A Fast Integral Equation Technique For Analysis Of Microflow Sensors Based on Drag Force Calculations

N. R. Aluru[+] and J. White[*]

[+]Assistant Professor, Department of General Engineering
3221 Beckman Institute, MC-251, 405 N. Mathews Avenue
University of Illinois at Urbana-Champaign, Urbana, IL 61801
e-mail: aluru@uiuc.edu, http://www-ceg.ceg.uiuc.edu/~aluru/

[*]Professor, Department of Electrical Engineering and Computer Science
36-817, Research Laboratory of Electronics, 50 Vassar Street
Massachusetts Institute of Technology, Cambridge, MA 02139
e-mail: white@rle-vlsi.mit.edu, http://sobolev.mit.edu/~white/

ABSTRACT

It is well known that MEMS based microfluidic devices operate in very low Reynolds number regime ($Re < 1$). Analysis and design of such microfluidic devices requires the solution of incompressible viscous fluid flow or incompressible Stokes equations. In this paper we present a new accelerated boundary-element method to rapidly solve the integral form of the Stokes equations. The computational complexity of the proposed approach is $O(Nlog(N))$, where N is the number of panels used to discretize the surface of the micromachined device.

Keywords: Microfluidic Devices, MEMS, Boundary Element Methods, Precorrected FFT Techniques, Microflow Sensors

INTRODUCTION

Many integrated microfluidic systems such as chemical analysis and liquid dosage systems contain a microflow sensor as one of the components. Microflow sensors have been designed based on several principles: mass flow, ion pulse generation, differential pressure, hot wire anemometry and drag force [2]. The key design problem in a drag force based microflow sensor is determining the shape of the obstacle on which the fluid will exert a drag or a force. These obstacles are innately three-dimensional and typically geometrically complicated making them expensive to simulate with standard finite-element based solvers. The drag on micromachined devices can be computed by the solution of Stokes equations.

In this paper we present a new approach, the precorrected FFT accelerated boundary-element method, to rapidly solve the integral form of the Stokes equations to compute the drag on complicated micromachined obstacles. The computational complexity of the approach is $O(Nlog(N))$, where N is the number of panels used to discretize the surface of an obstacle, and so it can be used to analyze complicated obstacles in a few minutes.

PROBLEM FORMULATION

The fluidic forces on an arbitrary and complicated micromachined device can be computed by considering the flow of an incompressible viscous fluid past the arbitrary shaped micromachined device at very low Reynolds number. The governing equations of an incompressible viscous fluid at very low Reynolds number are summarized as (see e.g. [6])

$$\mu \frac{\partial^2 v_i(x)}{\partial x_j \partial x_j} = \frac{\partial p(x)}{\partial x_i} \quad (1)$$

$$\frac{\partial v_i(x)}{\partial x_i} = 0 \quad (2)$$

where μ is the viscosity of the fluid, $x = (x_1, x_2, x_3) \in \Omega_e$, Ω_e is the three dimensional unbounded domain exterior to the micro object, $v = (v_1, v_2, v_3)$ is the fluid velocity vector, v_i denotes the $i-th$ component of the fluid velocity and p is the pressure. The boundary of the micro object is denoted by an arbitrary surface S. Equations (1) - (2) are also referred to as the Stokes equations.

Since the governing equations are linear and are posed in an exterior domain, an efficient approach, without resorting to meshing the exterior of the domain, is to consider a boundary-integral or a boundary-element formulation for the Stokes equations. For Stokes equations a boundary-integral formulation can be developed involving quantities defined only on the surface of the object. For microelectromechanical applications, the deforming or the moving object interacts with the fluid only on the surface, and the fluidic forces need to be computed only on the surface on the object and not in the interior of the object. For this reason, boundary-element methods seem best suited for modeling the Stokes equations for microelectromechanical applications. Starting from

the original work by Youngren and Acrivos in 1975 [9], boundary-element methods for Stokes equations have been considered by several authors (see e.g. [6], [7] and references therein).

