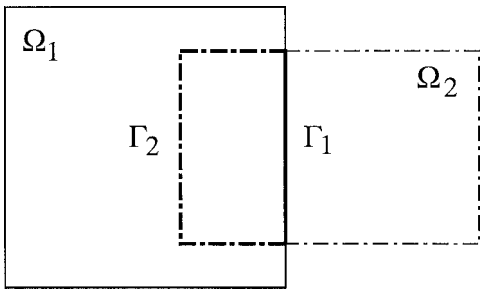

FIG. 4. Decomposition of a sample geometry into two overlapping subdomains.

to find the solution in the continuum subdomains. It needs to be noted that an imporatant property of this method is that the DSMC noise in the estimates interpolated to the continuum subdomain at each coupling iteration is larger than the DSMC noise in the final result, which is an average over a number of coupling iterations. The interpolation of the final result to the continuum subdomains and a solution of the continuum equations are performed to obtain the continuum solutions using these low noise boundary conditions.

3.3.1. *Schwarz Technique*

A Schwarz technique is employed to solve the coupled Stokes/DSMC problem on overlapping subdomains. To understand the Schwarz technique, consider two overlapping subdomains as shown in Fig. 4. An alternating Schwarz method for the subdomains shown in Fig. 4 can be summarized as follows:

Begin : $n = 0$; $u_2^{(0)}|_{\Gamma_1}$ = initial condition

Repeat :{ $n = n + 1$

Solve $Lu_1^{(n)} = f_1$ in Ω_1 with BC $u_1^{(n)} = u_2^{(n-1)}$ on Γ_1

Solve $Lu_2^{(n)} = f_2$ in Ω_2 with BC $u_2^{(n)} = u_1^{(n)}$ on Γ_2

} until convergence

where n is the iteration number, $u_i^{(n)}$ is the solution in domain Ω_i at iteration n, L is the partial differential operator describing the governing equations, and f_i are forcing functions of position in domain Ω_i. In the alternating Schwarz method, the subdomains are overlapped and Dirichlet type boundary conditions are employed. The alternating Schwarz method can be modified to use nonoverlapping domains and Neumann type boundary conditions. Several variations of the basic Schwarz technique are explained in [20]. For example, previous work on coupling time-dependent Navier–Stokes with DSMC employed a nonoverlapped Schwarz coupling method with Robin (mixed) type boundary conditions [6]. In this work, we employ only Dirichlet–Dirichlet type boundary conditions to avoid estimating derivatives of flow quantities.

The alternating Schwarz method as described above is a serial technique. In Algorithm 2, the serial alternating Schwarz method and two parallel implementations of the Schwarz technique for coupling Stokes and DSMC subdomains are shown. To understand the implementation of various Schwarz techniques, consider the geometry and its decomposition

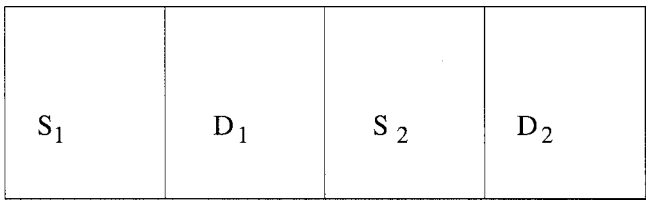

FIG. 5. Decomposition of a sample geometry into four subdomains.

shown in Fig. 5. S_i denotes the Stokes subdomain and D_i denotes the DSMC subdomain. In the serial alternating Schwarz method, each subdomain is solved sequentially, i.e., S_1, followed by D_1, followed by S_2 and so on. In a colored Schwarz method the subdomains are divided into groups (i.e., colored) and the subdomains in each group are solved concurrently, while each group is solved sequentially. The optimal coloring depends on the geometry and its decomposition. For example, for the subdomains shown in Fig. 5, all the D_i's are assigned one color and all the S_i's are assigned a different color. All the D_i's are solved at once, followed by the solution of all the S_i's. In a parallel alternating Schwarz method, all the subdomains are solved at once, i.e., the subdomains S_1, S_2, D_1, D_2 are all solved concurrently. In this paper, we primarily investigate the colored Schwarz method for microfilter analysis. However, for one of the examples considered in this paper, we compare the convergence characteristics of the colored and the parallel Schwarz methods.

For the operation of microfluidic filters under pressure gradients, it is important to impose proper boundary conditions at interfaces to avoid convergence problems. In order to simulate pressure driven flows, pressure must be exchanged between the subdomains at least in one direction. The stability of the coupled method for pressure driven flows is analyzed in the Appendix. It is shown that stability depends on the geometry of the problem and the choice of the boundary conditions transferred to the Stokes subdomain. For the microfluidic filter example considered in this work, either pressure or velocity can be transferred as a boundary condition from the DSMC subdomain to the Stokes subdomain. Because of the stability concerns associated with transfering a pressure boundary condition to the Stokes subdomain, we transfer velocity as a boundary condition from DSMC to the Stokes subdomain. Both pressure and velocity are transferred as boundary conditions from the Stokes subdomain to the DSMC subdomain.

3.3.2. *Interpolation between Domains*

In general, the position of the continuum nodes and the DSMC particles do not match in the overlapped regions. When a solution is computed in the DSMC subdomain, the solution for the continuum nodes in the DSMC subdomain can be obtained by using an interpolation scheme. This interpolation scheme is referred to as the DSMC to continuum interpolation. Similary, when a solution in the Stokes subdomain is computed, the solution for the DSMC particles (or DSMC cell centers) in the Stokes subdomain can again be computed by an interpolation scheme. This interpolation scheme will be referred to as the continuum to DSMC interpolation. Both continuum to DSMC and DSMC to continuum interpolation schemes are implemented by using the scattered point interpolation technique described in Section 3.1.

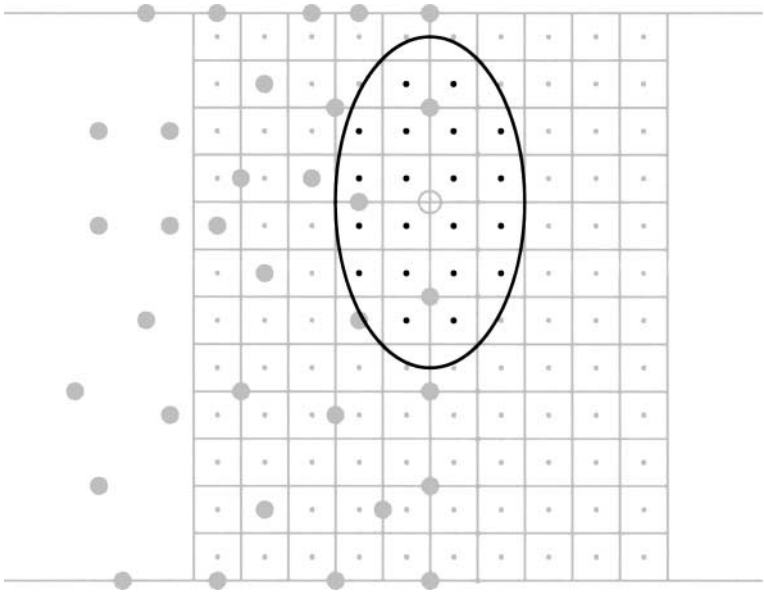

FIG. 6. Placement of the kernel and the definition of the cloud for DSMC to continuum interpolation.

DSMC to continuum interpolation. Denote by c_i the DSMC cells that will be used to compute the solution at the continuum boundary nodes n_j. Let x_{ci} denote the position of the DSMC cell center and x_{nj} denote the position of a continuum boundary node at which the solution needs to be interpolated. Let u_{ci} denote the nodal parameters for the solution at the DSMC cell centers (the solution that needs to be interpolated can be pressure, velocity, etc.) and $u(x_{nj})$ be the interpolated solution at the continuum boundary node n_j. To compute $u(x_{nj})$, a kernel or a weighting function is first centered at the position x_{nj}. The kernel centered at x_{nj} defines the cloud Ω_j and the number of DSMC cell centers, NC, that lie within the cloud Ω_j (see Fig. 6 for the definition of the cloud and the cells that fall within the cloud). When the kernel is centered at x_{nj}, the interpolation functions, $N_i(x_{nj})$, $i = 1, 2, \ldots, NC$, are computed. Once the interpolation functions are computed, $u(x_{nj})$ is computed by

$$u(x_{nj}) = \sum_{i=1}^{NC} N_i(x_{nj}) u_{ci}. \tag{21}$$

Continuum to DSMC interpolation. Let n_i denote the nodes in the continuum domain and c_j denote the DSMC cells. Let x_{ni} and x_{cj} denote the position of the continuum nodes and the DSMC cells, respectively. Let u_{ni} denote the nodal parameter for the solution at the continuum node with location x_{ni} and $u(x_{cj})$ be the interpolated solution at the DSMC cell center with location x_{cj}. To compute $u(x_{cj})$ a kernel or a weighting function is centered at the position x_{cj}. The kernel centered at x_{cj} defines the cloud Ω_j and the number of continuum nodes, NP_j, that lie within the cloud Ω_j (see Fig. 7 for the definition of the cloud and the continuum nodes that fall within the cloud). When the kernel is centered at x_{cj}, the interpolation functions, $N_i(x_{cj})$, $i = 1, 2, \ldots, NP_j$, are computed. Once the interpolation

