# Efficiency Improvements in Fast Stokes Solvers

S. De[*], X. Wang[*] and J. K.White[*]

[*]Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139
Email: white@mit.edu

## ABSTRACT

In this paper we present several techniques to improve the efficiency and accuracy of the precorrected FFT accelerated Fast Stokes solver based on a boundary element discretization of the integral form of the incompressible Stokes flow equations. It is shown that a factor-of-three reduction of grid data storage may be achieved by deriving an alternative form of the Stokes kernels using second order derivatives of the distance function. We propose two new techniques of approximating the second derivative operators; using a separate projection operator for each derivative and a finite difference discretization. The implementation using the finite difference discretization is shown to result in an order of magnitude improvement in accuracy compared to the other.

***Keywords***: Stokes flow, microfluidic devices, boundary element method, precorrected FFT.

## 1 INTRODUCTION

The efficient computation of fluid drag forces on micromachined devices such as comb-drive based structures (see figure 1) is a key step in the design and analysis of such devices. At the length scales at which these devices operate the Reynolds number (Re) of the flow is typically quite small (Re<1). Moreover, fluid compression may be ignored for devices using lateral actuation. Hence, it is justified to neglect the inertia terms in the Navier-Stokes equations compared to the pressure and viscous terms and solve the corresponding Stokes flow problem (linear) for an incompressible fluid [1].

Due to the innately 3D nature of the devices and the complicated geometries involved, it is computationally advantageous to use a boundary integral formulation of the Stokes equations [2]. In ref. [1] a Fast Stokes solver was presented based on a precorrected FFT [3] acceleration of the boundary integral method. The computational complexity of the approach is O(nlog(n)), where 'n' is the number of panels used in the discretization of the surface of the device. This allowed the accurate computation of drag forces in a matter of minutes for complicated comb-drives involving about a hundred thousand degrees of freedom on reasonably fast workstations. However, the storage requirements for the convolution kernels was still substantial.

In the current paper we report a new technique of representing the Stokes kernels so as to obtain a factor of three reduction in storage. We present two new schemes of projecting the Stokes kernels to the grid. One scheme uses a separate projection operator for each derivative. The other scheme uses finite difference discretizations of the required derivatives.

In the next section, we summarize the integral form of the Stokes flow problem for a viscous incompressible fluid, and briefly review the boundary element discretization scheme as well as the precorrected FFT acceleration technique used. In section 3 we discuss how the panel forces can be projected to the grid and present two new strategies. Finally, in section 4 we present a numerical study comparing the different schemes.

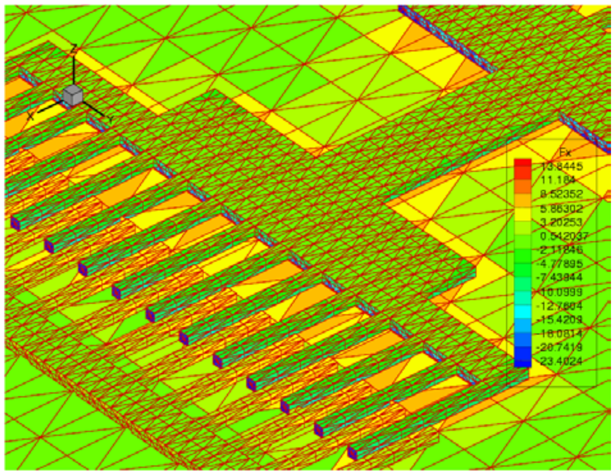

Figure 1: A comb-drive structure discretized into panels. The drag force in the x-direction ($F_x$) is shown in this figure.

## 2 THE STOKES FLOW PROBLEM

Let 'S' represent the surface of the device (assumed rigid) on which the fluid drag force is to be computed. The indirect (first kind) formulation of the exterior Dirichlet problem corresponding to the incompressible Stokes flow equations is given by [2]

$$v_i(\mathbf{x}_0) = -\frac{1}{8\pi\mu}\int_S G_{ij}(\mathbf{x}_0,\mathbf{x}) f_j(\mathbf{x})\, dS(\mathbf{x}) \tag{1}$$

$i, j \in \{1,2,3\}$. $v_i(\mathbf{x}_0)$ is the $i^{th}$ component of the velocity vector (assumed known) at the source point $\mathbf{x}_0$ located on the surface of the device, $f_j(\mathbf{x})$ is the $j^{th}$ component of the traction at the field point **x** also located on the surface of the device and $\mu$ is the fluid viscosity. The Green's functions $G_{ij}$ relating the traction components to the velocity components on the boundary of the device is traditionally given as (see for e.g. reference [2])

$$G_{ij} = \frac{\delta_{ij}}{r} + \frac{\hat{x}_i \hat{x}_j}{r^3}, \; i, j \in \{1,2,3\} \tag{2}$$

Here $\hat{x}_i = x_{0i} - x_i$ is the relative position vector between the source point and the field point; $r = \|\mathbf{x}_0 - \mathbf{x}\|_0$ and $\delta_{ij}$ is the Kronecker delta.

