Algorithms in FastStokes and Its Application to Micromachined Device Simulation

Xin Wang, *Member, IEEE,* Joe Kanapka, Wenjing Ye, Narayan R. Aluru, *Member, IEEE,* and Jacob White, *Associate Member, IEEE*

***Abstract*—For a wide variety of micromachined devices, designers need accurate analysis of fluid drag forces for complicated three-dimensional (3-D) problems. This paper describes FastStokes, a recently developed 3-D fluid analysis program. FastStokes rapidly computes drag forces on complicated structures by solving an integral formulation of the Stokes equation using a precorrected fast Fourier transform (PFFT)-accelerated boundary element method (BEM). The specializations of the PFFT algorithm to the Stokes flow problem are described, and computational results are presented. Timing results are used to demonstrate that FastStokes scales almost linearly with problem complexity, can easily analyze structures as complicated as an entire comb drive in under an hour, and can produce results that accurately match measured data.**

***Index Terms*—Boundary element method (BEM), FastStokes, fluid, microelectromechanical systems (MEMS), simulation, Stokes flow.**

I. Introduction

NEARLY all of the micromachined devices being developed for biological applications manipulate gas or liquid, and for many of these devices, performance optimization depends critically on understanding fluid forces in very complicated three-dimensional (3-D) geometries [1]. Although general finite-volume and finite-element fluid flow analysis programs can perform these analyses, these programs are too time consuming to be used for design optimization, particularly for very complicated geometries. In the case of micromachined devices, faster approaches can be developed by noting that the fluid flow is primarily Stokes flow and the quantities of interest are typically drag forces on bodies or structures in fluid.

In this paper, we describe the algorithms used in FastStokes, a very fast fluid analysis program useful for extracting surface forces on very complicated micromachined devices. The FastStokes program is based on solving an integral formulation of Stokes equation using a specialized accelerated boundary element method (BEM). We describe the approach by first reviewing background material on the integral formulation of the Stokes equation, the standard BEM discretization, and the precorrected fast Fourier transform (PFFT)-accelerated iterative algorithm for solving the BEM equations. In Sections III and IV, we present the main contributions of the paper, starting with a description of the specializations of the PFFT algorithm to Stokes flow problem in Section III. In Section IV, we discuss the singularity in the BEM operators and present a modified Krylov subspace algorithm for addressing the singularity. In Section V, we provide several numerical examples and compare computational results with experimental results to demonstrate the accuracy and efficiency of FastStokes.

II. Background

The FastStokes program numerically solves the incompressible Stokes equation. In this section, we describe the incompressible Stokes equation, which can be derived by assuming a small Reynolds number in the incompressible Navier–Stokes equation, and give an integral formulation. We also describe the basic BEM method for discretizing integral equations and show how it is as applied to the integral form of Stokes equation. Finally, we use a simple electrostatics example to give a brief presentation of the PFFT-accelerated iterative method for solving BEM equations.

A. Stokes Integral Equations

Fluid flow in which viscous forces dominate over inertial forces is referred to as a Stokes flow or a creeping flow. The flow in many micromachined devices, such as air-packaged actuators or liquid-handling mixers, pumps, and valves, is Stokes flow, as can be seen by examining the associated Reynolds number. The Reynolds number is defined as $\mathrm{Re} = UL/\nu$, where U is the velocity, L is the characteristic length, and ν is the kinematic viscosity of the fluid. Since $\mathrm{Re} \propto$ inertia force/viscous force, Reynolds number is frequently used in experiments to determine whether viscous or inertial forces dominate. Because the feature size of many microelectromechanical system (MEMS) devices is on the order of a micrometer, the UL product is small even if the movable parts oscillate at a reasonably high frequency. For example, consider an air-packaged resonator oscillating at 10 kHz with an oscillation

Manuscript received March 5, 2005; revised June 15, 2005. This paper was recommended by Associate Editor K. Chakrabarty. This work was supported by grants from the Defense Advanced Research Projects Agency (DARPA) Composite CAD program, the National Science Foundation (NSF), the Singapore-MIT Alliance, the Semiconductor Research Program, and Analog Devices Inc.

X. Wang is with Synopsys Inc., Mountain View, CA 94043 USA (e-mail: wangxin@synopsys.com).

J. Kanapka is with MathWorks, Inc., Natick, MA 01760 USA.

W. Ye is with the Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30339 USA.

N. R. Aluru is with the Beckman Institute for Advanced Science and Technology, the Department of Mechanical and Industrial Engineering, the Department of Electrical and Computer Engineering, and the Bioengineering Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA.

J. White is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: white@mit.edu).

Digital Object Identifier 10.1109/TCAD.2005.855938

amplitude of 1 μm. Using an air kinematic viscosity at 300 K of $\nu = 1.566 * 10^{-5}$ m^2/s [2], the Reynolds number is $\mathrm{Re} = (2\pi \times 10^4 \times 10^{-6} \times 10^{-6}/1.566 \times 10^{-5}) \approx 0.004$, and therefore the inertial force can be neglected. For structures in liquids, the kinematic viscosity is much lower because of the higher density of liquid, but the operating frequencies (or structure velocities) are commensurately lower than those in air, and so Reynolds number remains low. Applying the small Reynolds number assumption to the Navier–Stokes equations yields the steady Stokes equations

$$\begin{cases} -\nabla P + \mu \nabla^2 \vec{u} = 0 \\ \nabla \bullet \vec{u} = 0 \end{cases} \tag{1}$$

where $\vec{u}$ is the vector velocity, P is the pressure, and μ is the dynamic viscosity of the fluid. The surface velocities and forces satisfy an integral relation

$$\begin{aligned} u_i(\vec{x}) = & \sum_{j=1}^{3} \int G_{ij}(\vec{x}, \vec{y}) f_j(\vec{y}) ds(\vec{y}) \\ & + \sum_{j=1}^{3} \sum_{k=1}^{3} \int T_{ijk}(\vec{x}, \vec{y}) u_j(\vec{y}) n_k(\vec{y}) ds(\vec{y}), \\ & \qquad i = 1, 2, 3 \end{aligned} \tag{2}$$

where the domain of integration for the surface integrals is the union of the surfaces of the structures embedded in the fluid, $\vec{x}$ is any field point, $\vec{y}$ is a point on a structural surface, $u_i(\vec{x})$ and $f_i(\vec{x})$, i $= 1, 2, 3$, are the x-, y-, or z-directed surface velocities and surface forces, respectively, and $\vec{n}$ is the surface outward normal. Green's functions are given by [3]

$$\begin{aligned} G_{ij} &= -\frac{1}{8\pi\mu}\left(\frac{\delta_j^i}{r} + \frac{\hat{x}_i \hat{x}_j}{r^3}\right) \\ T_{ijk} &= -\frac{3}{4\pi}\frac{\hat{x}_i \hat{x}_j \hat{x}_k}{r^5} \\ & \qquad r = |\vec{x} - \vec{y}|, \ \hat{x}_i = x_i - y_i. \end{aligned}$$

For micromachined devices, the fluid-embedded structures are either stationary and rigid, or are deforming more slowly than the fluid response time and can be treated as quasi-statically rigid. For rigid bodies, the surface integral with kernel T_{ijk} in (2) is zero [3], greatly simplifying the integral equation. The FastStokes program makes use of this rigid body assumption and solves the simplified integral equation

$$u_i(\vec{x}) = \sum_{j=1}^{3} \int G_{ij}(\vec{x}, \vec{y}) f_j(\vec{y}) ds(\vec{y}), \qquad i = 1, 2, 3. \tag{3}$$

Integral equation (3) can be used to determine traction and pressure forces on the surface of a fluid-embedded structure given velocity boundary conditions. The form of (3) is also referred to as the single-layer integral equation representation of the Stokes flow problem [3].