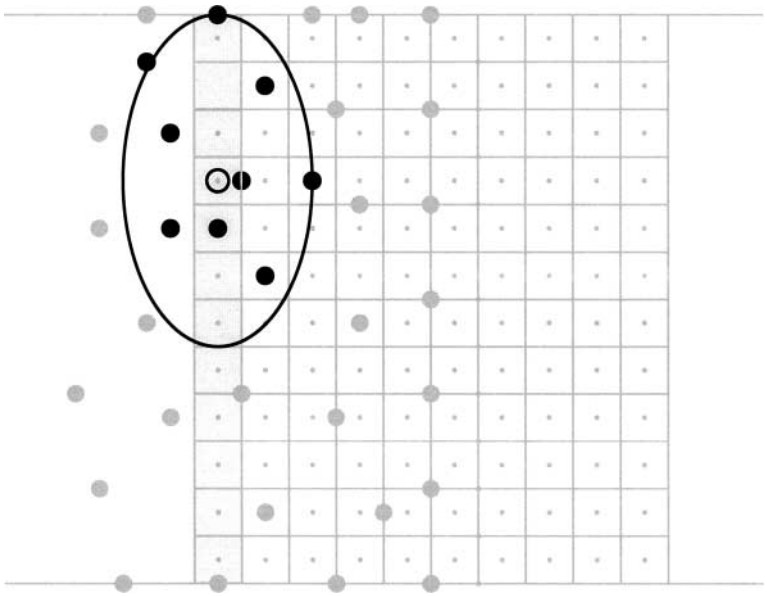

FIG. 7. Placement of the kernel and the definition of the cloud for continuum to DSMC interpolation.

functions are computed, $u(x_{cj})$ is computed by

$$u(x_{cj}) = \sum_{i=1}^{NP_j} N_i(x_{cj}) u_{ni}. \tag{22}$$

4. RESULTS

The multiscale or the coupled DSMC/Stokes approach described in the previous sections is applied to the microfilter geometry shown in Fig. 8. Two filters with different dimensions are considered: For both filters $h_f = 5\ \mu$m, $l_c = 1\ \mu$m is used. The first filter has a smaller

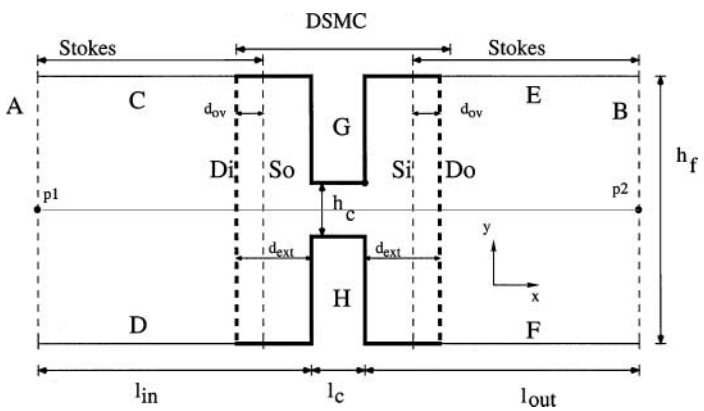

FIG. 8. The geometry of the microfilter device. Also shown in the figure are the Stokes and DSMC subdomains and the overlap between the two subdomains.

TABLE I
A Summary of Boundary Conditions on Various Surfaces of the Microfilter Geometry

	Pressure	x-velocity	y-velocity
Surface A	1.3 atm	—	0
Surface B	1.0 atm	—	0
Surface C, E	$\partial P/\partial y = 0$	0	$\partial v_y/\partial y = 0$
Surface D, F	$\partial P/\partial y = 0$	0	$\partial v_y/\partial y = 0$
Surface Si, So	—	v_x = DSMC estimate	—
Surface Di, Do	p = Stokes solution	v_x = Stokes solution[a]	v_y = Stokes solution[a]
Surface G, H	—	diffusive	diffusive

[a] v_x and v_y are not imposed as boundary conditions, but rather used to set the mean velocities of particles crossing into the DSMC subdomain from the buffer cells.

filter height of $h_c = 0.2$ μm, and the second has a filter height of $h_c = 0.8$ μm. For the filter with $h_c = 0.2$ μm, $l_{in} = 7$ μm and $l_{out} = 7$ μm is used. For the filter with $h_c = 0.8$ μm, $l_{in} = 6$ μm and $l_{out} = 8$ μm is used. Figure 8 also shows the decomposition of the filter geometry into Stokes and DSMC subdomains. The extension of the DSMC subdomain on each side of the channel is denoted by d_{ext}. Observe that for this example, we have one DSMC subdomain and two Stokes subdomains. In order to make sure that the flow is approximately incompressible at the interface, $d_{\text{ext}} = 2$ μm and $d_{\text{ext}} = 3$ μm are used for $h_c = 0.2$ and $h_c = 0.8$ μm, respectively. The overlap between DSMC and the Stokes subdomains is denoted by d_{ov}. The overlap is measured from the center of the DSMC estimation cells to the continuum nodes, i.e., the generation cells are not counted in the overlap as these cells do not have valid data that can be used. An identical overlap distance, d_{ov}, is used for both the input and the output regions. The initial state and the boundary conditions for the DSMC subdomain are selected far away from the expected steady state solution in order to test the convergence characteristics of the coupled approach. The boundary conditions imposed on various surfaces of the microfilter geometry are listed in Table I. The initial pressure is set to 5.0 atm and the initial velocity is set to 0 m/s for the DSMC subdomain. For all the simulations reported in this paper, a DSMC time step of 10 ps is used.

To test the accuracy of the coupled DSMC/Stokes method, the geometry shown in Figure 8 is simulated by DSMC only (which will serve as the exact solution) and by the coupled approach for two different values of $h_c = 0.2$ μm and $h_c = 0.8$ μm. For the DSMC only simulation and for the DSMC subdomain of the coupled approach, the same cells sizes are used. The cell size is set to be 20 nm for the input and output regions and 10 nm for the channel. For the filter with $h_c = 0.2$ μm, the number of particles per cell at the input region is about 38, whereas for the filter with $h_c = 0.8$ μm, this number is about 19. $N_{\text{step}} = 5000$ is used for the coupled analysis. Ten coupling interations were performed before the averaging process was started. This corresponds to a total 50e3 DSMC time steps, which in turn is 0.5 μs of simulated time. After the convergence of the coupled process, the averages were collected for 1 μs, which corresponds to 100e3 DMSC time steps. For the DSMC simulations, the averaging was started after 1.5 μs was simulated and the averages were collected for 1 μs. Note that 1.5 μs of simulated transient may have been longer than the minimum necessary. The CPU time for the DSMC simulation of the filter with $h_c = 0.2$ μm and $h_c = 0.8$ μm are 1340 and 640 hours, respectively.

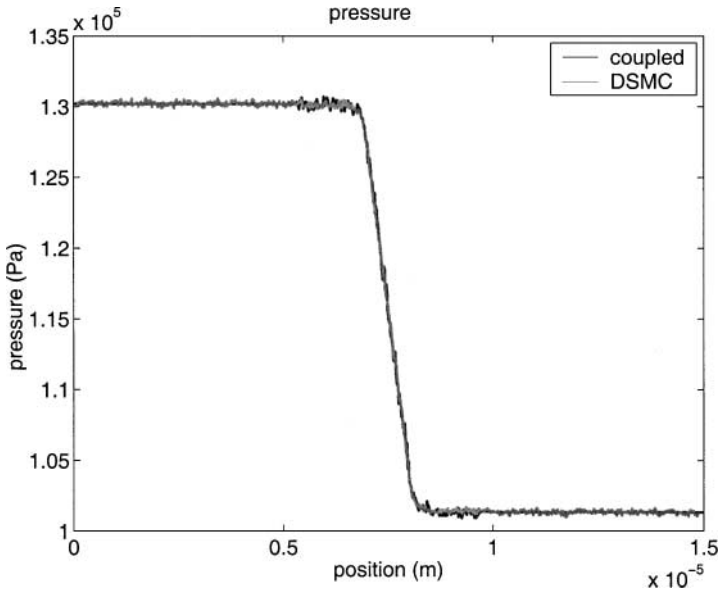

FIG. 9. Comparison of pressure along the midline of the channel obtained from the DSMC only and coupled simulations ($h_c = 0.2$ μm, $d_{ov} = 0$ μm).

The results for the two filters are shown in Figs. 9–13 with an overlap of zero, i.e., $d_{ov} = 0$. Figures 9 and 10 compare the pressure and the x-velocity, respectively, obtained with the DSMC and the coupled simulations for $h_c = 0.2$ μm. Figures 11 and 12 compare the pressure and the x-velocity, respectively, obtained with the DSMC and the coupled

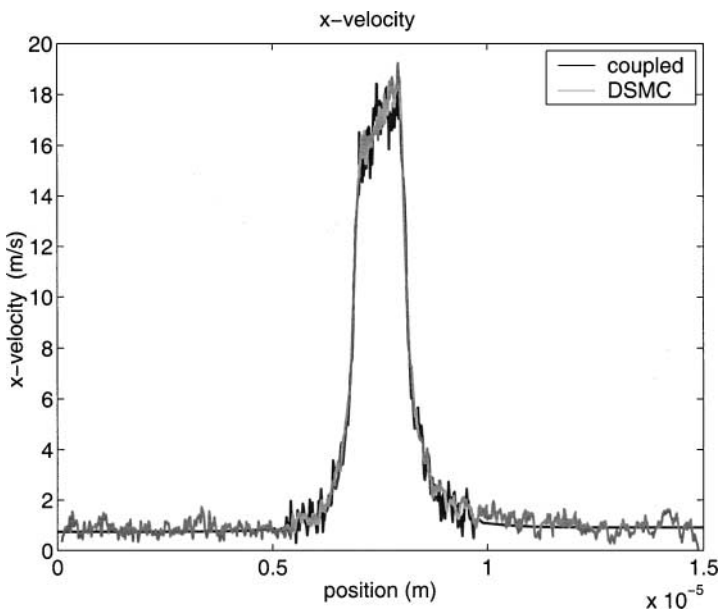

FIG. 10. Comparison of velocity along the midline of the channel obtained from the DSMC only and coupled simulations ($h_c = 0.2$ μm, $d_{ov} = 0$ μm).

