Journal of Modeling and Simulation of Microsystems, Vol. 1, No. 1, Pages 77-82, 1999.

FAST ALGORITHMS FOR 3-D SIMULATION

D. Ramaswamy, W. Ye, X. Wang and J. White[†]
Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA. 02139

Abstract

One difficulty associated with computer simulation of micromachined devices is that the devices are typically geometrically complicated and innately three-dimensional. For this reason, attempts to exploit existing finite-element based tools for micromachined device simulation has proved difficult. Instead, micromachine device designers have been early adopters of the recently developed fast solvers for integral equations. In this paper the author will describe a little of the history of these methods, primarily to point the interested reader to the relevant literature.

Keywords: integral equations, fast solvers

1. Introduction

Simulating a micromachined device, like the electromechanical resonator in Fig. 1, is extremely computationally challenging. One issue that makes simulation difficult is that the resonator's behavior is governed by the coupling between electrostatic, elastic and fluidic forces. The second issue is the computational expense of calculating the domain-specific forces in such a geometrically complicated example. Forces that require resolution of the geometry's exterior, such as electrostatic or fluidic forces, are particularly expensive to compute.

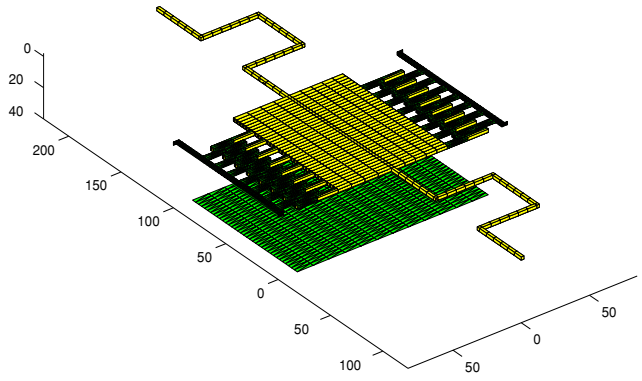

Fig. 1: A Comb Resonator Example.

When the equations that describe the exterior problem are linear and space invariant, as is typically the case for electrostatic and magnetic forces and can be the case for fluidic forces, an integral formulation of the problem will exist. Such formulations use Greens functions to eliminate the problem's exterior and typically involve only surface quantities. Such a formulation seems ideal when computing traction forces or electrostatic pressures on surfaces, but the integral formulation generates a particular numerical difficulty. Discretized integral equations generate dense matrices which are expensive to form and solve.

In the past decade, fast techniques have been developed for solving the systems of equations generated by discretized integral equations. One of the earliest practical programs using these fast techniques was developed to compute 3-D capacitances and electrostatic forces [8, 9]. Recent extensions have appeared, such as programs for computing inductance [11] or fluid drag [12], as well as algorithm improvements such as better adaptivity, higher-order elements and improved efficiency for high accuracy [13, 10].

In this brief paper we will try to make clear what is being exploited to develop these fast algorithms. We will try to present the differences in various approaches, but we will not present any complete algorithms. Instead, we will refer the reader to the relevant literature. In the follwing section we describe the basic approach to developing fast algorithms for solving integral equations, and show that the key computation is forming matrix-vector products. In sections 3 and 4, we describe two classes of methods for fast matrix-vector computation, one based on multiresolution and the other based on diagonalization. Finally, in section 5, we show some computation results to demonstrate the effectiveness of these methods.

2. The Fast Solver Approach

As an example, consider the first-kind integral equation problem,

$$\psi(x) = \int_{surface} G(x-x')\sigma(x')da', \qquad (1)$$

where x is a point on the surface, and $\psi(x)$ is the a known surface potential, and where $G(x-x')$ is the space invariant Greens function. For example, in electrostatics

†. white@mit.edu

$G(x-x') = \frac{1}{4\pi\varepsilon\|x-x'\|}$.