The key observation that has allowed us to reduce the storage requirements is that the Stokes kernels in equation (2) may be expressed in the following equivalent form

$$G_{ij} = \frac{2\delta_{ij}}{r} - \frac{\partial^2 r}{\partial x_i \partial x_j}. \tag{3}$$

While the form in equation (2) suggests that at least six independent kernels need to be stored for each grid point (taking into account the fact that $G_{ij} = G_{ji}$), the form in equation (3) implies that it is sufficient to store only two kernels; $r$ and $1/r$ and the different components of the Green's functions may be computed by differentiation.

## 2.1 Boundary element discretization

A piece-wise constant collocation scheme is used to discretize the set of equations (1). In this approach the surface of the device is discetized using 'n' small panels and the traction forces are assumed to be uniformly distributed on each panel. A square system of equations is obtained by insisting that (1) be satisfied at the centroid of each panel. This results in the dense linear system of equations

$$Af = g \tag{4}$$

where $f \in R^{3n}$ is the vector of panel forces, $g \in R^{3n}$ is the vector of nodal velocities (known) and $A \in R^{3n \times 3n}$ is the matrix given by its components

$$A_{IiJj} = \int_{S_J} G_{ij}(\mathbf{x}_I, \mathbf{x})\, dS(\mathbf{x}) \tag{5}$$

where $i, j = 1,2,3$ and $\mathrm{I}, \mathrm{J} = 1,2,\ldots,n$, $S_J$ is the surface of the $J^{th}$ panel and $\mathbf{x}_I$ is the position vector of the centroid of the $I^{th}$ panel.

## 2.2 Precorrected FFT acceleration

A direct solution of the dense linear system (4) using Gaussian elimination requires $O(n^3)$ operations and $O(n^2)$ storage and is therefore very expensive for n larger than a few hundred. An iterative scheme like GMRES [4], on the other hand, would require $O(n^2)$ operations per iteration. The precorrected FFT technique [3], originally developed for the solution of 3D potential equations, is a scheme that effectively sparsifies the matrix A by approximately computing "long-range" interactions through a coarse grid whereas computing the interactions between "nearby" panels directly. This allows the matrix-vector products in the inner loop of iterative schemes like GMRES to be computed in O(nlog(n)) operations. The precorrected FFT technique was extended to the Stokes flow problem in [1].

In the precorrected FFT approach the object which has been discretized into 'n' panels is tightly enclosed by a parallelopiped. The parallelopiped is subdivided into a $k \times l \times m$ array of small cubes (called "cells") so that each cell contains only a few small panels. Each cell is discretized using $p \times p \times p$ grid points. An approximation to the distant interactions may be achieved by projecting the panel forces in each cell onto the grid points of that cell. Then the computation of the velocities at the grid points due to the grid forces is nothing but a discrete convolution which may be effectively computed using the FFT. The precorrected FFT scheme is summarized in the following four steps [3]:

1. **Projection** of the panel forces onto a uniform grid of point forces.
2. **Convolution** using FFT to obtain the grid velocities from the grid forces.
3. **Interpolation** of the grid velocities to panel velocities.
4. **Precorrection** by computing the nearby interactions directly.

## 3 PROJECTION OF PANEL FORCES

In this section we describe how the panel forces may be projected onto the grid for the computation of long range interactions. Let us consider a panel P with centroid $\mathbf{x}_P$ (see figure 2) inside a cell which we denote as the "projection stencil". The grid force at grid point $\alpha \in \{1,2,\ldots,p^3\}$ due to a traction component $f_j$ ($j$=1,2,3) at $\mathbf{x}_P$ is $\hat{f}_j^{\alpha} = W_\alpha(\mathbf{x}_P) f_j$ where $W_\alpha(\mathbf{x}_P)$ is a complete tensor product Lagrangian polynomial of degree $(p-1)^3$ (having the property that it has a value of unity at node α and a value of zero at all the other nodes of the cell) evaluated at $\mathbf{x}_P$. The grid velocity component at a grid point $\beta \in \{1,2,\ldots,p^3\}$ in the "interpolation stencil" far removed from the projection stencil may be obtained using the following approximation

$$\hat{v}_i^\beta = \sum_j \sum_\alpha G_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) W_\alpha(\mathbf{x}_P) f_j = \sum_j M_{ij}^\beta f_j \tag{6}$$

where

$$M_{ij}^\beta = \sum_\alpha G_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) W_\alpha(\mathbf{x}_P) \tag{7}$$

is the operator that acts on the traction component $f_j$ to produce the velocity component $\hat{v}_i^\beta$.