The integral equation for the velocity at a point x on the surface of the body is given as

$$v_i(x) = \frac{-1}{4\pi\mu}\int_S G_{ij}(x,y)f_j(y)dS_y + \frac{3}{2\pi}\int_S T_{ijk}(x,y)n_j(y)v_k(y)dS_y \qquad (3)$$

$$x \in S$$

where

$$G_{ij}(x,y) = \frac{\delta_{ij}}{r_{xy}} + \frac{(x_i-y_i)(x_j-y_j)}{r_{xy}^3} \qquad (4)$$

$$T_{ijk}(x,y) = \frac{(x_i-y_i)(x_j-y_j)(x_k-y_k)}{r_{xy}^5} \qquad (5)$$

In the above equations, v_i is the $i-th$ component of the velocity vector, $x = (x_1, x_2, x_3)$ is the source point, $y = (y_1, y_2, y_3)$ is the field point, $r_{xy} = \|x-y\|$ is the distance between points x and y, $n = (n_1, n_2, n_3)$ is the unit outward normal vector, $f_j = \sigma_{ij}n_j$ is the $j-th$ component of the traction vector, and the stress tensor σ_{ij} is defined as

$$\sigma_{ij} = -p\delta_{ij} + \mu(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}). \qquad (6)$$

In this paper we will focus on the case where the velocities on the surface of the object are prescribed and the unknowns are the tractions on the surface. In particular, the velocity vector v is assumed to satisfy the no-slip boundary condition on the surface of the object i.e.

$$v_i(x) = V_i(x) \quad x \in S. \qquad (7)$$

When the prescribed velocities are constant (i.e. $v_i(x) = V_i$), equation (3) simplifies to

$$v_i(x) = \frac{-1}{8\pi\mu}\int_S G_{ij}(x,y)f_j(y)dS_y. \qquad (8)$$

A standard approach to numerically solving (3) for the tractions (or forces) is to use a piece-wise constant collocation scheme. That is, the surface of the object is broken up into n small panels, and it is assumed that on each panel a force is uniformly distributed. The result is a dense linear system

$$Af = g \qquad (9)$$

where $A \in \Re^{(3\cdot n)\times(3\cdot n)}$, $f \in \Re^{(3\cdot n)}$ is the vector of panel forces, $g \in \Re^{(3\cdot n)}$ is the vector of known right-hand side involving velocity terms, and

$$A_{IJ}(m,n) = \int_{\Delta_J} \left(\frac{\delta_{mn}}{r_{x^{(I)}y^{(J)}}} + \frac{(x_m^{(I)}-y_m^{(J)})(x_n^{(I)}-y_n^{(J)})}{r^3_{x^{(I)}y^{(J)}}} \right) dS_y \qquad (10)$$

$$g_j^{(I)} = -4\pi\mu\left(V_i^{(I)} - \frac{3}{2\pi}\int_S T_{jkl}(x^{(I)},y)n_k(y)V_l(y)\right)dS_y \qquad (11)$$

$$T_{jkl}(x^{(I)},y) = \frac{(x_j^{(I)}-y_j)(x_k^{(I)}-y_k)(x_l^{(I)}-y_l)}{r_{xy}^5} \qquad (12)$$

$$g = \left(g_1^{(1)}, g_2^{(1)}, g_3^{(1)}, \ldots, g_1^{(n)}, g_2^{(n)}, g_3^{(n)}\right)^T \qquad (13)$$

where $m,n = 1,2,3$; $I,J = 1,2,\ldots,n$; $x_m^{(I)}$ is the $m-th$ coordinate of source point I, $y_n^{(J)}$ is the $n-th$ coordinate (m,n take values of 1, 2 and 3) of field point J, and $r_{x^{(I)}y^{(J)}} = \|x^{(I)} - y^{(J)}\|$ is the distance between source and field points. Note that each A_{IJ} is a 3×3 matrix and is symmetric (i.e. $A_{IJ}(m,n) = A_{IJ}(n,m)$). Note, however, that A is not symmetric, $A_{IJ} \neq A_{JI}$.