The simplest discretization of (1) is to divide the surfaces into n flat panels over which the charge density is assumed constant. Then, the charges are determined by insisting (1) be exactly satisfied at a collection of collocation points. Typically, the collocation points, denoted by x_i's, are selected to be the centroids of each panel. The resulting discretized system is

$$Pq = \Psi \tag{2}$$

where q is the n-length vector of panel charges, Ψ is the n-length vector of known centroid potentials, and $P_{i,j} = \int_{panel_j} G(x_i - x')da'$.

If direct factorization is used to solve (2), then the memory required to store the matrix will grow like n^2 and the matrix solve time will increase like n^3. If instead, a preconditioned Krylov-subspace method like GMRES [1] is used to solve (2), then it is possible to reduce the solve time to order n^2 but the memory requirement will not decrease.

In order to develop algorithms that use memory and time that grows more slowly with problem size, it is essential *not* to form the matrix explicitly. Instead, one can exploit the fact that Krylov-subspace methods for solving systems of equations only require matrix-vector products and not an explicit representation of the matrix. For example, note that for P in (2), computing Pq is equivalent to computing n potentials due to n sources and this can be accomplished in nearly order n operations [2-4]. Several researcher simultaneously observed the powerful combination of discretized integral equations, Krylov-subspace methods, and fast matrix-vector products [5-7]. Such methods are now referred to, somewhat pejoratively, as fast solvers.

3. Fast Matrix-Vector Products using Multiresolution

Multiresolution methods for rapidly computing matrix-vector products can best be understood by considering the simple example of determining potentials at M collocation points due to M charged panels, as shown in Fig. 2. The matrix $\tilde{P}$, which relates the panel charges to the collocation point potentials, is $M \times M$ and dense. Therefore, determining the M potentials requires M^2 operations.

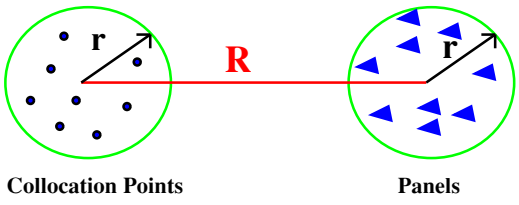

Fig. 2: A Cluster of collocation points seperated from a cluster of panels.

If the Greens function, $G(x-x')$, is the free-space electrostatics Greens function, $\frac{1}{\|x-x'\|}$, then the $\tilde{P}$ matrix can be well approximated by a low rank matrix. The easiest way to show this is to consider the singular value decomposition of $\tilde{P}$,

$$\tilde{P} = U\Sigma V^T \tag{3}$$

where U and V are orthonormal matrices and Σ is the diagonal matrix of singular values of $\tilde{P}$. In Fig. 3, the largest 20 singular values of the $\tilde{P}$ are plotted for various separation distances. For this example, 100 randomly placed collocation points and panels were used.

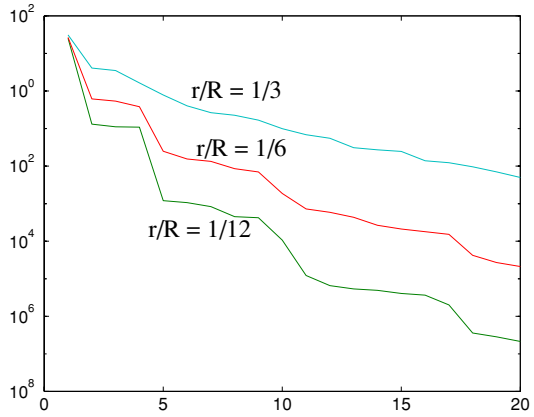

Fig. 3: The Singular Values of the Matrix Relating the Cluster of Panels to the Cluster of Collocation Points for different seperation distances. r and R are as defined in Fig. 2.