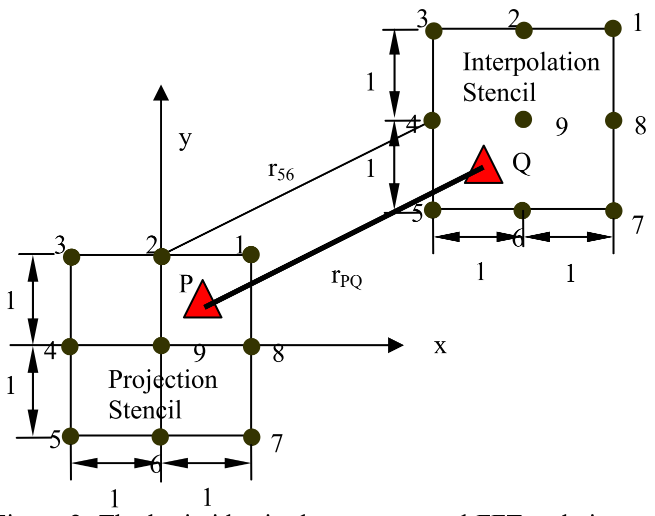

Figure 2: The basic idea in the precorrected-FFT technique is to compute the long-range interactions approximately. This figure shows how the approximation to the Stokes kernels can made using projection and interpolation stencils in a fictitious 2D example.

The original implementation of the Fast Stokes code uses equation (6) with the Green's function in equation (2).

## 3.1 Improved scheme using multiple projection operators

In this paper we use the Stokes kernels given in equation (3) which we write as

$$G_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) = \overline{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) - \widetilde{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) \tag{8}$$

where

$$\overline{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) = \frac{2\delta_{ij}}{r_{\alpha\beta}} \text{ and } \widetilde{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) = \frac{\partial^2 r_{\alpha\beta}}{\partial x_i \partial x_j} \tag{9}$$

with $r_{\alpha\beta} = \left\| \mathbf{x}_\alpha - \mathbf{x}_\beta \right\|_0$. $M_{ij}^\beta$ in equation (7) now has two components. We may write $M_{ij}^\beta$ alternatively as

$$M_{ij}^\beta = \sum_\alpha \left( \overline{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) W_\alpha(\mathbf{x}_P) - \hat{W}_{ij}^\alpha(\mathbf{x}_P) r_{\alpha\beta} \right) \tag{10}$$

where

$$\hat{W}_{ij}^\alpha(\mathbf{x}) = \frac{\partial^2 W_\alpha(\mathbf{x})}{\partial x_i \partial x_j} \tag{11}$$

are projection operators obtained by differentiating the operators $W_\alpha$. Thus, six independent projection operators for the six independent spatial derivatives are required at each grid point in this scheme.

## 3.2 Improved scheme using finite differences

Another idea is to approximate the derivatives in $\widetilde{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta)$ using finite differencing on the grid, i.e. we write symbolically

$$\widetilde{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) = D_{ij} r_{\alpha\beta} \tag{12}$$

where $D_{ij}$ is a finite difference operator. Thus

$$M_{ij}^\beta = \sum_\alpha \left( \overline{G}_{ij}(\mathbf{x}_\alpha, \mathbf{x}_\beta) - D_{ij} r_{\alpha\beta} \right) W_\alpha(\mathbf{x}_P) \tag{13}$$

Note that a single projection operator is required per grid point. Moreover, since $D_{ij}$ is a linear operator, we may write

$$\sum_\alpha D_{ij} r_{\alpha\beta} W_\alpha(\mathbf{x}_P) = D_{ij} \sum_\alpha r_{\alpha\beta} W_\alpha(\mathbf{x}_P) \tag{14}$$

to compute these sums very efficiently.