B. Discretization

In order to compute traction and pressure forces using (3), FastStokes uses a BEM scheme in which the integral equation is first discretized by subdividing the surface into flat panels. In particular, the FastStokes program reads an input file that contains data describing a surface mesh discretization comprised of flat triangle or quadrilateral panels. The primary reason for using flat panel discretizations is that they are easy to generate, but flat panels are also particularly well suited to micromachined structures as they usually have nearly flat surfaces.

After surface discretization, a piece-wise constant collocation method is applied to solve the integral equation in (3). This collocation method is based on assuming panel force densities are constant on each panel and that when these panel force densities are used as the traction forces in (3) they produce velocities at panel centroids that exactly match the given velocity boundary conditions. This discretized form of the velocity integral equation is then

$$\begin{aligned} u_i(\vec{x}_l) = & \sum_{k=1}^{\text{number of panels}} \sum_{j=1}^{3} f_j(\vec{y}_{\text{centroid}}) \\ & \times \int\limits_{\text{panel } k} G_{ij}(\vec{x}_l, \vec{y}) ds_k(\vec{y}), \qquad i = 1, 2, 3 \end{aligned} \tag{4}$$

where $\vec{x}_l$ is the centroid of the lth panel. The panel integrals in (4) can be evaluated analytically, at least for the case of polygonal flat panels, using an extension of an approach in [10]. The technique is described in detail in the Appendix.

Equation (4) can be written in matrix–vector form as

$$\begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \mathbf{G}_{13} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \mathbf{G}_{23} \\ \mathbf{G}_{31} & \mathbf{G}_{32} & \mathbf{G}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \end{bmatrix} \tag{5}$$

or

$$\mathbf{U} = \mathbf{GF}$$

where $\mathbf{U}_1$, $\mathbf{U}_2$, and $\mathbf{U}_3$ are the x, y, and z components of the known panel centroid velocity vectors; $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$ are the x, y, and z components of the unknown piece-wise constant panel surface force density vectors; and $\mathbf{G}$ is the matrix form of the G_{ij} integral operator. Given the surface velocities, which automatically satisfy the continuity equation if all the surfaces are boundaries of rigid bodies, (5) can be used to compute surface forces. Finding methods for efficiently solving (5) is the key to developing a fast solver since the $\mathbf{G}$ matrix is not only dense but also singular. We discuss solving the dense matrix problem in the next subsection; the singularity problem is discussed in Section IV.

C. PFFT Algorithm

It is well known that traditional BEM methods are too slow for large problems because they generate dense linear systems that are expensive to form and solve. Very efficient techniques for handling these linear systems were developed during the past two decades by combining sparsification techniques with rapidly converging preconditioned iterative schemes, as first proposed in [4] and first used for accelerating BEM in general 3-D problems in [5]. As will be described below, such methods avoid explicitly forming (5) and can be used to reduce the cost of solving (5) from $O(n^3)$ to nearly $O(n)$ operations.

The basic idea behind these accelerated BEM methods follows from first considering applying an iterative method, like the Krylov subspace based GMRES algorithm [8], to solve a dense system like the one in (5). When applied to solving a generic linear system $\boldsymbol{Ax} = \boldsymbol{b}$, the qth iteration of a Krylov subspace method constructs an approximate solution by selecting the best weighted combination of the vectors in a qth-order Krylov subspace

$$\{b, Ab, A^2b, \ldots, A^q b\}.$$

The $q + 1$th-order Krylov subspace can be computed from the qth-order subspace by multiplying a vector by the matrix $\boldsymbol{A}$, and since that matrix is dense in the BEM case, computing matrix–vector products is $O(n^2)$ operations and dominates the cost of iteratively solving BEM equations.

For the case of many BEM matrices, the matrix–vector products can be computed approximately in $O(n)$ or $O(n \log(n))$ time by exploiting the fact that the matrices associated with BEM have certain properties. For example, nearby panels can be clustered together when evaluating their contributions to the potential at distant collocation points. This multiresolution idea is exploited in methods based on the fast multiple algorithm [6]. Alternatively, the near convolutional structure of the underlying integral equation can be exploited using the PFFT algorithm [7]. The FastStokes program uses a modification of the PFFT algorithm, which is described in more detail below.

The basic PFFT algorithm is easily illustrated using the single-variable electrostatic problem as an example. In the simpler electrostatic problem, the integral equation and its discretized form are

$$V(\vec{x}) = \frac{1}{4\pi\varepsilon} \int \frac{1}{\|\vec{x} - \vec{y}\|} q'(\vec{y}) ds(\vec{y})$$

$$V_i = \sum_{j=1}^{\text{number of panels}} \Pi_{ij} q_j \qquad \text{or} \qquad \mathbf{V} = \mathbf{\Pi q}$$

with

$$\Pi_{ij} = \frac{1}{4\pi\varepsilon} \frac{1}{\text{area(panel } j)} \int_{\text{panel } j} \frac{1}{\|\vec{x} - \vec{y}\|} ds(\vec{y}) \qquad (6)$$

where V_i is the voltage at the centroid of the ith panel and $\mathbf{V}$ is the voltage vector. The charge density q' is assumed to be constant over each panel, and q_j is used to denote the net charge on the jth panel. The matrix element Π_{ij} is the potential at collocation point i due to unit net charge on panel j.

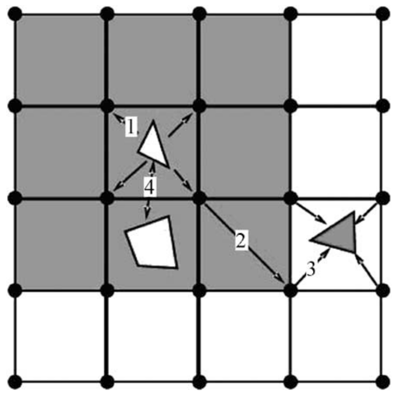

Fig. 1. Four major steps of the PFFT algorithm.

Given a voltage vector $\mathbf{V}$, consider computing the charge vector $\mathbf{q}$ by solving $\mathbf{V} = \mathbf{\Pi q}$ using the GMRES algorithm mentioned above. Since GMRES is a Krylov subspace method, it will be necessary to compute many matrix–vector products with the dense matrix $\mathbf{\Pi}$. The PFFT algorithm can be used to reduce the cost of computing matrix–vector products by separating the panel interactions into nearby and far field interactions. Then, the cost-dominant and smoother far field interactions are computed very rapidly by projecting and interpolating from an underlying uniform grid, and then resolving the grid interactions using multidimensional FFTs. Note that we say the far field interaction is smoother because the kernel $1/\|\vec{x} - \vec{y}\|$ varies much more slowly in space when the source point $\vec{y}$ is far from the field point $\vec{x}$. Nearby interactions have very rapid spatial variations so they are computed directly using an accurate kernel integration algorithm to avoid large numerical errors.