Note that in the figure, the singular values of $\tilde{P}$ are decreasing very rapidly, particularly when the clusters are further apart. This makes it clear that for an approximate representation of $\tilde{P}$ not all the singular vectors are needed. Instead, one could represent $\tilde{P}$ approximately using only q singular values as

$$\tilde{P} \approx U(1{:}q)\Sigma(1{:}q)V(1{:}q)^T \tag{4}$$

where $U(1{:}q)$, $V(1{:}q)$ are the $M \times q$ matrices which are the first q columns of U and V respectively and $\Sigma(1{:}q)$ is the $q \times q$ diagonal submatrix of Σ. For example, one can see from the graph in Fig. 3 that if one part in one thousand accuracy is required, q should be larger than 15 if the cluster radius is one third of the separation distance, but q can be less than 5 if the cluster radius is one twelfth of the separation distance. Very effective general algorithms have been developed using the singular-value decomposition in a hierarchical fashion [16].

One of the difficulties in using the SVD approach is that the dense interaction matrix must first be computed before the low rank approximation can be formed. Heuristics can be used to avoid forming the matrix completely, but such approaches correspond roughly to a second technique diagrammed in Fig. 4.

In this second approach, the charged panels are represented by projecting them onto the grid of q point charges on the right-hand side of Fig. 4, and the collocation point potentials are interpolated from q potentials from the left-hand side grid points. The interaction between panels and collocation points is then represented by an interaction between the right- and

left-hand grid points. This grid-based approach corresponds to representing the $\tilde{P}$ matrix in the form

$$\tilde{P} \approx Interp(q)G2G(q)Proj(q)^T \tag{5}$$

where q is the number of grid points, $Proj(q)$ and $Interp(q)$ are the $M \times q$ projection and interpolation matrices, and $G2G(q)$ is the $q \times q$ matrix representing computing the left-hand grid potentials from the right-hand grid of charges.

Fig. 4: The Singular Values of the Matrix Relating the Cluster of Panels to the Cluster of Collocation Points for different seperation distances.

There are many ways to project onto the grid, including polynomial interpolation [15], moment matching [25], and point matching [23]. Also, rather than surrounding the panels and collocation points with each grid, one could shrink the grids to single points in the center of each cluster. This limiting process will, in effect, represent the panels and collocation points with higher derivatives at a single point rather than evaluations at multiple points. Using these derivative representations lead to what are referred to as fast multipole algorithms [3].

As a final note, compare (4) to (5). These equations make it clear that the $V(1{:}q)$ singular vectors effectively project the panels onto a reduced representation, and the $U(1{:}q)$ singular vectors effectively interpolate the potentials from that representation.

4. Fast Matrix-Vector Products Using Diagonalization

In both the SVD and grid-based approaches described above, the cost of computing M potentials from M charges is reduced from order M^2 to order Mq. Therefore, these methods are efficient only if q is much smaller than M. The plot of the singular values of $\tilde{P}$ for the $\frac{1}{\|x-x'\|}$ Greens function in Fig. 3 makes it clear that q will be very small indeed. This pleasing result is not true for all Greens functions.

For problems in acoustics or full-wave electromagnetics, the Greens function is the Helmholtz Greens function

$$G(x-x') = \frac{e^{j\omega\|x-x'\|}}{\|x-x'\|} \tag{6}$$

where $j = \sqrt{-1}$ and ω is the frequency of interest. For the Helmholtz Greens function, reconsider the $\tilde{P}$ matrix, which again relates the M panel charges to the M collocation point potentials in the clusters example from Fig. 2. In Fig. 5, the largest 20 singular values of the $\tilde{P}$ computed using the Helmholtz Greens function are plotted for a fixed separation distance, $\frac{r}{R} = 1/3$, and for $\omega = 0$, $\omega = 1$ and $\omega = 4$.

As is clear from Fig. 5, for the $\omega = 4$ case the singular values of $\tilde{P}$ are not decreasing quickly. A good approximation to $\tilde{P}$ in this case would involve using almost all the singular vectors, implying q would be nearly equal to M and the cost of multiplying by $\tilde{P}$ would still be order M^2. Note in the graph that the $\omega = 0$ case corresponds to the electrostatic Greens function, and its singular values are dropping rapidly.