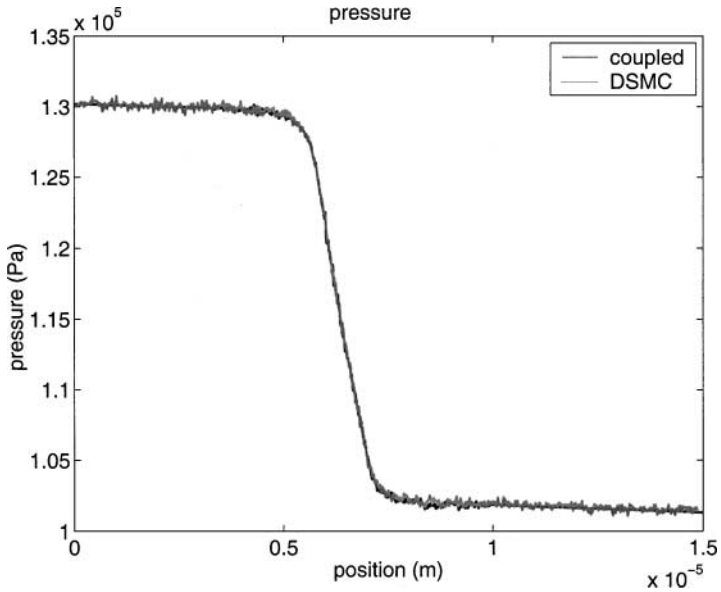

FIG. 11. Comparison of pressure along the midline of the channel obtained from the DSMC only and coupled simulations ($h_c = 0.8\ \mu$m, $d_{ov} = 0\ \mu$m).

simulations for $h_c = 0.8\ \mu$m. We observe that for both filter heights, the coupled simulations are in good agreement with the DSMC results. The good agreement between the two methods establishes the accuracy of the coupled method. From the results, we can also conclude that the multiscale approach presented in this paper achieves proper coupling between the DSMC and the Stokes subdomains. Figure 13 shows the x-velocity along the

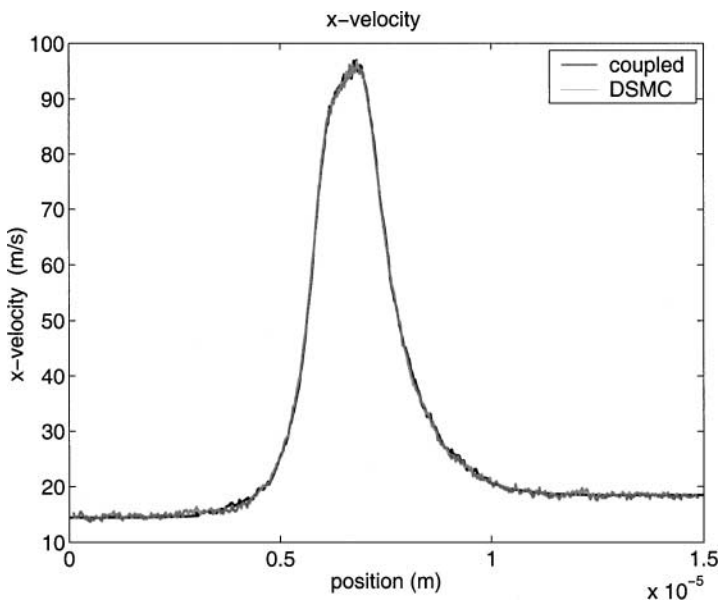

FIG. 12. Comparison of velocity along the midline of the channel obtained from the DSMC only and coupled simulations ($h_c = 0.8\ \mu$m, $d_{ov} = 0\ \mu$m).

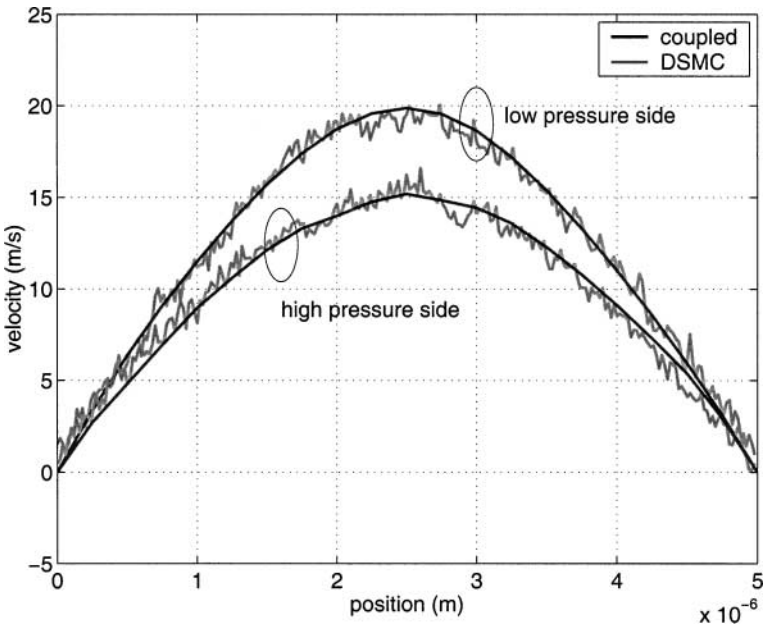

FIG. 13. Comparison of coupled and DSMC results for x-velocity along the DSMC/Stokes interface ($h_c = 0.2\ \mu$m, $d_{ov} = 0\ \mu$m). The high pressure side is the interface at x = 3.05 μm, and the low pressure side is the interface at x = 9.95 μm.

DSMC/Stokes interface for the filter with $h_c = 0.8\ \mu$m. From this figure, it is seen that the coupling between the Stokes and the DSMC subdomains is two-dimensional as the x-velocity exhibits significant variation in the direction perpendicular to the flow.

To characterize the efficiency of the coupled approach, we introduce an ideal speed-up factor S, which is defined as

$$S = \frac{Nt(\text{DSMC})}{Nt(\text{coupled})} \frac{Np(\text{DSMC})}{Np(\text{coupled})}, \tag{23}$$

where Nt(DSMC) and Nt(coupled) are the total number of time steps simulated for the DSMC simulation and the coupled simulation, respectively. Similarly, Np(DSMC) and Np(coupled) are the total number of particles for the DSMC simulation and the coupled simulation, respectively. This definition ignores the overhead (e.g., continuum to DSMC and DSMC to continuum interpolations) associated with the coupling procedure as well as the time required to solve the continuum Stokes equations. Thus, the ideal speed-up defined above could be unrealistic and may never be attained by the coupled approach. However, on parallel computers, it is possible to get larger than ideal speed-ups because of the memory usage. For example, a smaller memory usage can enhance performance because of the reduction in the ratio of cache misses to hits. Thus, if the overhead associated with the coupled approach as well as the time spent in solving the continuum equations is small, then it is possible to get near ideal speed-ups because of performance gains attained on parallel computers. The speed-up factors reported in this paper should be understood in proper context by considering the above remarks.

The performance of the coupled method is measured by comparing the CPU times required by the DSMC and by the coupled approach. In order to simplify the comparison,

both simulations were run for the same number of DSMC time steps (i.e., $Nt(\text{DSMC}) = Nt(\text{coupled})$). Speed-up comparisons were performed for both the smaller filter ($h_c = 0.2\ \mu$m) and for the bigger filter ($h_c = 0.8\ \mu$m). The ideal speed-up factors for the smaller and the bigger filters are 3.25 and 2.18, respectively. These speed-ups are computed by assuming $Nt(\text{DSMC}) = Nt(\text{coupled})$. The observed speed-ups for the two filters are 3.2 and 2.17, respectively, demonstrating an efficiency greater than 99%.

4.1. Effect of Various Parameters on Convergence

An important property to investigate of the multiscale approach is its convergence behavior. The dependence of the convergence on various parameters contained in the multiscale approach needs to be understood to guarantee the convergence of the multiscale approach. In this paper, we report on the dependence of the convergence characteristics on the overlap size, d_{ov}, the DSMC particle weight, w_p, and the number of DSMC time steps between coupling iterations, N_{step}. The details on each of each these studies is described below.

4.1.1. *Overlap Size*

The overlap size in Fig. 8, d_{ov}, describes the overlap between the Stokes and the DSMC subdomains i.e., both DSMC and Stokes equations are solved in the overlap region. The dependence of the convergence on the overlap is investigated by considering various overlap sizes: $d_{\text{ov}} = 0.6, 0.4, 0.2, 0.0\ \mu$m. The overlap is increased or decreased by adjusting the distance d_{ext} (see Fig. 8). The implementation of the interface boundary conditions for the $d_{\text{ov}} = 0.0\ \mu$m case is not different from the other d_{ov} cases.