The four major steps of the PFFT algorithm are listed below and are also pictorially illustrated in Fig. 1:

1) project the panel charges onto the FFT grid $\mathbf{q}^{\mathbf{grid}} = \mathbf{W}^{\mathbf{projection}} \mathbf{q}^{\mathbf{panel}}$;
2) compute grid voltages due to grid charges using the FFTs. This step can be expressed as $\mathbf{V}^{\mathbf{grid}} = \mathbf{ifft}(\mathbf{fft}(\mathbf{\Pi}^{\mathbf{grid}})\mathbf{fft}(\mathbf{q}^{\mathbf{grid}}))$;
3) interpolate the grid voltages back to panel voltages, i.e., $\mathbf{V}^{\mathbf{panel}} = \mathbf{W}^{\mathbf{interpolation}} \mathbf{V}^{\mathbf{grid}}$;
4) directly compute nearby interactions and use the results to replace the inaccurate nearby parts of the voltages calculated from the grid.

The cost of the PFFT algorithm is dominated by the cost of the FFT step, which costs $O(n \log(n))$ operations. The panel charges are projected onto neighboring grid points using the sparse projection matrix $\mathbf{W}^{\mathbf{projection}}$, the elements of which are calculated by matching the panel moments with the nearby grid moments. The interpolation step assumes the potential distribution is smooth, so that panel centroid potentials can be computed accurately by polynomially interpolating grid potentials. Since the number of the neighboring grid points associated with a panel is bounded by a constant, the costs of the local projection and interpolation operations are only $O(n)$. Therefore, the total computational cost of the PFFT-accelerated BEM is $O(n \log(n))$.

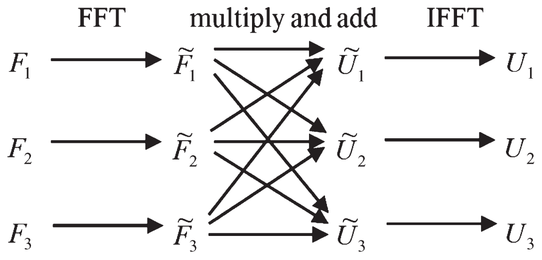

Fig. 2. FFT operations in FastStokes.

III. PFFT FOR THE STOKES PROBLEM

The PFFT algorithm described in the previous section can be used for both one-variable (scalar) and multivariable (vector) problems, but the most obvious vector extension is not the most efficient. In this section, we discuss how to efficiently adapt the PFFT algorithm to the vector Stokes flow problem.

When applying an iterative method to solve (5), $\mathbf{U} = \mathbf{GF}$, the most expensive computation is forming matrix–vector products using the dense matrix $\mathbf{G}$. Forming the needed matrix–vector products can be equivalently considered as computing panel centroid velocities due to a candidate set of panel forces. In order to use the PFFT algorithm to compute the vector of centroid velocities, it is most straightforward to consider using the algorithm to separately compute the nine terms associated with the contribution of three force components to three velocity components.

To see why the straightforward approach is inefficient, consider the second step of the PFFT algorithm described in Section II-C. This second step is a convolution computed using an FFT and an inverse FFT (IFFT). The straightforward approach to performing this step for the vector case can be described as

$$U_j^{\text{grid}} = \sum_{k=1}^{3} \text{IFFT}\left(\widetilde{G}_{jk}^{\text{grid}} \times \text{FFT}\left(F_k^{\text{grid}}\right)\right). \tag{7}$$

Note that the formula in (7) requires a total of 18 FFTs and IFFTs. A more efficient approach that avoids repeating the calculations of $\text{FFT}(F_j^{\text{grid}})$ is to save the result of $\widetilde{F}_j^{\text{gird}} = \text{FFT}(F_j^{\text{grid}})$. In addition, only one IFFT is needed for every grid velocity calculation if the following scheme is used, i.e.,

$$\begin{aligned}\widetilde{F}_j^{\text{grid}} &= \text{FFT}(F_j^{\text{grid}})\\ U_j^{\text{grid}} &= \text{IFFT}\left(\sum_{k=1}^{3}\left(\widetilde{G}_{jk}^{\text{grid}} \times \widetilde{F}_j^{\text{grid}}\right)\right).\end{aligned} \tag{8}$$

Note that using the approach in (8), only six FFT and IFFT operations are needed for the matrix–vector product calculation, rather than the 18 FFTs and IFFTs needed using (7). This idea can be shown schematically in Fig. 2.

In addition to the modification of the transform part of the PFFT algorithm, there are other optimizations helpful for Stokes problems. The Stokes integral equation has three velocity components and three force components, but only six independent kernels, as $G_{ij} = G_{ji}$. This is a helpful observation because the FFTs of the grid kernels $\widetilde{G}_{jk}^{\text{grid}} = \text{FFT}(G_{jk}^{\text{grid}})$ are calculated once and then stored so that they can be used repeatedly for each matrix–vector product. The projection and interpolation matrices are also stored, and if polynomial projection is used, then these matrices are coordinate independent and only one set is needed [9].

IV. NULL SPACE OF THE SINGULAR INTEGRAL OPERATORS AND THE MODIFIED GMRES

The fact that only the derivative of pressure arises explicitly in the Stokes equation implies that any constant pressure can be added to the solution of the Stokes equation, and therefore the equation does not have a unique solution. This constant-pressure zero-velocity solution is a "singular mode" or a null space vector that does not affect the total forces on a single rigid body, but the singularity can impact the results produced by a numerical procedure. One approach to eliminating the null space is to add an addition operator to the integral equation that maps the Stokes flow operator's null space to its defect in the range [11]. Below, we describe an alternative, one that removes the null space using a modification of the GMRES iterative matrix solution algorithm. Our approach is not as general as the technique in [11], but it fits with the fast solver methodology and guarantees a null space free solution independent of discretization or sparsification errors. Note that the null space free solution is only useful for computing total body forces. To correctly compute the detailed force distribution, the null space contributions must be determined by solving an additional pressure matching equation [12].

Constant pressure force on the surface of a rigid body generates zero net body force or torque, and therefore zero velocity. Consider such a force, denoted as f_j, acting on a rigid body. This force acts only on the surface normal direction of the rigid body and has a constant magnitude, so it is a multiple of the surface normal vector for the rigid body, denoted n_j. And since f_j generates zero velocity, it must follow that

$$\int_{\text{surface}} G_{ij} n_j ds = 0 \tag{9}$$

and therefore n_j is in the null space of the integral equation. In general, a problem with m independent bodies will have m independent null space vectors that correspond to being equal to the surface normal on one body and zero on the others. The discretization of an m-body system generates a system equation $\mathbf{U} = \mathbf{GF}$, where $\mathbf{G}$ is now the discrete form of the integral operator with an m-dimension null space given by the outward-normal vectors of the m objects in the system.