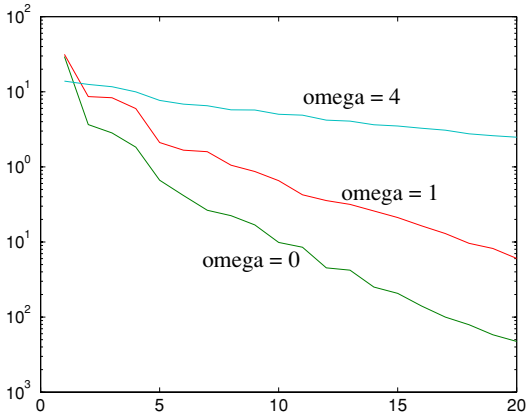

Fig. 5: The Singular Values of the Matrix Relating the Cluster of Panels to the Cluster of Collocation Points for different frequencies.

It is still possible to improve the efficiency of the matrix-vector product in the Helmholtz case, but not by using the SVD. Instead, consider again the grid projection strategy diagrammed in Fig. 4. For the Helmholtz case, it is likely that the number of grid points required will be order M, so grid projection seems to have the same problem as using the SVD. However, if the grid points are laid out uniformly then computing the M grid potentials from M grid charges can be performed in $M \log M$ operations using the Fast Fourier transform (FFT) [25].

Just using the FFT to compute grid potentials from grid charges is insufficient to reduce the cost of multiplying by $\tilde{P}$ below order M^2. To see this, examine the representation of $\tilde{P}$,

$$\tilde{P} \approx Interp(M)G2G_{FFT}(M)Proj(M)^T \tag{7}$$

where $G2G_{FFT}$ is used to denote that the grid potentials are computed from the grid charges using the FFT. The interpolation and projection operations, which project M panels onto M point charges and interpolate M grid potentials on to M collocation points respectively, each require M^2 operations. However, if the potential due to a cluster of panels will be evaluated for many clusters of collocation points, the cost of projection can be amortized over many $\tilde{P}$'s. In addition, there is a dual savings for interpolation.

Using the FFT makes the grid-to-grid operation "more diagonal", and exploiting this diagonalization is at the heart of specialized algorithms for solving high frequency Helmholtz problems [20-22].

5. A Few Comments and Results

Many of the mature fast solvers are for $\frac{1}{r}$ kernels and use multipole expansions. More recently there has been interest in developing techniques which are Greens function independent, and a variety of approaches have appeared. There is the panel clustering idea [6], a multigrid style method [15], a technique based on the singular-value decomposition [16], and approaches based on using wavelet-like methods [17-19].

For problems with oscillatory kernels, such as acoustic or electromagnetic scattering, there are specialized multipole algorithms [20-22], but these techniques collapse numerically at low frequencies. The techniques that are effective for general kernels are based on projecting to a global uniform grid and using the FFT [4,23,24]. Global FFT-based techniques, unfortunately, have efficiency problems for inhomogenous geometries.

The Greens function generality of fast solver algorithms based on using the FFT globally have made such techniques the method of choice for many applications. In the subsections below, we describe a few results using such a fast solver approach.

5.1. Stokes Flow Analysis of a Comb

In order to determine the quality factor of a comb-drive resonator packaged in air, it is necessary to determine the drag force on the comb. The small spatial scale of micromachined combs implies that flow in these devices typically have very low Reynolds numbers, and therefore convection can often be ignored. In addition, fluid compression can be ignored for devices which use lateral actuation, like many of the comb-drive based structures fabricated using micromachined polysilicon. The result of these two simplifications is that fluid damping forces on laterally actuated microdevices can be accurately analyzed by solving the incompressible Stokes equation, rather than by solving the compressible Navier-Stokes equation.

The simplification to the Stokes equation certainly makes analysis of fluid damping in microdevices much more computationally tractable. However, analysis of an entire comb in a reasonable period of time only recently became possible due to the extension of an accelerated boundary-element method [23] to computing Stokes flow solutions[12, 26].