The convergence results are summarized in Figs. 14 to 16. Figures 14 and 15 show the convergence of the pressure and the velocity boundary conditions, respectively, at the

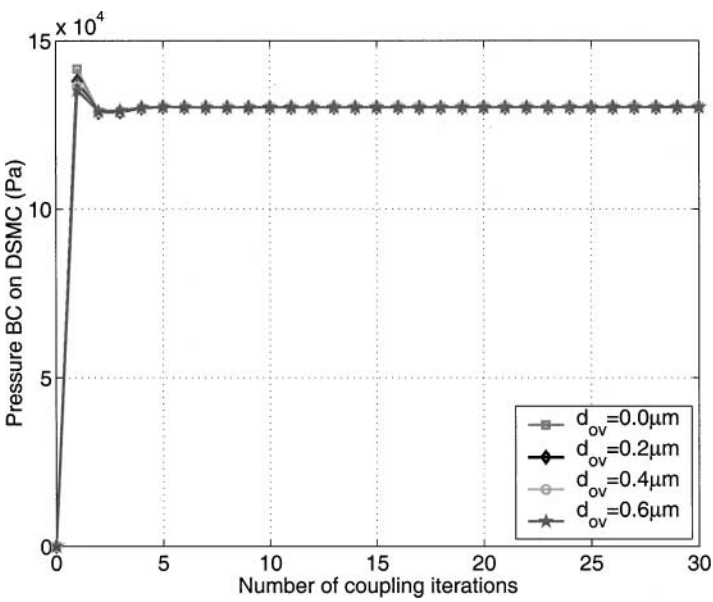

FIG. 14. Convergence of pressure boundary condition transferred from the continuum side to the DSMC subdomain for different overlaps. $N_{\text{step}} = 5000$ is used.

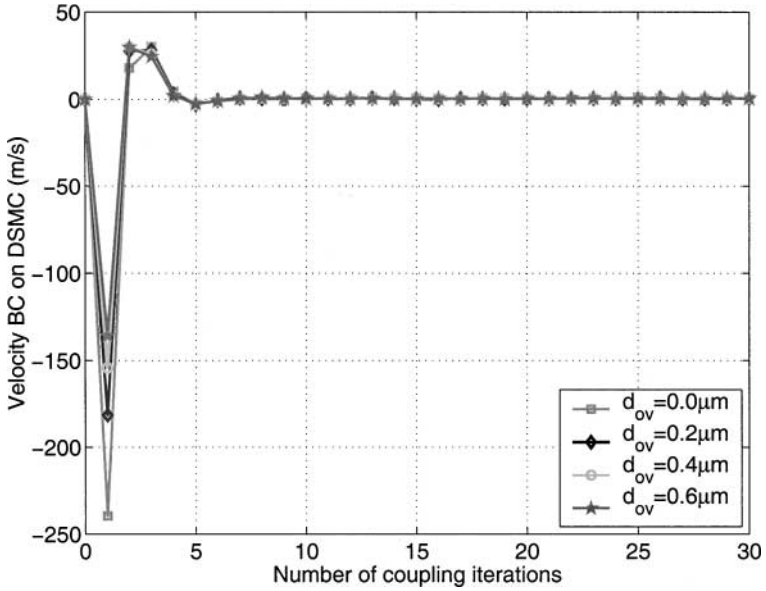

FIG. 15. Convergence of velocity boundary condition transferred from the continuum side to the DSMC subdomain for different overlaps. $N_{\text{step}} = 5000$ is used.

input section. The pressure boundary condition in Fig. 14 is plotted by computing the average of the boundary condition imposed on all the buffer cells. The velocity boundary condition in Fig. 15 is plotted by taking the value from the buffer cell at $h_f/2$. The results in Figs. 14 and 15 indicate that the convergence is weakly dependent on the overlap size for this problem, with larger overlap size exhibiting slightly better convergence. In order to further investigate the dependence of the convergence characteristics on the overlap size, we show the absolute error in the pressure boundary condition transferred from the Stokes subdomain to the DSMC subdomain in Fig. 16. The absolute error is defined as the absolute deviation between the computed value and an exact value, which is determined from the DSMC only simulation. Again, the values plotted are the average of all the DSMC buffer cells. The results in Fig. 16 were obtained for $N_{\text{step}} = 5000$. The results in Fig. 16 show that the number of iterations until convergence is weakly dependent on the overlap size. A mathematical analysis of the Schwarz method reveals that a faster convergence should be obtained for a larger overlap [20]. However, the result in Fig. 16 indicates clearly that the number of coupling iterations is only weakly dependent on the overlap size. The observed behaviour leads to the conclusion that the Stokes and the DSMC subdomains are weakly coupled.

4.1.2. *Particle Weight*

In statistical approaches, such as DSMC, noise in the solution is a typical characteristic and should be given careful consideration. For example, the presence of noise in the DSMC solution can create a variety of problems to the convergence of the multiscale approach. The noise in the DSMC estimates can be controlled by decreasing the particle weight or by increasing the number of DSMC time steps. To evaluate the dependence of convergence on the particle weight, w_p, simulations are performed using particle weights of 25e4, 5e4, 1e4,

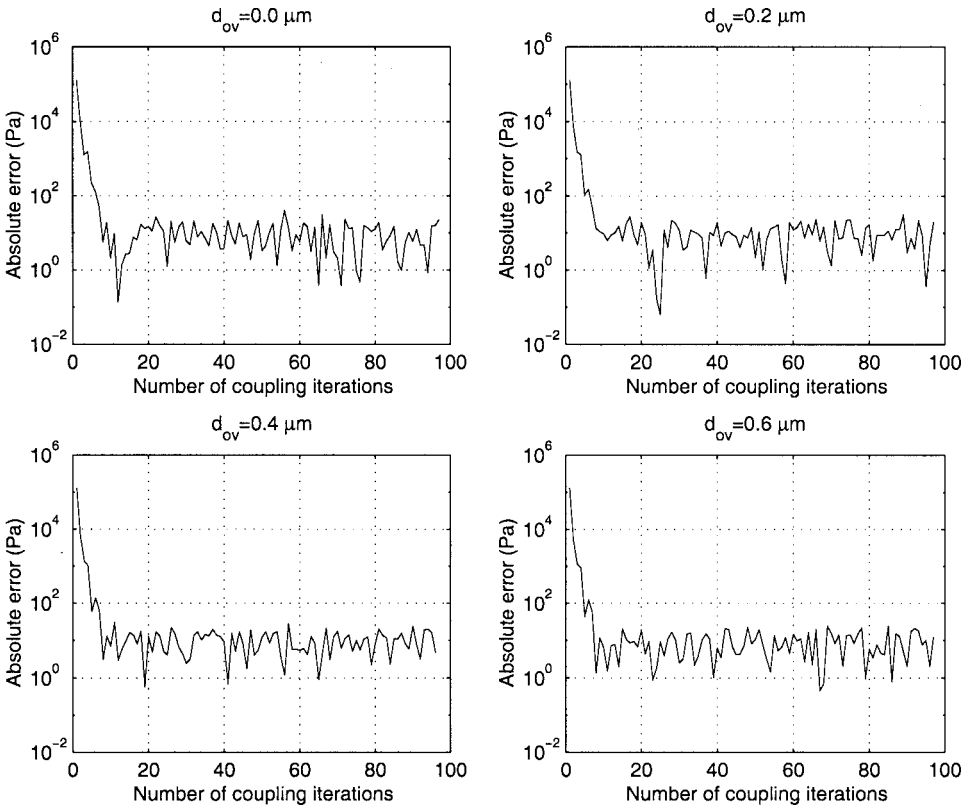

FIG. 16. The absolute error in the pressure boundary condition transferred from the continuum subdomain to the DSMC subdomain for different overlaps. $N_{\text{step}} = 5000$ is used.

and 1e3. The absolute error is plotted on a logarithmic scale in Fig. 17. The result in Fig. 17 indicates that a smaller particle weight accounts for more number of particles and exhibits better convergence characteristics with less noise. Since the convergence is not delayed significantly because of a larger noise (due to a larger particle weight), we can conclude that the method is fairly robust. The observation that the coupled method converges when using a large particle weight is important for application of the particle cloning method [21] to speed up the DSMC simulation. The particle weight that should be selected according to the DSMC accuracy requirements is 1e3. However, a lower weight requires extensive CPU times and the simulation time can be reduced by using a particle cloning method that starts with 16 times the desired weight of 1e3 and using four cloning steps to get the desired accuracy.

4.1.3. *Number of Time Steps in DSMC*

The selection of the number of DSMC time steps, N_{step}, during each coupling iteration is important for the efficiency of the coupled method. Two issues need to be considered when selecting N_{step}. The first issue deals with the noise in the DSMC solution. The noise considerations that were discussed in connection with the particle weight also apply for the selection of N_{step}. In the simulations reported here, the particle weight, w_p, is reduced proportionally as N_{step} is decreased. The second issue that needs to be considered in selecting

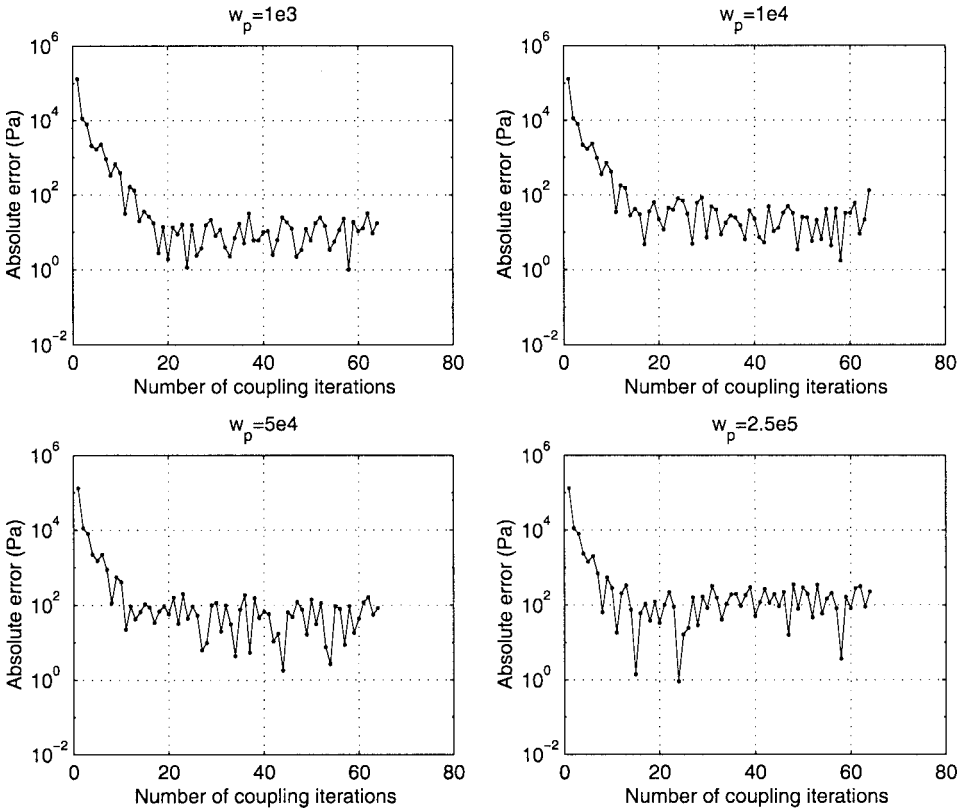

FIG. 17. The absolute error in the pressure boundary condition transferred from the continuum subdomain to the DSMC subdomain for different values of w_p. $N_{\text{step}} = 1000$ is used.