If a Krylov subspace based method is applied to solve $\mathbf{U} = \mathbf{GF}$, then removing the null space of the $\mathbf{G}$ matrix can be performed by removing the null space from every Krylov subspace vector since the final solution is in the Krylov subspace, i.e.,

$$\text{Krylov Subspace} = [U, GU, G^2U, G^3U, \ldots]. \tag{10}$$

A simple approach to removing the null space is to remove the orthogonal projection on to the null space from every

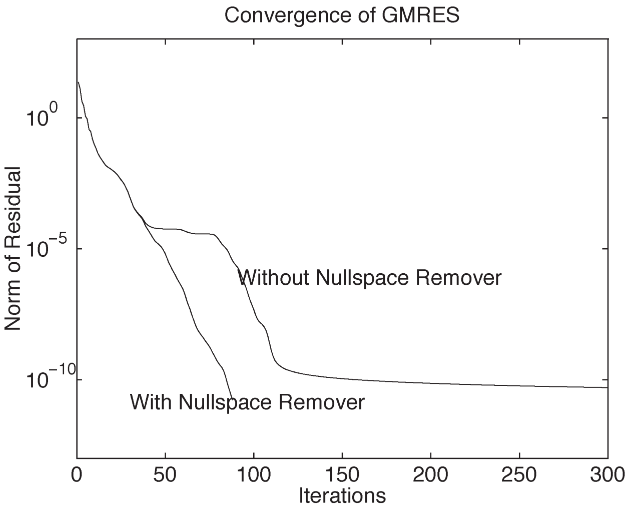

Fig. 3. Convergence of the modified GMRES algorithm.

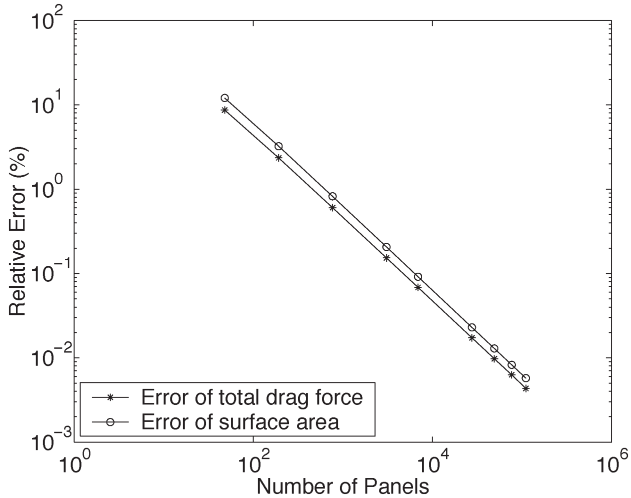

Fig. 4. Percentage relative error of the sphere versus the number of panels.

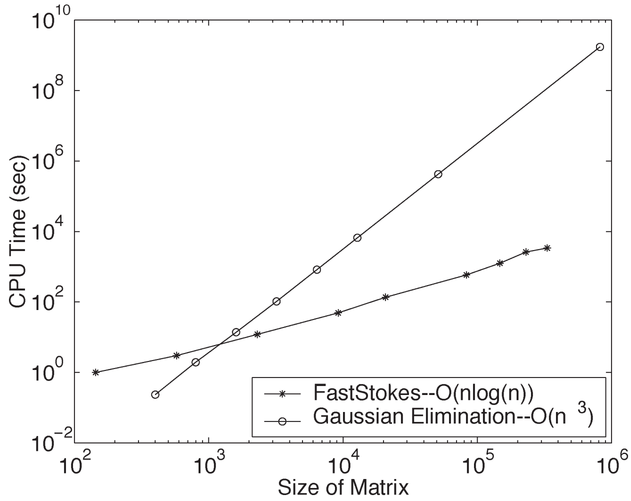

Fig. 5. CPU times of FastStokes and Gauss elimination.

matrix–vector product computed in the Krylov subspace algorithm. Such an approach guarantees that the null space vectors will not contaminate any orthogonalization being performed on the Krylov subspace. This is important because contamination of the Krylov subspace by the null space can interfere with convergence. Thus, this modification not only generates a null-space-free solution but also makes the Krylov subspace algorithm converge faster. To demonstrate this phenomenon, the GMRES algorithm was applied to solve a system of the form in (5) generated from a complicated fluid analysis. The convergence of GMRES with and without the null space remover is shown in Fig. 3, and demonstrates that without the null space remover the GMRES algorithm stalls.

It is worth noting that when a velocity vector associated with rigid body motion forms the right-hand side of (5), that velocity vector must satisfy a divergence-free condition. This implies that the velocity vector is orthogonal to the integral equation null space. Orthogonality should guarantee that the associated Krylov subspace is also null space free and the null-space remover should be unnecessary. However, since the PFFT algorithm is used to compute approximates to matrix–vector products with $\mathbf{G}$, the null space can easily appear and contaminate the subspace. Therefore, the null space remover substantially enhances robustness.

V. Simulation Examples

We present three simulation examples in this section to show the effectiveness of the steady incompressible FastStokes solver. The first simple sphere example demonstrates that the fast solver does not interfere with the convergence of the discretization method. The second and third examples, a comb drive and a micromirror, are used to demonstrate that the FastStokes program generates drag results that correlate surprisingly well with measured data.

A. Translating Sphere

For the simple spherical geometry, an analytical solution of the Stokes equation exists. Given the radius of the sphere R_0 and a constant velocity $\vec{U}$, the drag force on the sphere is

$$\vec{F} = 6\pi\mu R_0 \vec{u}. \tag{11}$$

For this computational experiment, it is assumed that $\mu = 1$, $R_0 = 1$, $U_x = 1$, $U_y = U_z = 0$, and FastStokes is used to calculate the x-direction drag forces numerically. The red line in Fig. 4 shows the percentage relative error and clearly indicates that the error decreases from approximately 2% for 100 panels to 0.004% for 100 000 panels and that decrease is a straight line when viewed on a log–log plot. The blue line in Fig. 4 shows the error of total surface area due to the flat panel discretization. Note that the blue line is parallel and very close to the red line. This is because the error of the flow calculation is mainly due to the geometrical error of using a flat panel discretization, and this geometrical error is reflected by the error of the total surface area. The CPU times of using the $O(n\log(n))$ FastStokes solver and the traditional $O(n^3)$ Gaussian elimination method (LU decomposition) are compared in Fig. 5. If 5000 panels are used, FastStokes is about 3000 times faster than Gaussian elimination. The memory used

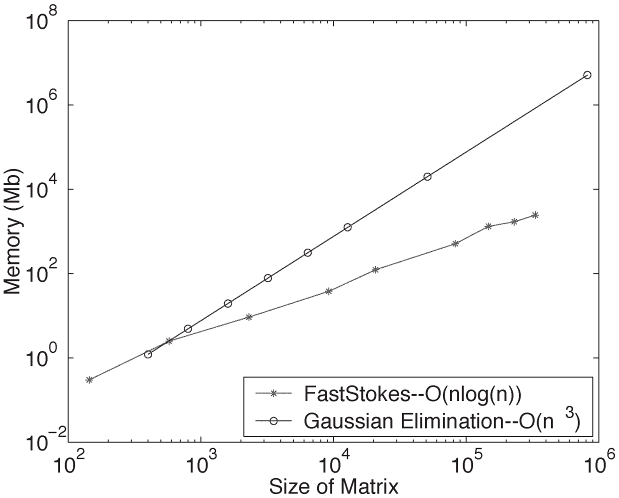

Fig. 6. Memory usage of FastStokes and Gauss elimination.

by Gaussian elimination is $O(n^2)$ while that of FastStokes is much less (about $O(n) \sim O(n^{1.5})$); the comparisons are shown in Fig. 6. A 500-MHz dual-processor computer running an Alpha-Linux system is used for the simulations.

