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@elec.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 $O(N\log(N))$, 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 microfluidic 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 microfluidic sensor is determining the shape of the obstacle on which the fluid will exert a drag or a force. These obstacles are inately 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 $O(N\log(N))$, 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])

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

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

where $\mu$ is the viscosity of the fluid, $x = (x_1, x_2, x_3) \in \Omega$, $\Omega$ 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 Acivos 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 object is given as

\[
 v_i(x) = \int_S G_i(x, y) f_j(y) dS_y + \frac{1}{2\pi} \int_S T_{ijk}(x, y) n_j(y) u_k(y) dS_y
\]

where

\[
 G_i(x, y) = \frac{\delta_i - y_j(x_j - y_j)}{r_{xy}} \\
 T_{ijk}(x, y) = \frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)}{r_{xy}^3}
\]

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 \( y \), and \( n = (n_1, n_2, n_3) \) is the unit outward normal vector, \( f_j = \sigma_{ij} u_j \) is the \( j \)-th component of the traction vector, and the stress tensor \( \sigma_{ij} \) is defined as

\[
 \sigma_{ij} = -\rho \delta_{ij} + \mu \left( \frac{\delta_{ij}}{2\xi} + \frac{\delta_{ijk}}{2\xi_k} \right).
\]

In this paper we will focus on the case of the velocity \( v \) on the surface of the object; in particular, the velocity vector \( u \) is assumed to satisfy the no-slip boundary condition on the surface of the object. The velocity \( u \) is prescribed and the unknowns are the tractions on the surface. From equation (3), it follows that

\[
 v_i(x) = V_i(x) \quad x \in S.
\]

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

\[
 v_i(x) = -\frac{1}{8\pi\mu} \int_S G_i(x, y) f_j(y) dS_y.
\]

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

\[
 A f = g
\]

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

\[
 A_{ij}(m, n) = \int_{\Delta_j} \left( \frac{\delta_{ij}}{r_{ij}^3} + (\frac{\delta_{ij}}{2\xi_j}) (\frac{\delta_{ijk}}{2\xi_k}) \right) dS_y
\]

where \( m, n = 1, 2, 3 \), \( I, J = 1, 2, \ldots, n \); \( x_m^{(I)} \) is the \( m \)-th coordinate of source point \( J \), \( y_n^{(I)} \) is the \( n \)-th coordinate of source point \( J \), and \( \xi_{n^{(I)}, m^{(J)}} = ||x^{(I)} - y^{(J)}|| \) is the distance between source and field points. Note that each \( A_{ij} \) is a \( 3 \times 3 \) matrix and is symmetric (i.e. \( A_{ij}(m, n) = A_{ij}(n, m) \)). 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. Instead of Gaussian elimination, an iterative approach such as GMRES [8] is used. If \( \epsilon \) is the tolerance, 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 \( A f \) associated with interactions between panels which are close to each other. The rest of \( A f \) is then somehow approximated to accelerate the computation [9], [11], [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 parallelepiped. The parallelepiped 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 3-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

\[ H^s_f(k) = H^f(k) \]

\[ f(k) \in \mathbb{R}^{3 \times N_p \times 1} \] is a vector of forces in cube k, \( f(k) \in \mathbb{R}^{3 \times G \times 1} \) is a vector of grid forces (a hat symbolizes that the quantity is associated with the grid). \( H^D \in \mathbb{R}^{(2 \times N_t) \times (2 \times G)} \) is the mapping between grid forces and velocities at the test points and is given by

\[
H^D_{ij}(m,n) = \frac{\delta_{mn}}{r_{D(j)g(i)}} + (g_{m}^{(j)} - z_{m}^{(i)}) (g_{n}^{(j)} - z_{n}^{(i)})
\]

\[ I = 1, 2, \ldots, N_t; \; J = 1, 2, \ldots, G; \; m, n = 1, 2, 3; \; g_{m}^{(j)} \] is the mth coordinate of grid node \( J; \; z_{m}^{(i)} \) is the mth coordinate of test point \( I; \; r_{D(j)g(i)} \) is the distance between points \( z^{(j)} \) and \( g^{(i)}. Note that each \( H^D_{ij} \) is a \( 3 \times 3 \) matrix. \( H^D \in \mathbb{R}^{2 \times N_t \times (2 \times G)} \) is the mapping between panel forces and velocities at the test points and is given by

\[
H^D_{ij}(m,n) = \int_{\Delta_i} \frac{\delta_{mn}}{r_{D(j)g(i)}} + (z_{m}^{(j)} - z_{m}^{(i)}) (z_{n}^{(j)} - z_{n}^{(i)})
\]

\[ I = 1, 2, \ldots, N_p; \; J = 1, 2, \ldots, N_p; \; m, n = 1, 2, 3; \; N_p \] is the number of panels in cube k, \( z_{m}^{(j)} \) is the mth coordinate (centroid) of panel j, \( z^{(j)} \) is the distance between points \( z^{(j)} \) and \( z^{(j)}. Note that each \( H^D_{ij} \) is a \( 3 \times 3 \) matrix.

The vector of grid forces can then be computed as

\[ f(k) = W(k) F(k) \]

where

\[ W(k) = [H^D]^{-1} [H^f] \]

\([H^D]^{-1}\) indicates the pseudo-inverse of matrix \( H^D \). \( W(k) \in \mathbb{R}^{(3 \times G) \times (2 \times N_p)} \) 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 abscissae 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 $N \log(N)$. The beam structure considered here is $50 \mu m$ long, $10 \mu m$ wide and $10 \mu m$ thick and the drag on this beam in an 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 22.66. A * indicates that the technique requires more than 1 GB of memory

<table>
<thead>
<tr>
<th># Panels</th>
<th>Drag</th>
<th>CPU(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Direct</td>
<td>P-FFT</td>
</tr>
<tr>
<td>192</td>
<td>22.14</td>
<td>21.97</td>
</tr>
<tr>
<td>768</td>
<td>22.54</td>
<td>22.46</td>
</tr>
<tr>
<td>3072</td>
<td>22.66</td>
<td>22.63</td>
</tr>
<tr>
<td>12288</td>
<td>*</td>
<td>22.66</td>
</tr>
</tbody>
</table>

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