N_{step} is the time-dependent nature of the DSMC solution computed during the coupling iterations. Starting from an initial state, the multiscale coupled algorithm will take a certain number of iterations to compute a converged solution. During each iteration, the flow in the DSMC subdomain will evolve in a time-accurate manner toward a steady-state solution determined by the boundary conditions. If N_{step} is large enough during each coupling iteration, a steady-state solution can be reached. However, there is no need to compute steady-state solution during each coupling iteration as the boundary conditions enforced on the DSMC subdomain are not necessarily steady-state boundary conditions. Since the goal is to compute a steady-state solution for the entire system (including both Stokes and DSMC subdomains), N_{step} can be selected shorter and the boundary conditions can be updated in an iterative manner until a steady-state solution is reached.

To investigate the effect of N_{step}, the filter geometry is simulated by keeping N_{step}/w_p constant as N_{step} is changed. This keeps the noise in the DSMC estimates at the same level. N_{step} values of 200, 1000, 5000, and 25000 are investigated. The convergence of the pressure boundary condition in the input section is plotted against the number of iterations in Fig. 18. It can be observed that for $N_{\text{step}} = 200$ and 1000, a larger number of coupling iterations are needed for convergence when compared to $N_{\text{step}} = 5000$ and 25,000. A comparison of the total simulated DSMC time steps until convergence shows that for $N_{\text{step}} = 200$ and 1000, the total DSMC time steps are approximately equal. However,

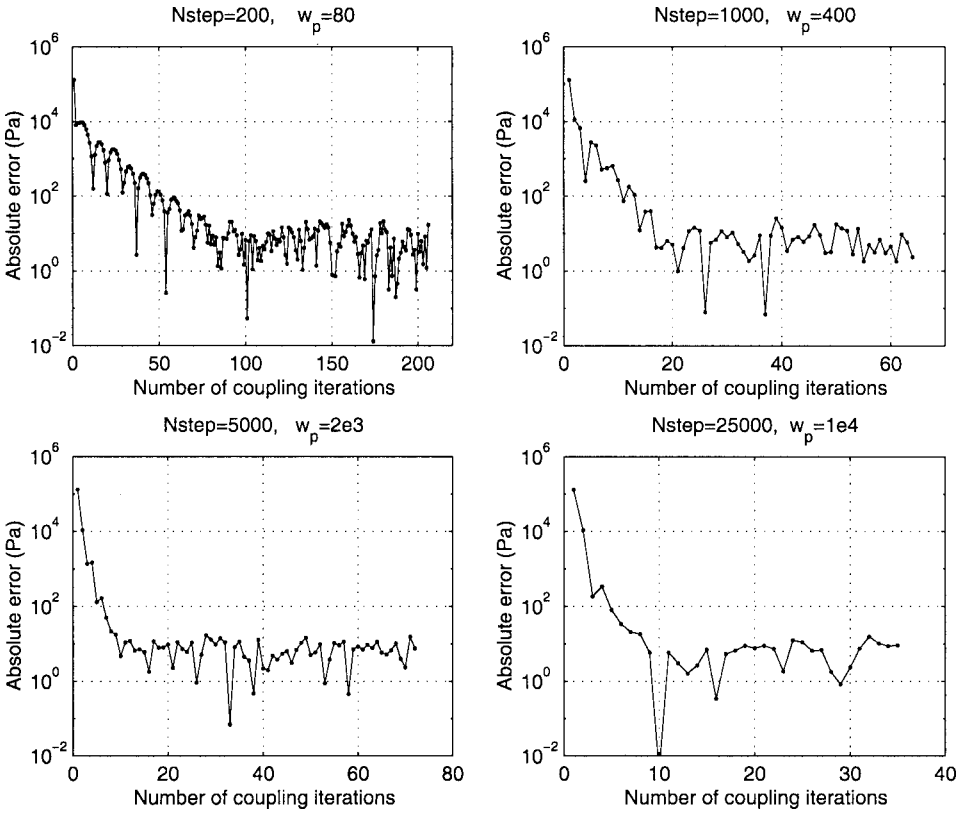

FIG. 18. The absolute error in the pressure boundary condition transferred from the continuum subdomain to the DSMC subdomain for different values of N_{step} and w_p. w_p is changed to keep the noise level approximately the same.

for the other two cases, the DSMC time steps until convergence are much larger. Thus, we can conclude that, if $N_{\text{step}} * N_{\text{cpl}}$ (where N_{cpl} is the number of coupling iterations until convergence) is longer than the number of time steps the DSMC subdomain takes to steady state, convergence is determined by the properties of the coupling method, where as if $N_{\text{step}} * N_{\text{cpl}}$ is smaller then the DSMC subdomain will evolve in a quasi-static manner and the number of coupling iterations until convergence, N_{cpl}, will be increased. For an efficient implementation, $N_{\text{step}} * N_{\text{cpl}}$ should be close to the time constant of the DSMC subdomain.

4.1.4. *Comparison of Colored and Parallel Schwarz Methods*

All the results shown in the previous sections were obtained by using a colored Schwarz method, where the DSMC and Stokes subdomains are solved alternatively as described in algorithm 2. The colored Schwarz method is a reasonably good solution scheme for the examples considered in this paper as the solution of the Stokes subdomain takes relatively much shorter time compared to the solution of the DSMC subdomain. However, for more complicated flows, where the solution of the continuum region can be quite involved, the sequential solution of the continuum and the DSMC subdomains can be inefficient. In such cases, a parallel Schwarz method, where both the continuum and the DSMC subdomains are solved simultaneously, is an attractive approach.

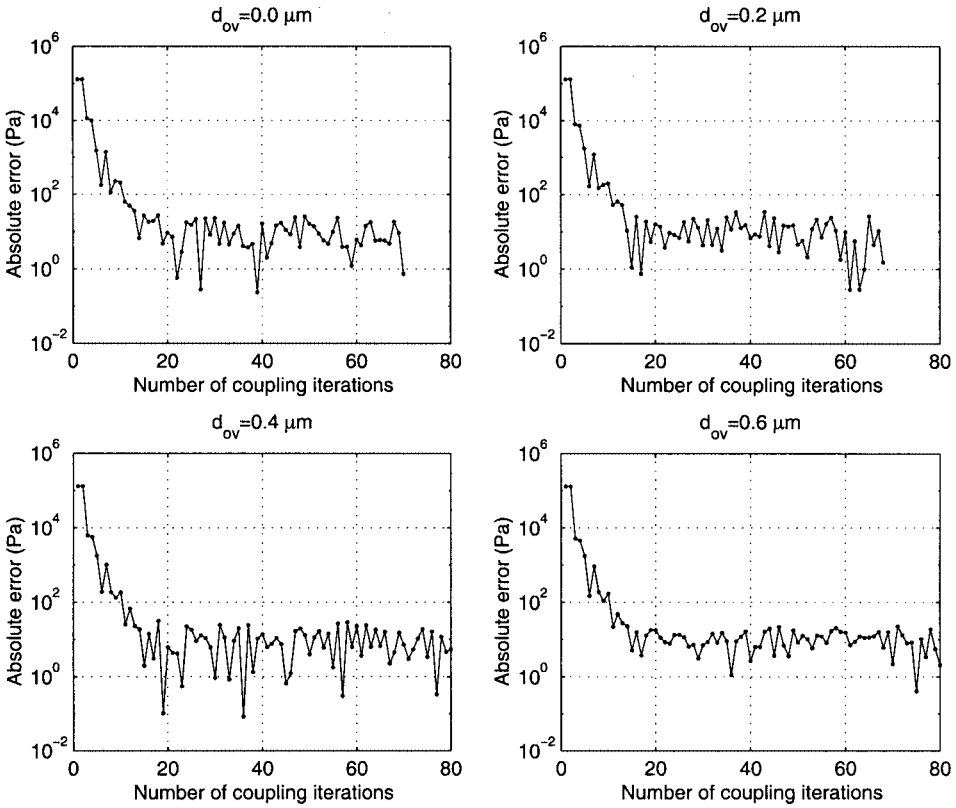

FIG. 19. The absolute error in the pressure boundary condition transferred from the continuum subdomain to the DSMC subdomain for different overlaps. The parallel Schwarz method is employed and $N_{\text{step}} = 5000$ is used.