B. Comb-Drive Resonator

A lateral comb-drive resonator is shown in Fig. 7. The test structure was fabricated using the MUMPS process at MCNC (now Cronos Integrated Microsystems Inc., Research Triangle Park, NC). The dimensions of the resonator are given in Table I. The movable comb-drive was set into motion in air at atmospheric pressure using an electrical stimulus to one static comb-drive. The magnitude and angle of the resulting motions were measured using the computer microvision technology [13]. The measured resonant frequency of the later motion is 19.2 kHz and the quality factor is 27.

A discretization using 16 544 panels is shown in Fig. 8. The lateral direction surface force solution is shown in Fig. 9. Using the rigid-body assumption and a second-order spring-mass-damper system as a macromodel, we calculate the damping coefficient b from the FastStokes result and then further calculate the quality factor Q, i.e.,

$$\begin{aligned} m_{\text{eff}}\ddot{x} + b\dot{x} + kx &= F_{\text{electrostatic}} \\ Q &= \frac{\sqrt{km_{\text{eff}}}}{b} \\ m_{\text{eff}} &= m_m + \frac{12}{35}m_b + \frac{1}{4}m_t \\ &= 5.61 \times 10^{-11}\ \text{kg} \end{aligned} \tag{12}$$

where m_m and m_t are the masses of the movable comb-drive and the tethers, respectively [13], and m_t is the mass of the connecting truss. Stiffness k can be calculated from the resonance frequency and the effective mass using $k = (2\pi f_0)^2 m_{\text{eff}} = 0.816$ N/m. The simulation result is compared with the experimental result in Table II. The steady incompressible FastStokes solver gave a numerical solution that is very close to the experimental results, while simple approaches such as using the Couette flow model failed. The convergence is shown in Fig. 10; the solution is accurate even if a coarse mesh with 4868 quadrilateral panels is used. The CPU time is shown in Fig. 11, a very fine discretization with 59 280 panels takes a little more than an hour's time.

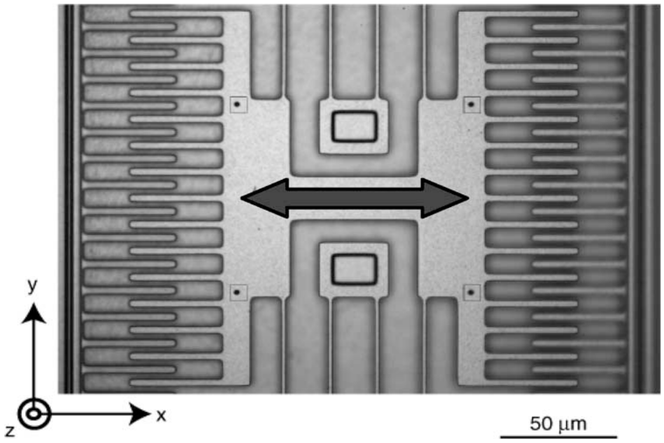

Fig. 7. SEM of a lateral resonator.

TABLE I
RESONATOR DIMENSIONS

	Dimensions (μm)
Finger gap	2.88
Finger length	40.05
Finger overlap	19.44
Tether length	151
Tether width	1.1
Thickness	1.96
Substrate gap	2

C. Micromirror

An electrostatically actuated micromirror is simulated using FastStokes [12]. The micromirror is fabricated and tested at the Micromachined Product Division of Analog Devices Inc. (Cambridge, MA). The air-packaged micromirror is the critical part of an optical switch, and its dynamic performance is strongly affected by the viscous drag forces. Testing data have shown that the mirror is heavily damped with a quality factor of around two in certain designs. Two major modes, the "mirror only" rotation mode and "mirror + gimbal" rotation mode, are simulated here. Table III compares the simulation and experimental results of two different designs.

The simulated and measured quality factors match within 10%. Again, the small differences prove the accuracy of the FastStokes program. Fig. 12 shows the z-direction force on a mirror when both mirror and gimbal rotate. Only half of the mirror is plotted in Fig. 12 in order to show a clear view of the force distribution. Fig. 13 shows that the simulation solution

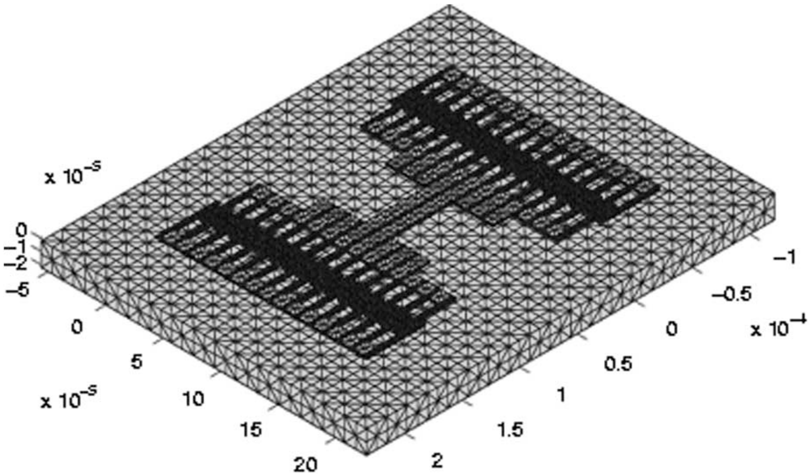

Fig. 8. Surface discretization of the lateral resonator.

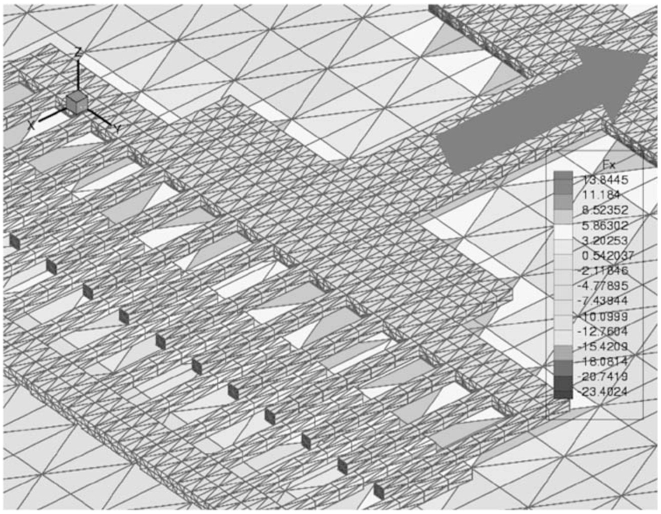

Fig. 9. Detailed drag force on a lateral resonator using the incompressible Stokes model.