## 4 NUMERICAL STUDY AND CONCLUDING REMARKS

To compare the different projection schemes (7), (10) and (13) we have performed numerical tests using a fictitious two-dimensional problem (shown in figure 2). The problem is to obtain the x-component of the "velocity" at the point Q ($x_Q$,3) due to unit traction applied at P along the x-direction and zero traction applied along the y-direction. Figures 3, 4 and 5 show the percentage relative error for different positions of the point Q when the different projection schemes (7), (10) and (13), respectively, are used. It is clear that the scheme in (13) results in a more accurate computation of the velocity than the scheme in equation (10). The accuracy of the former scheme is comparable to that of the original scheme in equation (7) and is therefore recommended.

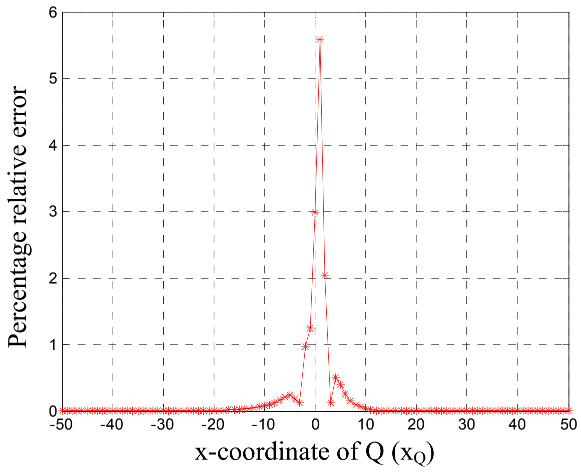

Figure 3: This figure shows the percentage relative error when the x-component of the velocity at Q ($x_Q$,3) is computed corresponding to unit force at P (0.5,0.5) in the x-direction using the projection scheme in equation (7). The interpolation stencil is locked in position relative to Q.

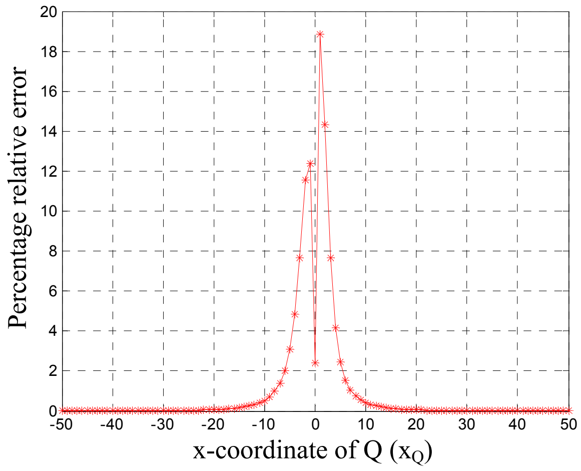

Figure 4: The percentage relative error for the same problem in Figure 2 when a different projection operator is used for computing the second derivative with respect to x as in equation (10).

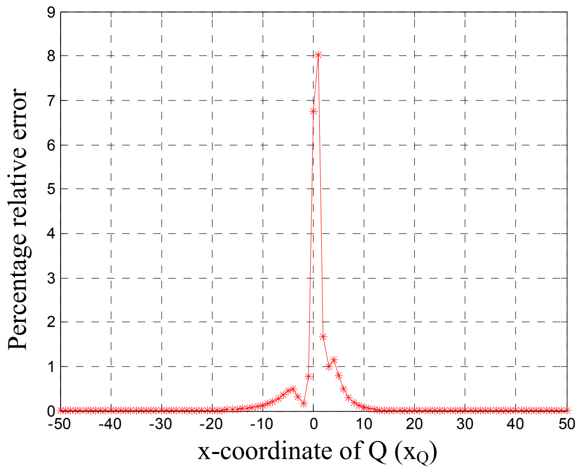

Figure 5: A much more accurate approximation is obtained when the second derivative operator is discretized using a finite difference approximation at every grid point as in (13).

## 5 ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the National Science Foundation as well as the Singapore-MIT Alliance program and thank B. Buchmann and J. Phillips for providing the precorrected FFT code for use in this project.

## REFERENCES

[1] N. R. Aluru and J. K. White, "A fast integral equation technique for the analysis of microflow sensors based on drag force calculations", *Proc. of MSM*, pp. 283-286, 1998.

[2] C. Pozrikidis, "Boundary integral and singularity methods for linearized viscous flow", *Cambridge University Press, New York,* 1992.

[3] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures", *IEEE Trans. on Computer-Aided Design*, Vol. 16, No. 10, pp. 1059-1072, 1997.

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