The convergence of the colored Schwarz method for different overlap sizes was already discussed in Fig. 16. The convergence of the parallel Schwarz method for different overlap sizes is shown in Fig. 19. The results indicate that the parallel and colored coupling techniques exhibit similar convergence characteristics. The convergence is again weakly dependent on the overlap size. Thus, for filter simulations, the use of either colored Schwarz or a parallel Schwarz approach produces good efficiency.

4.2. Application of the Method

4.2.1. *Filter Element with Random Point Distribution*

The multiscale approach described in this paper allows for flexibility in the treatment of the interface between the Stokes and the DSMC domains. In particular, the continuum domain can be simulated by a random distribution of points and the continuum points don't need to match with the DSMC cells. As discussed in Section 3.3.2, the solution from one subdomain can be interpolated to the other subdomain by using a scattered point interpolation technique. To demonstrate the flexibility of the method, the microfilter element is simulated by using a random distribution of points for the Stokes domain. The filter element along with the point distribution is shown in Fig. 20. The filter is again simulated by using the colored Schwarz method and by employing elliptical and circular clouds for

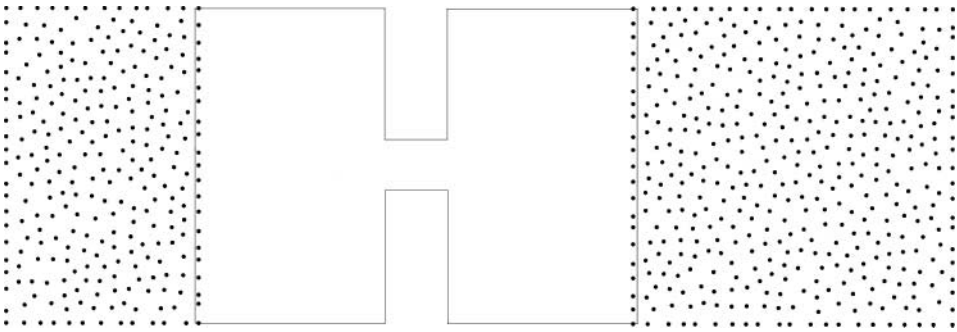

FIG. 20. Microfilter element with a random distribution of points for the Stokes subdomains.

cell-to-node and node-to-cell interpolation, respectively. Simulation results for pressure and x-velocity are plotted along the midline of the channel in Figs. 21 and 22, respectively. Also shown in the figures is the comparison to the DSMC only solution. The good match observed between the two solutions demonstrates the robustness and flexibility of the multiscale approach.

4.2.2. *Filter Array*

The development of a multiscale approach enables simulation of complex microfluidic devices and systems. To further demonstrate the multiscale approach, we consider a microfluidic filter array. The geometry of the simulated microfilter array is shown in Fig. 23. The array consists of a 0.5-μm membrane with 60 0.02-μm filter channels. The spacing

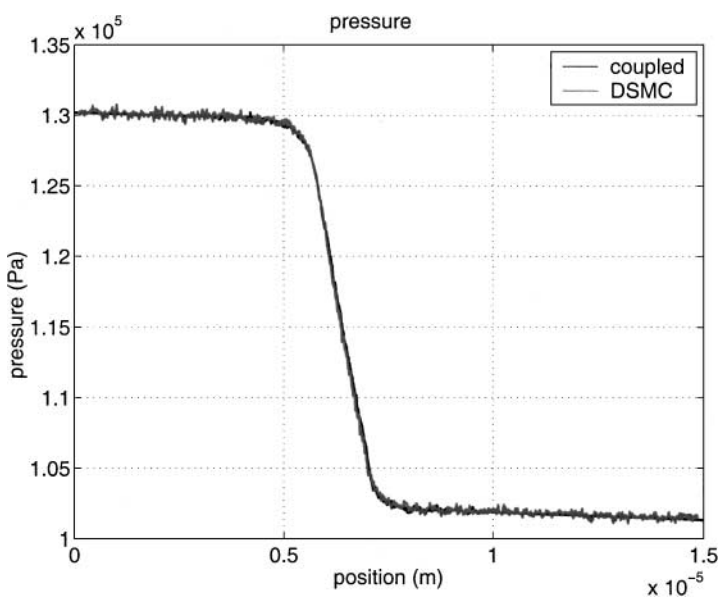

FIG. 21. Comparison of pressure, along the midline of the channel, obtained from the DSMC only simulation and the coupled simulation with a random point distribution in the Stokes subdomains ($h_c = 0.8\ \mu$m, $d_{ov} = 0\ \mu$m).

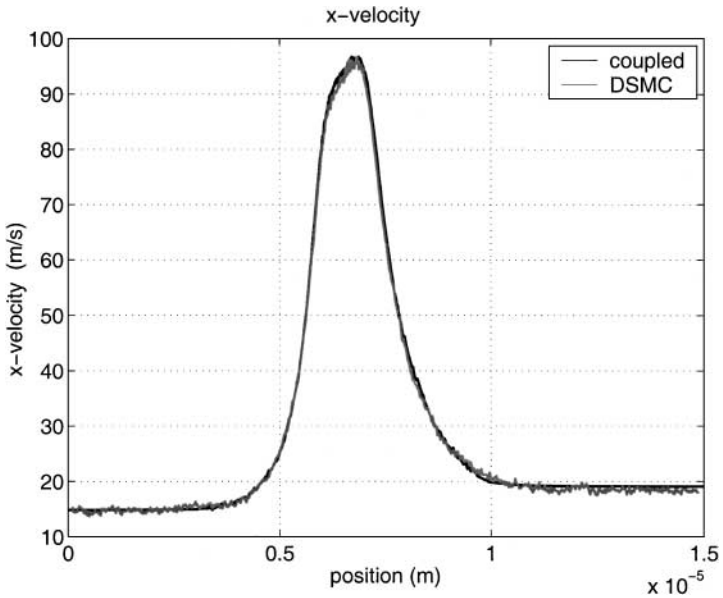

FIG. 22. Comparison of velocity, along the midline of the channel, obtained from the DSMC only simulation and the coupled simulation with a random point distribution in the Stokes subdomains ($h_c = 0.8\ \mu$m, $d_{ov} = 0\ \mu$m).

between the filters is 0.05 μm. The structure of each filter in the filter array is similar to the geometry considered in Fig. 8. Within each filter channel and in the vicinity of the filter, continuum theories breakdown and necessitate the use of DSMC. The use of DSMC to simulate the entire device can be computationally very intensive. Hence, in the regions where continuum theories breakdown, we employ DSMC and the rest of the device is simulated by using Stokes equations. Specifically, the DSMC subdomain includes the membrane and 1 μm on either side. The simulated pressure and the x-velocity for the filter array are

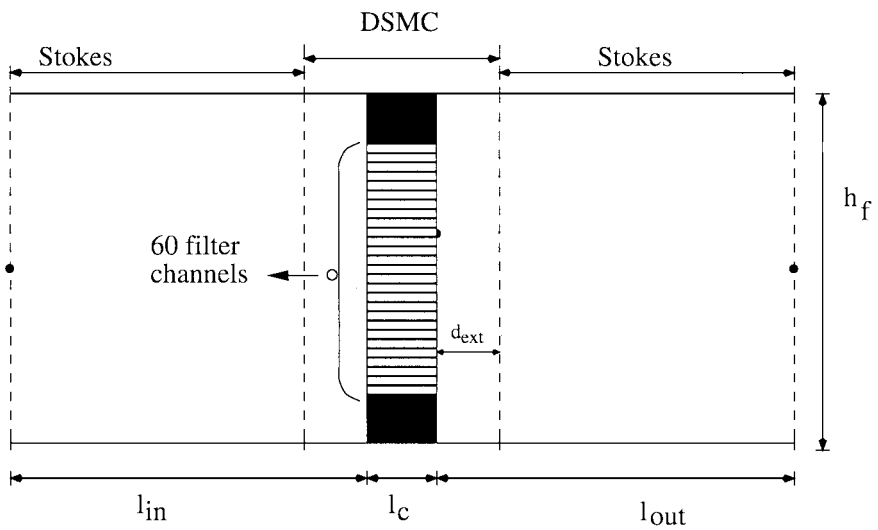

FIG. 23. The geometry of the filter array. 60 filters, each of 0.02 μm high and 1 μm long are considered.

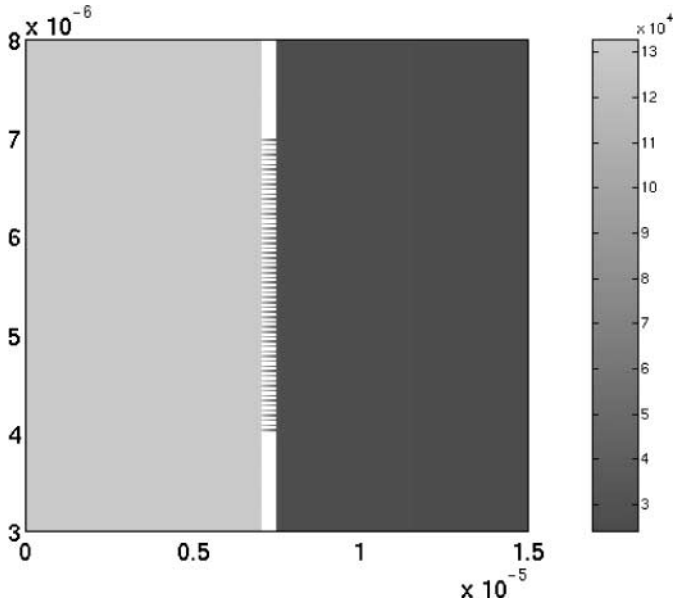

FIG. 24. Pressure (Pa) versus position (m) computed with the coupled approach for the filter array example.

shown in Figs. 24 and 25, respectively. These results again demonstrate the effectiveness of the multiscale approach. Fig. 26 plots the x-velocity at the DSMC/Stokes interface at the low-pressure side, which shows again that the coupling between the Stokes and the DSMC subdomains is two-dimensional as the x-velocity exhibits significant variation in the direction perpendicular to the flow.