The dense linear system of (9) can be solved to compute the unknown force vector on each panel. The direct approach of solving (9) via Gaussian elimination, which requires $O((3\cdot n)^3)$ operations and $O((3\cdot n)^2)$ storage, becomes computationally intractable if the number of panels exceeds several hundred. If, instead of Gaussain elimination, an iterative approach such as GMRES [8] is used to solve (9), then each iteration of the GMRES will cost $(3\cdot n)^2$ operations. This is because the matrix in (9) is dense, and therefore evaluating candidate solution vectors involves a dense matrix-vector multiply. Several sparsification techniques for A are based on the idea of directly computing only those portions of Af associated with interactions between panels which are close to each other. The rest of Af is then somehow approximated to accelerate the computation [3], [1], [4].

PRECORRECTED FFT METHOD

The acceleration technique proposed in this paper is the precorrected FFT technique, which was originally developed for the solution of 3-D potential equations [5] for capacitance calculations. While the Green's function for the potential equation is scalar, the Green's function for the Stokes equations is vectorial (see equation (4)). In this paper, the precorrected FFT approach is extended and developed for the vectorial Stokes equations. The key idea in the precorrected FFT method is that the interaction between nearby panels is computed directly and the interaction between far-off panels is computed using a coarse grid.

In the precorrected FFT approach, the object which has been discretized into n small panels is enclosed by a parallepiped. The parallepiped is subdivided into a $k\times l\times m$ array of small cubes so that each small cube contains only a few panels. Figure 1 shows a $3\times 3\times 3$ array of cubes enclosing a discretized three-finger comb structure. An approach to compute distant interactions is to represent the given cell's force distribution with a small number of weighted point forces. For example, in Figure 1(b), each cell is represented by a $3\times 3\times 3$ array

of point forces. If the point forces all lie on a uniform grid, for example at cell vertices, then the computation of the velocities at the grid points due to grid forces is a discrete convolution which can be performed using the FFT. The precorrected FFT technique to compute

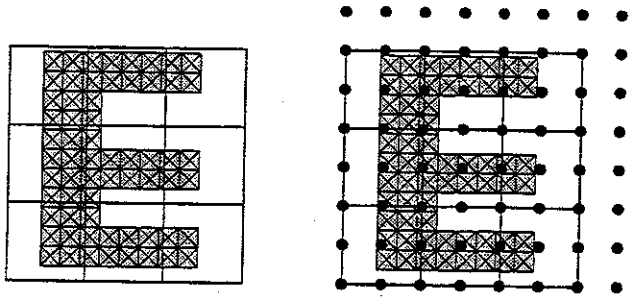

Figure 1: (a) A three-finger comb structure discretized into panels. The comb structure is enclosed by a $3\times3\times3$ array of cubes. (b) Superimposed grid corresponding to the cell decomposition. In each cell a $3 \times 3 \times 3$ array of grid points is used to represent the long range interactions between panels. The grid extends outside the problem domain because the number of grid points is required to be a factor of two for FFT calculations

the matrix-vector product Af (the matrix-vector product has dimensions of velocity and this product will be referred to as the computation of velocity in the precorrected FFT approach) is summarized in Figure 2 and is implemented in four steps:

1. projection of panel forces onto a uniform grid of point forces
2. computation of grid velocities due to grid forces using an FFT
3. interpolation of grid velocities to panel velocities and
4. computation of precorrected direct interactions or nearby interactions

Only the first step is described here in detail, and the remaining steps follow from the work described in [5].

Projection of Panel Forces

To compute the grid forces used to represent the forces in cube k, consider the following scheme: suppose a $p \times p \times p$ array of grid forces is used to represent the forces in a cube. For example, in Figure 1(b) a $3 \times 3 \times 3$ ($p = 3$) grid is employed to represent the cube forces. Next, N_c test points are selected on the surface of a sphere of radius r_c which has cube k as its center. Then, velocities due to the $G = p^3$ grid forces are forced to match the velocities due to the cube's actual force distribution at the test points, i.e.