TABLE II
COMB-DRIVE RESONATOR SIMULATION AND MEASUREMENT RESULTS

	Q
Couette Flow	58.9
FastStokes	29.8
Experiment	27

quickly converges as the discretization is refined. Fig. 14 shows the CPU time. The simulation was finished in less than an hour when 42 340 panels were used.

VI. Summary

In this paper, we summarized the algorithms in FastStokes, and in particular described several specializations of the PFFT-accelerated BEM algorithm to the Stokes flow problem. In addition, we gave timing results on several examples to demonstrate that FastStokes scales almost linearly with problem complexity, can easily analyze structures as complicated as an entire comb drive in under an hour, and can produce results that accurately match measured data.

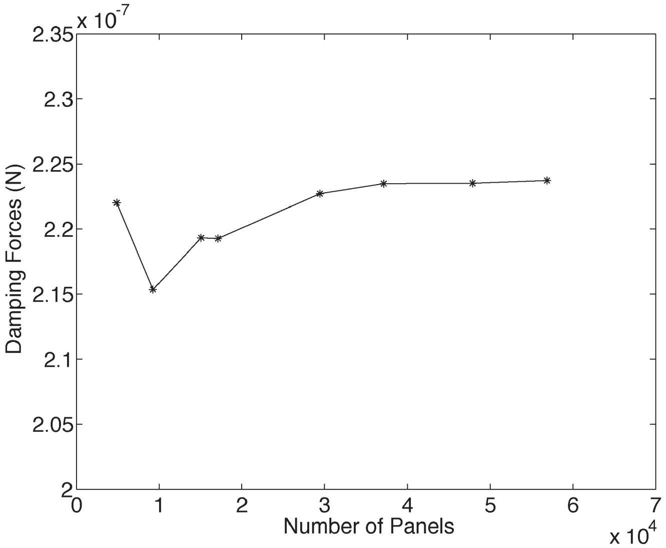

Fig. 10. Convergence of the drag forces of comb-drive resonator simulation.

The techniques in FastStokes have been extended to include slip boundary conditions as these conditions are used to model noncontinuum microfluidic effects [14]–[16]. As devices are scaled, slip effects will become important, but geometries in current common designs are still so large that the impact of slip effects on net drag is limited. Future work is on developing more efficient methods for handling substrate ground planes and extending these fast fluid solver techniques to unsteady problems, convection–diffusion problems, cells-in-flow problems, and noncontinuum problems.

Appendix
Analytical Flat Panel Integration Algorithm

Accurate calculation of the elements of $\mathbf{G}$ in (5) associated with nearby interactions is crucial to ensuring the accuracy of the Stokes flow calculation. Although these nearby terms are

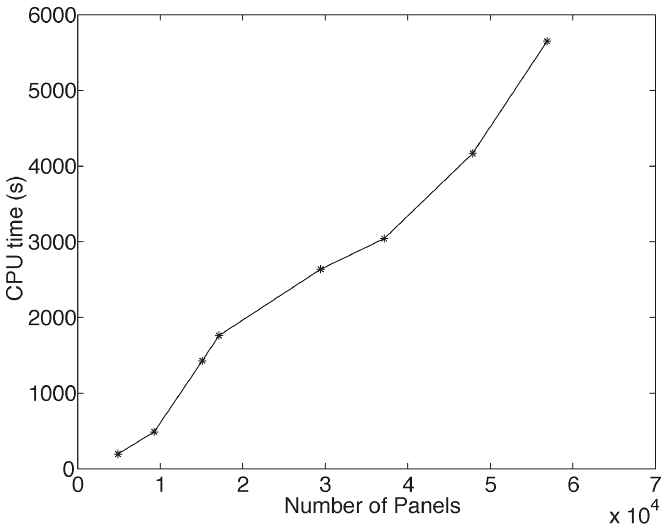

Fig. 11. CPU times of comb-drive resonator simulation.

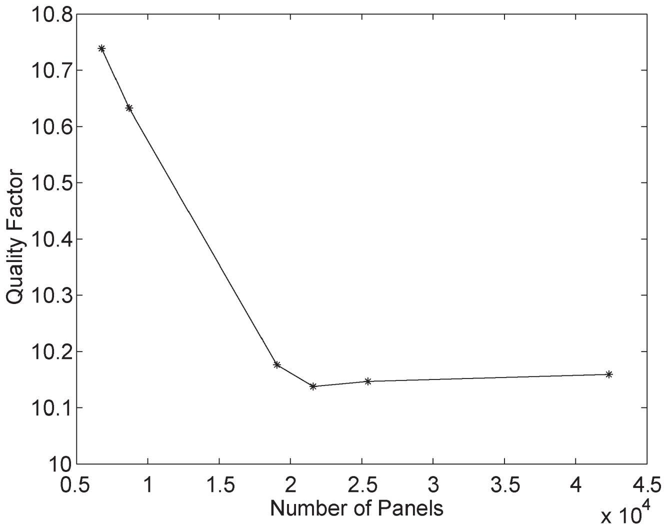

Fig. 13. Convergence of the micromirror simulation.

TABLE III
QUALITY FACTORS OF THE MICROMIRROR SIMULATIONS AND MEASUREMENTS

		Measured Q	Simulated Q	Error (%)
Mirror 1	Mirror+gimbal	2.31	2.36	2.16
	Mirror	3.45	3.14	8.99
Mirror 2	Mirror+gimbal	4.27	4.69	9.84
	Mirror	10.63	10.16	4.42

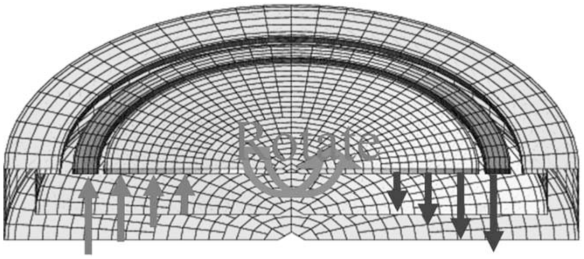

Fig. 12. z-direction force on a micromirror.

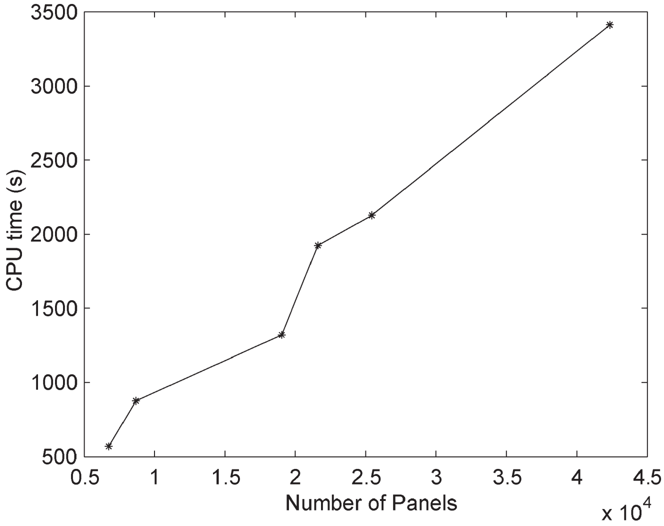

Fig. 14. CPU times of micromirror simulation.

few in number, they are large and very sensitive to spatial variation. The most reliable approach to computing nearby interactions is to develop analytical formulas for panel integrals of Stokes kernels. For the FastStokes program, a fast analytical kernel integration algorithm was developed based on an extension of the method in [10]. This precise algorithm is given below.