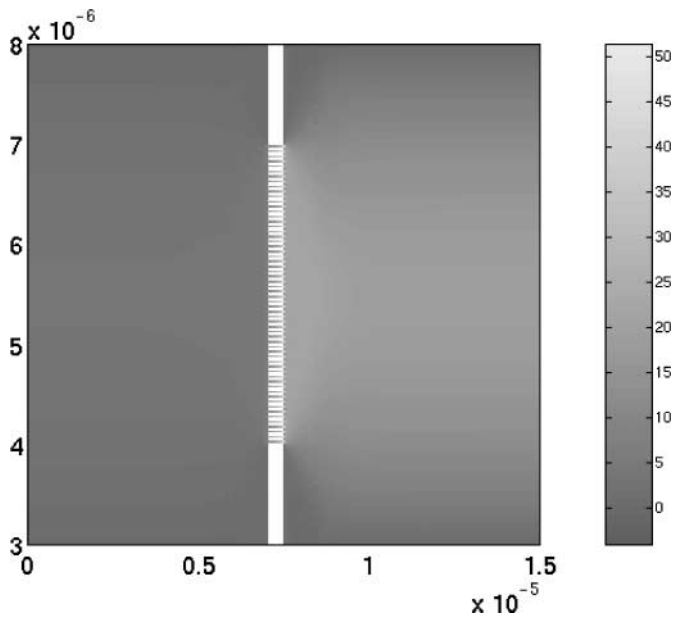

FIG. 25. x-velocity (m/s) versus position (m) computed with the coupled approach for the filter array example.

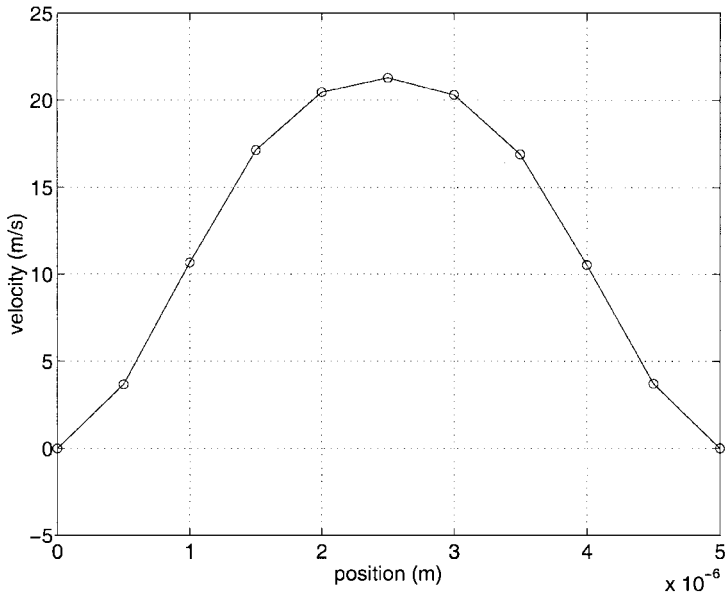

FIG. 26. x-velocity versus position at the DSMC/Stokes interface of the low pressure side, defined by x = 7.5 μm.

4.2.3. *Dual-Stage Filter*

A dual-stage filter is simulated by the coupled method to demonstrate the effectiveness of the method in handling multiple subdomains. The filter structure shown in Fig. 25 is decomposed into two DSMC subdomains and three Stokes subdomains. For accurate analysis, the second continuum domain needs to be simulated by a compressible solver. However, the second continuum subdomain is also modeled by an incompressible solver as the objective is to demonstrate the coupling of multiple subdomains. The coupling of a compressible solver with DSMC will be reported in a future publication.

Each filter stage comprises a filter array, similar to the one considered in section 4.2.2. Both filter arrays have a membrane thickness of $l_c = 0.5\ \mu$m and a total height of $h_f = 5\ \mu$m. The first stage consists of six filter channels with $h_c = 0.3\ \mu$m and $0.5\ \mu$m separation between filters. The second stage consists of 15 filter channels with $h_c = 0.1\ \mu$m and $0.2\ \mu$m

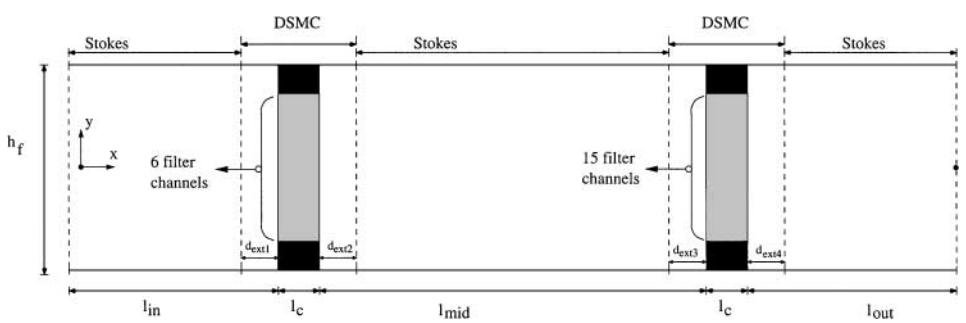

FIG. 27. The geometry of the dual-stage filter.

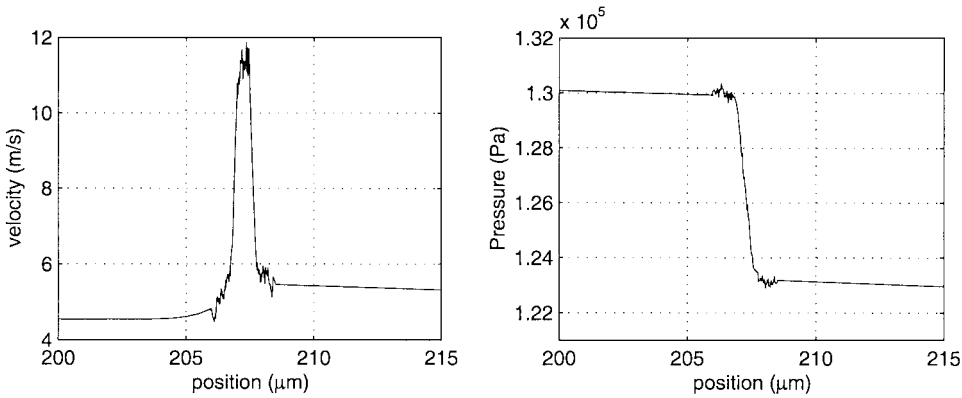

FIG. 28. The velocity and pressure at the first level filter plotted along $y = 0.25$ μm, which is the centerline for a filter in the first level.

separation between filters. The filter channels are placed symmetrically around the channel axis starting from the center. The first filter membrane was placed at $x = 207$ μm, and the second at $x = 907$ μm. The decomposition of the geometry into various subdomains is shown in Fig. 27. For the DSMC subdomains, $d_{\text{ext1}} = d_{\text{ext2}} = d_{\text{ext3}} = 1$ μm and $d_{\text{ext3}} = 3$ μm is used. The Stokes subdomains have lengths of $l_{\text{in}} = 10$ μm, $l_{\text{mid}} = 698$ μm, and $l_{\text{out}} = 10$ μm.

Since there is no filter channel at the midline ($y = 0.0$ μm), in Figs. 28 and 29 the results for pressure and velocity from the coupled procedure are shown along lines that pass through filter channels close to the midline. For the first filter stage, $y = 0.25$ μm defines a centerline of a filter. For the second filter stage, $y = 0.25$ μm does not define a conterline of a filter, so $y = 0.3$ μm is selected. The results in Fig. 28 are plotted along $y = 0.25$ μm, and the results in Fig. 29 are plotted along $y = 0.3$ μm. The ideal speed-up factor for this filter is around 150, and the actual value obtained is about 100. The reduction in efficiency is in part due to the large solution time associated with the Stokes domain in the middle. A full DSMC simulation of the dual stage filter is extremely costly due to the very large volume, the very low velocity in the connecting channels, and the very long time constant. Multiscale methods prove efficient for such problems.

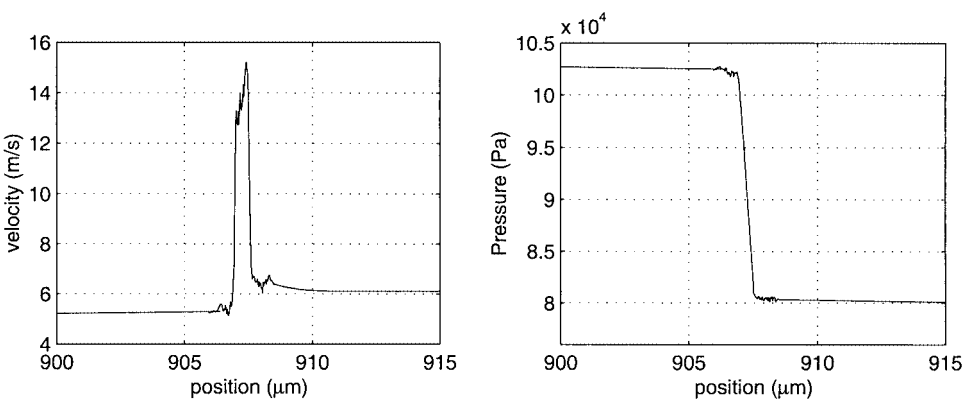

FIG. 29. The velocity and pressure at the second level filter plotted along $y = 0.3$ μm, which is the centerline for a filter in the second level.