$$H^{gt}\hat{f}(k) = H^{ft}f(k) \tag{14}$$

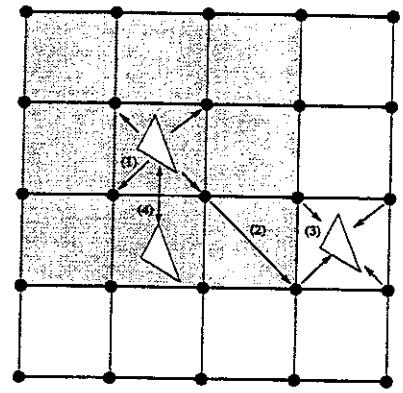

Figure 2: 2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (grey area) are computed directly, interactions between distant panels are computed using the grid.

where $f(k) \in \Re^{3\cdot N_p\times1}$ is a vector of forces in cube k, $\hat{f}(k)$ $\in \Re^{3\cdot G\times1}$ is a vector of grid forces (a hat symbolizes that the quantity is associated with the grid), $H^{gt} \in \Re^{(3\cdot N_c)\times(3\cdot G)}$ is the mapping between grid forces and velocities at the test points and is given by

$$H^{gt}_{IJ}(m,n) = \frac{\delta_{mn}}{r_{z^{(I)}y^{(J)}}} + \frac{(y_m^{(J)} - z_m^{(I)})(y_n^{(J)} - z_n^{(I)})}{r^3_{z^{(I)}y^{(J)}}} \tag{15}$$

$I = 1,2,\ldots,N_c$; $J = 1,2,\ldots,G$; $m,n = 1,2,3$; $y_m^{(J)}$ is the mth coordinate of grid node J, $z_n^{(I)}$ is the nth coordinate of test point I, and $r_{z^{(I)}y^{(J)}}$ is the distance between points $z^{(I)}$ and $y^{(J)}$. Note that each H^{gt}_{IJ} is a 3×3 matrix. $H^{ft} \in \Re^{(3\cdot N_c)\times(3\cdot N_p)}$ is the mapping between panel forces and velocities at the test points and is given by

$$H^{ft}_{IJ}(m,n) = \int_{\Delta_J} \frac{\delta_{mn}}{r_{z^{(I)}x^{(J)}}} + \frac{(x_m^{(J)} - z_m^{(I)})(x_n^{(J)} - z_n^{(I)})}{r^3_{z^{(I)}x^{(J)}}} \tag{16}$$

$I = 1,2,\ldots,N_c$; $J = 1,2,\ldots,N_p$; $m,n = 1,2,3$; N_p is the number of panels in cube k, $x_m^{(J)}$ is the mth coordinate (centroid) of panel J, $z_n^{(I)}$ is the nth coordinate of test point I, and $r_{z^{(I)}x^{(J)}}$ is the distance between points $z^{(I)}$ and $x^{(J)}$. Note that each H^{ft}_{IJ} is a 3×3 matrix. The vector of grid forces can then be computed as

$$\hat{f}(k) = W(k)f(k) \tag{17}$$

where

$$W(k) = \left[H^{gt}\right]^{\dagger}\left[H^{ft}\right] \tag{18}$$

$[H^{gt}]^{\dagger}$ indicates the pseudo-inverse of matrix H^{gt}. $W(k)$ $\in \Re^{(3\cdot G)\times(3\cdot N_p)}$ is defined as the projection operator for cube/cell k and the accuracy of the projection operator hinges on the proper selection of the test points on the sphere. The accuracy of the projection operator can be

improved by selecting the test points to be abscissas of a high-order quadrature rule.

RESULTS

Numerical results are presented for an infinite fluid flow past three obstacles: a spheroid, a beam and a comb structure. An analytical solution is available for the spheroid example, and a comparison between direct and precorrected FFT methods for the drag and CPU time is presented in Table 1. Note that the precorrected FFT method is much faster than the direct method and the CPU time for the precorrected FFT method grows as $Nlog(N)$. The beam structure considered here is $80\mu m$ long, $10\mu m$ wide and $10\mu m$ thick and the drag on this beam in a infinite fluid flow for constant velocities is plotted in Figure 3(a). Note that the drag increases linearly as a function of velocity i.e. as the velocity increases the drag also increases. In these calculations, the viscosity of the fluid is assumed to be one. The drag on the beam for several different fluids can be computed by simply multiplying the value given in Figure 3(a) by the viscosity of the fluid. If the viscosity of the fluid is not known, then results such as those in Figure 3 can be used to determine the viscosity of the fluid i.e by knowing the obstacle (beam, in this case) and the velocity of the fluid, one can compute the material properties (such as viscosity) of an unknown fluid.