Local Coordinate System

To simplify the calculations, a local Cartesian coordinate system (ξ, η, ζ) is set up so that the panel is in the $\xi-\eta$ coordinate plane with the centroid at the origin. Major computations of the kernel integration are done in the local coordinate system and the solutions are then transferred back to the global coordinate system. Transition between the local coordinate system (ξ, η, ζ) and the global coordinate system (x, y, z) can be expressed as

$$\begin{bmatrix}\xi\\ \eta\\ \zeta\end{bmatrix}_{3\times1} = \begin{bmatrix}\text{Coordinate}\\ \text{Transformation}\\ \text{Matrix}\end{bmatrix}_{3\times3}\left\{\begin{bmatrix}x\\ y\\ z\end{bmatrix}_{3\times1} - \text{Centroid}\right\}$$
$$\begin{bmatrix}X\\ Y\\ Z\end{bmatrix}_{3\times1} = \begin{bmatrix}\text{Coordinate}\\ \text{Transformation}\\ \text{Matrix}\end{bmatrix}_{3\times3}\left\{\begin{bmatrix}X'\\ Y'\\ Z'\end{bmatrix}_{3\times1} - \text{Centroid}\right\} \tag{A1}$$

where (X, Y, Z) is the local coordinate of the evaluation point and (X', Y', Z') is the corresponding global coordinate.

Integration of the Stokes Kernels

In [10], the Gauss–Bonnet theorem was used to show that the potential due to a constant -4π normal dipole distributed over the flat panel is given by (A2) at the bottom of the next page, where r is the distance between the evaluation point and a point on the panel; R_i is the distance between the evaluation point and the ith panel corner; ξ_i and η_i are the local coordinate of the ith panel corner; $\delta\xi_i = \xi_{i+1} - \xi_i, \delta\eta_i = \eta_{i+1} - \eta_i$; and ns is the number of corners. Integrating Φ in the direction of the panel normal yields Ψ, which is the potential due to -4π monopole distribution over a flat panel. The resulting formula is

$$\begin{aligned}\Psi &= \iint \frac{1}{r} d\xi\, d\eta \\ &= \sum_{i=1}^{ns} \left[(X-\xi_i)\sin\theta_i - (Y-\eta_i)\cos\theta_i\right] Q_i - Z\Phi\end{aligned}$$

and

$$Q_i = \log \frac{R_i + R_{i+1} + s_i}{R_i + R_{i+1} - s_i}. \tag{A3}$$

Here, θ_i is the polar angle of the ith edge and s_i is the length of the ith edge. Furthermore, the potentials due to linear, bilinear, and higher order dipole distributions can be obtained in a similar way, i.e.,

$$\begin{aligned}\begin{pmatrix}\Phi_x \\ \Phi_y\end{pmatrix} &= Z \iint \begin{pmatrix}\xi \\ \eta\end{pmatrix} \frac{d\xi\, d\eta}{r^3} \\ &= \begin{pmatrix}X \\ Y\end{pmatrix} \Phi \pm Z \sum_{i=1}^{ns} Q_i \begin{pmatrix}\sin\theta_i \\ \cos\theta_i\end{pmatrix} \\ \Phi_{xy} &= X\Phi_y + Y\Phi_x - XY\Phi \\ &\quad + Z \sum_{i=1}^{ns} \cos\theta_i \left[v_i Q_i \sin\theta_i - (R_{i+1} - R_i)\cos\theta_i\right] \\ \Phi_{xx} &= \Psi + \sum_{i=1}^{ns} \Bigg\{ (R_{i+1} - R_i)\cos\theta_i \sin\theta_i \\ &\qquad + (\xi_i + u_i\cos\theta_i - X) \\ &\qquad \times \sin\theta_i \ln\left(\frac{R_{i+1} - U_i}{R_i - u_i}\right)\Bigg\} \\ \Phi_{YY} &= \Psi - \sum_{i=1}^{ns} \Bigg\{ (R_{i+1} - R_i)\cos\theta_i \sin\theta_i \\ &\qquad - (\eta_i + u_i\sin\theta_i - Y) \\ &\qquad \times \cos\theta_i \ln\left(\frac{R_{i+1} - U_i}{R_i - u_i}\right)\Bigg\}\end{aligned} \tag{A4}$$

where $(u_i, -v_i)$ and $(U_i, -V_i)$ are real and imaginary parts of two 2-D vectors starting from the ith corner and the $(i+1)$th corner individually; both vectors end at the projection of the evaluation point on the ith edge.

Transferring Local Solutions Back to the Global Coordinate System

The above solutions are local solutions that must be transferred back to the global coordinate system. Here, we offer a simple approach to the Stokes kernels. Assume that the solutions in the local coordinate system and in the global coordinate system are defined as

$$\begin{aligned}\Phi_{m,n,k}^{\text{local}} &= \iint \frac{1}{r^3}(X-\xi)^m (Y-\eta)^n Z^k ds \\ \Phi_{m,n,k}^{\text{global}} &= \iint \frac{1}{r^3}(X'-x)^m (Y'-y)^n (Z'-z)^k ds\end{aligned}$$

and

$$\begin{aligned}[\Phi]_1^{\text{local}} &= \begin{bmatrix}\Phi_{1,0,0}^{\text{local}} \\ \Phi_{0,1,0}^{\text{local}} \\ \Phi_{0,0,1}^{\text{local}}\end{bmatrix} \\ [\Phi]_2^{\text{local}} &= \begin{bmatrix}\Phi_{2,0,0}^{\text{local}} & \Phi_{1,1,0}^{\text{local}} & \Phi_{1,0,1}^{\text{local}} \\ \Phi_{1,1,0}^{\text{local}} & \Phi_{0,2,0}^{\text{local}} & \Phi_{0,1,1}^{\text{local}} \\ \Phi_{1,0,1}^{\text{local}} & \Phi_{0,1,1}^{\text{local}} & \Phi_{0,0,2}^{\text{local}}\end{bmatrix} \\ [C] &= \begin{bmatrix}\text{Coordinate} \\ \text{Transformation} \\ \text{Matrix}\end{bmatrix}_{3\times 3}.\end{aligned} \tag{A5}$$