5. CONCLUSIONS

A multiscale approach combining continuum Stokes equations with the direct simulation Monte Carlo (DSMC) methods is presented for analysis of microfluidic devices. The continuum Stokes equations are solved by a scattered point based finite cloud method. The multiscale method is based on an overlapped Schwarz technique with Dirichlet–Dirichlet type boundary conditions. The use of Dirichlet boundary conditions avoids the calculation of fluxes and results in an increased efficiency. Within the overlapped Stokes and the DSMC subdomains, the solution is interpolated from one subdomain to the other subdomain by using scattered point interpolation.

The multiscale method was applied for steady-state analysis of microfluidic filters. The convergence characteristics of the multiscale approach were investigated in detail. In particular, the dependence of convergence on the overlap size, DSMC noise, and the number of time steps considered in DSMC were investigated. The results showed that good convergence is obtained even for noisy DSMC data and short averaging times. The convergence was found to be weakly dependent on the overlap size. Finally, using the multiscale method, simulation results are also presented for a microfilter array and a dual-stage microfilter array.

APPENDIX: STABILITY OF THE COUPLED METHOD

The stability and convergence of the Schwarz method has been proven for a range of flow conditions [22]. However, for pressure driven flows the stability of the method can be an issue. Since pressure driven flows are of interest for microfluidic filters, the stability of the coupled method is investigated using a model problem of a 2D channel flow. For simplicity, the flow is assumed to be incompressible. The geometry under consideration is shown in Fig. 30. The channel is decomposed into two subdomains—the first subdomain is of length L_a and the second subdomain is of length L_b. The overlap length is L_o. The pressure, velocity and the slope of the pressure in the first and second subdomains are denoted by P_a, V_a, D_a and P_b, V_b, D_b, respectively.

For an incompressible flow, the slope of the pressure in the channel is given by $D = (P_{\text{in}} - P_{\text{out}})/L$, where P_{in}, P_{out}, and L are the input pressure, output pressure, and the length of the channel, respectively. The flow velocity at the midline is constant and is given by

$$V = \frac{1}{2\mu} D \frac{h^2}{4}, \tag{24}$$

where μ is the fluid viscosity, D is the slope of the pressure in the channel, and h is the

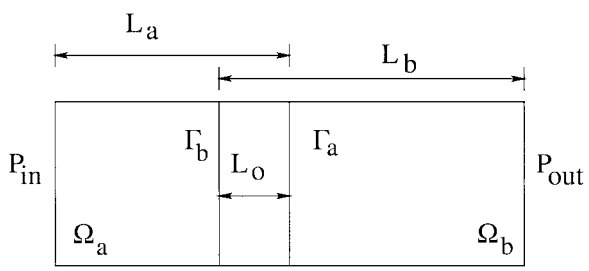

FIG. 30. A channel geometry with overlapped subdomains for the stability analysis of pressure driven flows.

channel height. For the stability analysis discussed below, we express $V = AD$, where $A = h^2/(8\mu)$.

A single iteration of an overlapped Schwarz method is considered. At iteration $n-1$, the channel is assumed to be at rest, with $P_{\text{in}} = P_{\text{out}} = p_o$, and $v = 0$, and a disturbance in the pressure is introduced at $P_b(\Gamma_b)$. Consider the case where the coupling is achieved by interpolating the pressure from Ω_b to Ω_a and interpolating the velocity from Ω_a to Ω_b:

$$P_b^n(\Gamma_b) = p_o + \delta p^n \tag{25}$$

$$P_b^n(\Gamma_a) = \left(P_b^n(\Gamma_b) - p_o\right)\frac{L_b - L_o}{L_b} + p_o \tag{26}$$

$$P_a^n(\Gamma_a) = P_b^n(\Gamma_a) = \delta p^n \frac{L_b - L_o}{L_b} + p_o \tag{27}$$

$$D_a^n = \frac{p_o - P_a^n(\Gamma_a)}{L_a} = -\delta p^n \frac{L_b - L_o}{L_b L_a} \tag{28}$$

$$V_a^n = AD_a^n = A(-\delta p^n)\frac{L_b - L_o}{L_b L_a} \tag{29}$$

$$V_b^{n+1} = V_a^n \tag{30}$$

$$D_b^{n+1} = \frac{V_b^{n+1}}{A} = (-\delta p^n)\frac{L_b - L_o}{L_b L_a} \tag{31}$$

$$P_a^{n+1}(\Gamma_a) = D_b^{n+1} L_b + p_o = (-\delta p^n)\frac{L_b - L_o}{L_b L_a} L_b + p_o \tag{32}$$

$$\delta_p^{n+1} = (-\delta p^n)\frac{L_b - L_o}{L_a}. \tag{33}$$

For the coupled implementation, where the pressure is transferred as a boundary condition from subdomain Ω_b to subdomain Ω_a and velocity is transferred as a boundary condition from Ω_a to Ω_b, if $L_b - L_o > L_a$ the initial disturbance will grow at each iteration and the coupled method will be unstable. However, the coupled implementation will be stable if the variables are passed the other way around, i.e., pressure is transferred as a boundary condition from subdomain Ω_a to subdomain Ω_b and velocity is transferred as a boundary condition from Ω_b to Ω_a. Hence, for pressure driven flows care should be exercised in enforcing interface boundary conditions for the coupled approach.

ACKNOWLEDGMENTS

This research was supported by an NSF CAREER award to N. R. Aluru. O. Aktas was also supported by a fellowship from the Computational Science and Engineering Program at UIUC. The access to parallel machines on the NCSA supercluster at UIUC is gratefully acknowledged.

REFERENCES

1. G. T. A. Kovacs, *Micromachined Transducers Sourcebook* (McGraw-Hill, New York 1998).
2. X. Yang, J. M. Yang, X. Q. Wang, E. Meng, Y. C. Tai, and C. M. Ho, Micromachined membrane particle filters, in *The Eleventh Annual International Workshop on Micro Electro Mechanical Systems (MEMS '98), Heidelberg, Germany*, Jan. 25–29 1998, pp. 137–142.

3. W.-H. Chu, R. Chin, and M. Ferrari, Silicon membrane nanofilters from sacrificial oxide removal. *J. Microelectromech. Sys.* **8**, 34 (1999).
4. W.-S. Ho, *Model Membranes: Knudsen Flow Through Microcapillaries*, Ph.D. thesis (University of Illinois at Urbana-Champaign, 1971).
5. D. B. Hash and H. A. Hassan, Assessment of schemes for coupling Monte Carlo and Navier–Stokes solution methods, *J. Thermophys. Heat Trans.* **10**, 242 (1996).
6. P. LeTallec and F. Mallinger, Coupling Boltzmann and Navier–Stokes equations by half fluxes, *J. Comput. Phys.* **136**, 51 (1997).
7. A. L. Garcia, J. B. Bell, W. Y. Crutchfield, and B. J. Alder, Adaptive mesh and algorithm refinement using direct simulation Monte Carlo, *J. Comput. Phys.* **154**, 134 (1999).
8. N. G. Hadjiconstantinou, Hybrid atomistic-continuum formulations and the moving contact-line problem, *J. Comput. Phys.* **154**, 245 (1999).
9. G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows* (Oxford Univ. Press, New York, 1994).
10. E. S. Oran, C. K. Oh, and B. Z. Cybyk, Direct simulation Monte Carlo: Recent advances and application, *Ann. Rev. Fluid Mech.* **30**, 403 (1998).
11. D. Zelesnik, M. M. Micci, and L. N. Long, Direct simulation Monte Carlo model of low Reynolds number nozzle flows, *J. Propul. Power* **10**, 546–553 (1994).
12. E. S. Piekos and K. S. Breuer, DSMC modeling of micromechanical devices, in *Proceedings of AIAA Thermophysics Conference* (AIAA 95-2089, 1995).
13. A. Beskok and G. E. Karniadakis, A model for flows in channels, pipes and ducts at micro and nano scales, *Microscale Thermophys. Eng.* **3**, 43 (1999).
14. E. B. Arkilic, M. A. Schmidt, and K. S. Breuer, Gaseous slip flow in long microchannels, *J. Microelectromech. Sys.* **6**, 167 (1997).
15. O. Aktas, U. Ravaioli, and N. R. Aluru, Application of a parallel DSMC technique to predict flow characteristics in microfludic filters, *J. Microelectromech. Sys.* **10**, 538 (2001).
16. M. Mitchell, R. Qiao, and N. R. Aluru, Meshless analysis of steady-state electro-osmotic transport, *J. Microelectomech. Sys.* 435 (2000).
17. N. R. Aluru and G. Li, Finite cloud method: A true meshless technique based on a fixed reproducing kernel approximation, *Int. J. Numer. Meth. Eng.* **50**, 2373 (2001).
18. R. P. Nance, D. B. Hash, and H. A. Hassan, Role of boundary conditions in Monte Carlo simulation of microelectromechanical systems, *J. Thermophys. Heat Trans.* **12** (1998).
19. W. W. Liou and Y. C. Fang, Implicit boundary conditions for direct simulation Monte Carlo method in MEMS flow predictions, *CMES*, **1**, 119 (2000).
20. B. Smith, B. Petter, and W. Gropp, *Domain Decomposition* (Cambridge Univ. Press, Cambridge, UK, 1996).
21. G. Chen and I. D. Boyd, Statistical error analysis for the direct simulation Monte Carlo method, *J. Comput. Phys.* **126**, 443 (1996).
22. M. Fortin and R. Aboulaich, Schwarz's decomposition method for incompressible flow problems, in *Proc. First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, January 1987*, edited by R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périauxm (Soc. for Industr. & Appl. Math., Philadelphia, 1987), pp. 333–349.