Table 1: Comparison of direct and precorrected FFT (P-FFT) methods for the drag and CPU(sec) for the spheroid example. The analytical drag is 22.69. A * indicates that the technique requires more than 1 GB of memory

# Panels	Drag		CPU(sec)	
	Direct	P-FFT	Direct	P-FFT
192	22.14	21.97	26	8
768	22.54	22.46	561	40
3072	22.66	22.63	30785	415
12288	*	22.66	*	2943

The final example considered here is a comb structure as shown in Figure 4. The drag per unit viscosity as a function of fluid velocity for the comb structure is plotted in Figure 3(b). The comb structure considered here has more than 6000 panels and the precorrected FFT accelerated boundary-element method takes under 10 minutes to compute the drag on this structure, while the direct method is about 300 times slower.

ACKNOWLEDGEMENTS

The authors thank Dr. J. Phillips for several valuable discussions and this work is supported under the DARPA composite CAD program by contracts J-FBI-95-215 and F30602-97-2-0333.

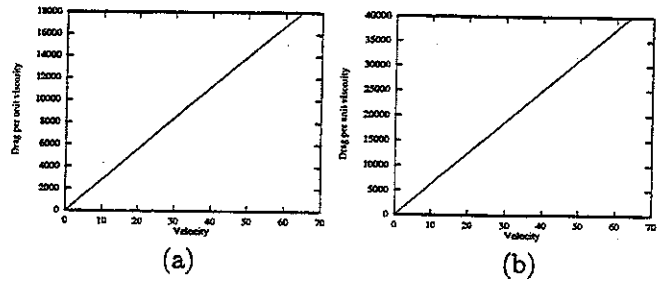

Figure 3: Drag per unit viscosity as a function of the fluid velocity for (a) a beam structure and (b) a comb structure

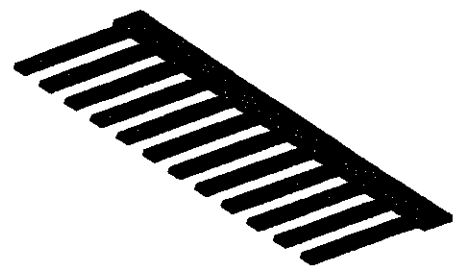

Figure 4: Infinite fluid flow past a comb

REFERENCES

[1] A. Brandt, A. Lubrecht, "Multilevel matrix multiplication and fast solution of integral equations," *J. of Comp. Phys.*, vol. 90, pp. 348-370, 1990.

[2] V. Gass, B. van der Schoot, N. de Rooij, "Nanofluid handling by microflow sensor based on drag force measurements," *Proc. of the IEEE MEMS Workshop*, pp. 167-172, 1993.

[3] L. Greengard, *The rapid evaluation of potential fields in particle systems*. Cambridge, Massachusetts: M.I.T. Press, 1988.

[4] L. Greengard, M. Kropinski, A. Mayo, "Integral equation methods for Stokes flow and isotropic elasticity in the plane," *Journal of Computational Physics*, vol. 125, no. 2, pp. 403, 1996.

[5] J. Phillips, "Rapid solution of potential integral equations in complicated 3-dimensional geometries", *Ph.D. Thesis*, M.I.T., 1997.

[6] C. Pozrikidis, *Introduction to theoretical and computational fluid dynamics*, Oxford University Press, New York, 1997.

[7] H. Power, L. Wrobel, *Boundary integral methods in fluid mechanics*, Computational Mechanics Publications, 1995.

[8] Y. Saad, M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856-869, July 1986.

[9] G. Youngren, A. Acrivos, "Stokes flow past a particle of arbitrary shape: a numerical method of solution," *J. Fluid Mech.*, vol. 69, pp. 377-403, 1975.