Then applying coordinate transition equations in (A1) yields

$$\begin{aligned}\Phi_{0,0,0}^{\text{global}} &= \Phi_{0,0,0}^{\text{local}} \\ \Psi_{0,0,0}^{\text{global}} &= \Psi_{0,0,0}^{\text{local}} \\ [\Phi]_1^{\text{global}} &= [C]^T [\Phi]_1^{\text{local}} \\ [\Phi]_2^{\text{global}} &= [C]^T [\Phi]_2^{\text{local}} [C].\end{aligned} \tag{A6}$$

Acknowledgment

The authors would like to thank J. Phillips and B. Buchmann for providing their software for the PFFT algorithm, and D. Freeman and W. Hemmert for supplying the comb drive example and measured data.

$$\begin{aligned}\Phi &= Z \iint \frac{1}{r^3} d\xi\, d\eta \\ &= \sum_{i=1}^{ns} \Bigg\{ \tan^{-1}\left(\frac{\delta\eta_i\left[(X-\xi_i)^2 + Z^2\right] - \delta\xi_i (X-\xi_i)(Y-\eta_i)}{R_i Z \delta\xi_i}\right) \\ &\qquad - \tan^{-1}\left(\frac{\delta\eta_i\left[(X-\xi_{i+1})^2 + Z^2\right] - \delta\xi_i (X-\xi_{i+1})(Y-\eta_{i+1})}{R_{i+1} Z \delta\xi_i}\right)\Bigg\}\end{aligned} \tag{A2}$$

REFERENCES

[1] J. Voldman, M. L. Gray, and M. A. Schmidt, "Microfabrication in biology and medicine," *Annu. Rev. Biomed. Eng.*, vol. 1, pp. 421–425, Aug. 1999.

[2] A. F. Mills, *Heat Transfer*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[3] C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[4] V. Rokhlin, "Rapid solution of integral equation of classical potential theory," *J. Comput. Phys.*, vol. 60, no. 2, pp. 187–207, Sep. 1985.

[5] K. Nabors and J. White, "FastCap: A multipole-accelerated 3-D capacitance extraction program," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 10, no. 10, pp. 1447–1459, Nov. 1991.

[6] L. Greengard, *The Rapid Evolution of Potential Fields in Particle Systems*. Cambridge, MA: MIT Press, 1988.

[7] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, no. 10, pp. 1059–1072, Oct. 1997.

[8] Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, Jul. 1986.

[9] W. Ye, J. Kanapka, X. Wang, and J. White, "Efficiency and accuracy improvements for FastStokes, a precorrected-FFT accelerated 3-D Stokes solver," in *Proc. Modeling and Simulation of Microsystems (MSM)*, San Juan, PR, 1999, pp. 502–505.

[10] J. N. Newman, "Distribution of sources and normal dipoles over a quadrilateral panel," *J. Eng. Math.*, vol. 20, no. 2, pp. 113–126, 1986.

[11] J. Tausch, "Rapid solution of Stokes flow using multiscale Galerkin BEM," in *Proc. Appl. Math. Mech. (PAMM)*, vol. 1, no. 1, pp. 8-11, Mar. 2002.

[12] X. Wang, "FastStokes: A fast 3-D fluid simulation program for micro-electro-mechanical systems," Ph.D. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, Jun. 2002.

[13] W. Ye, X. Wang, W. Hemmert, D. Freeman, and J. White, "Air damping in lateral oscillating micro resonators: A numerical and experimental study," *J. Microelectromech. Syst.*, vol. 12, no. 5, pp. 557–566, Oct. 2003.

[14] A. B. Basset, *A Treatise on Hydrodynamics*. Cambridge, U.K.: Cambridge Univ. Press, 1888.

[15] R. W. Barber and D. R. Emerson, *Advances in Fluid Mechanics IV*. Billerica, MA: WIT Press, 2002, pp. 207–216.

[16] J. Ding and W. Ye, "A fast integral approach for drag force calculation due to oscillatory slip Stokes flows," *Int. J. Numer. Methods Eng.*, vol. 60, no. 9, pp. 1535–1567, May 2004.

Xin Wang (S'99–M'01) was born in Harbin, China, in 1972. He received the B.S. degree in mechanical engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1995, the M.S. degree in mechanical engineering from Washington University, St. Louis, MO, in 1998, and the Ph.D. degree in electrical engineering and computer sciences from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002.

From September 1998 to May 2002, he was a Research Assistant at the Research Laboratory of Electronics and the Electrical Engineering and Computer Science Department, MIT. He is currently with Synopsys Inc., Mountain View, CA. His research interests include modeling and simulation of large circuit and microelectromechanical systems.

Joe Kanapka received the Ph.D. degree in computer science from the Massachusetts Institute of Technology, Cambridge, in 2002.

He is currently a Software Developer at the Physical Modeling Group, The MathWorks, Inc., Natick, MA. His research interests are in the area of numerical algorithms, specifically in efficient representations of densely coupled parasitic effects for use in circuit simulation, and more generally in dynamic simulation of physical systems, with a side interest in large-scale search problems in computational number theory.

Wenjing Ye received the B.S. degree in mechanical engineering from the University of Science and Technology of China, Beijing, the S.M. degree in mechanical engineering from the University of California at San Diego, and the Ph.D. degree in theoretical and applied mechanics from Cornell University, Ithaca, NY.

From 1998 to 1999, she was a Postdoctoral Associate at the Research Lab of Electronics, Massachusetts Institute of Technology. She is currently an Assistant Professor at the Georgia Institute of Technology, Atlanta, GA. Her current research interests are in modeling and design of micro/nanosystems and advanced numerical techniques.

Narayan R. Aluru (M'00) received the B.E. degree (with honors and distinction) from the Birla Institute of Technology and Science (BITS), Pilani, India, in 1989, the M.S. degree from Rensselaer Polytechnic Institute, Troy, NY, in 1991, and the Ph.D. degree from Stanford University, Stanford, CA, in 1995.

From 1995 to 1997, he was a Postdoctoral Associate at the Massachusetts Institute of Technology (MIT), Cambridge. In 1998, he was an Assistant Professor at the University of Illinois at Urbana-Champaign (UIUC). He is currently an Associate Professor at the Department of Mechanical and Industrial Engineering, UIUC.

Dr. Aluru is a Subject Editor for the IEEE/ASME JOURNAL OF MICROELECTROMECHANICAL SYSTEMS, an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II, and serves on the Editorial Board of a number of other journals. He received the National Science Foundation (NSF) CAREER award and the NCSA Faculty Fellowship in 1999, the 2001 CMES Distinguished Young Author Award, the 2001 Xerox Award for Faculty Research, and the Willett Faculty Scholar Award in 2002.

Jacob White (A'88) received the B.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, and the S.M. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley.

From 1985 to 1987, he was with the IBM T. J. Watson Research Center. From 1987 to 1989, he was an Analog Devices Career Development Assistant Professor at MIT, and was a Presidential Young Investigator in 1988. He is currently a C. H. Green Professor of Electrical Engineering and Computer Science and an Associate Director of the Research Laboratory of Electronics at MIT. His current research interests are in numerical algorithms for problems in circuits, interconnect, micromachining for biological applications, biomolecule design, and systems biology.

Dr. White was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN from 1992 to 1996, and was the Chair of the International Conference on Computer-Aided Design in 1999.