As an example, consider the comb in Fig. 6. The number of unknowns in the system exceeds 50,000, and yet the accelerated Stoke's flow solver finished in under 20 minutes. If direct methods were used instead, the simulation would have taken weeks.

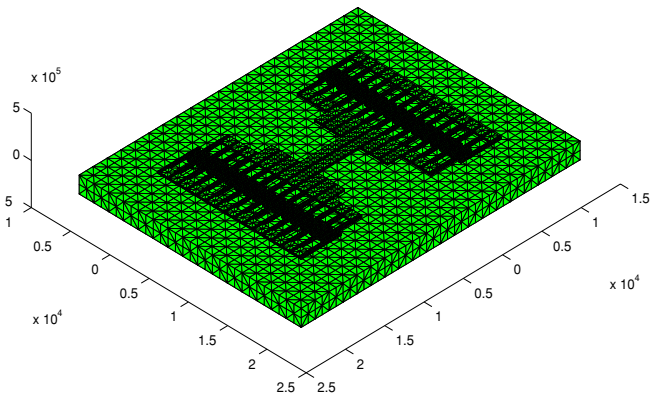

Fig. 6: A discretized comb drive resonator over a substrate.

5.2. Coupled Electromechanical Analysis of Comb Levitation

In order to determine levitation in a Comb drive, it is necessary to solve a coupled electromechanical problem. In this subsection we present results from our multilevel-Newton coupled electromechanical code. The program uses the precorrected-FFT accelerated integral equation solver with planar triangular panels to compute the electrostatic forces. A finite-element, mixed rigid/elastic mechanical analysis program using 20 noded isoparametric brick elements is used to compute displacments. The multilevel-Newton method uses pressure sensitivites to improve efficiencies [27].

An 18 finger PolySi resonator (YM = 150 GPa, PR = 0.3) Fig. 7 is supended with 400 μm beams with a uniform depth of 1.94 μm and finger dimensions 13.8×4.6 μm. The movable structure and the ground electrode are kept at 0 V and a non zero voltage is applied to the driving electrodes which interdigitate with the movable fingers at the sides.

The effect of varying the width of the suspension beam was investigated Fig. 8. Each load step in the simulation took about 70 minutes of CPU time (Sun Ultra 30). Without the fast electrostatic solver, the simulation would have lasted for weeks.

6. Conclusions

In this brief paper the authors described some of the history of fast methods for solving integral equations. The authors would like to thank the many students who have developed codes using fast solvers including Keith Nabors, Joel Phillips, Matt Kamon, Michael Chou, Narayan Aluru, and Joe Kanapka. This work was supported by the DARPA composite CAD program, the DARPA muri program, and grants from the Semiconductor Research Corporation.

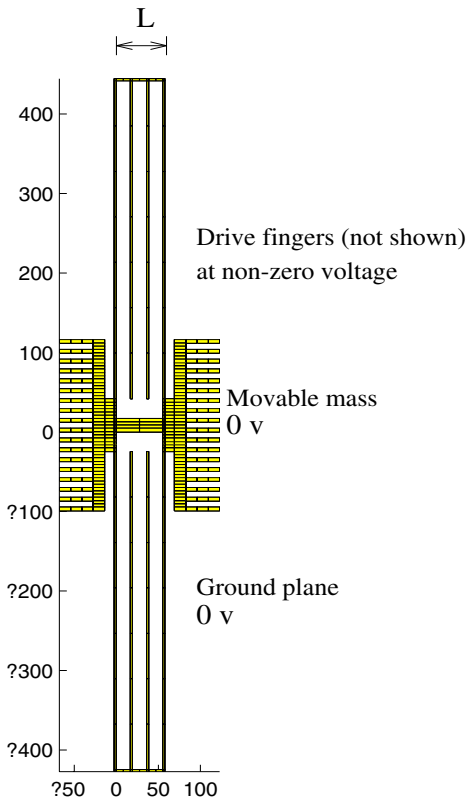

Fig. 7: Comb drive resonator.

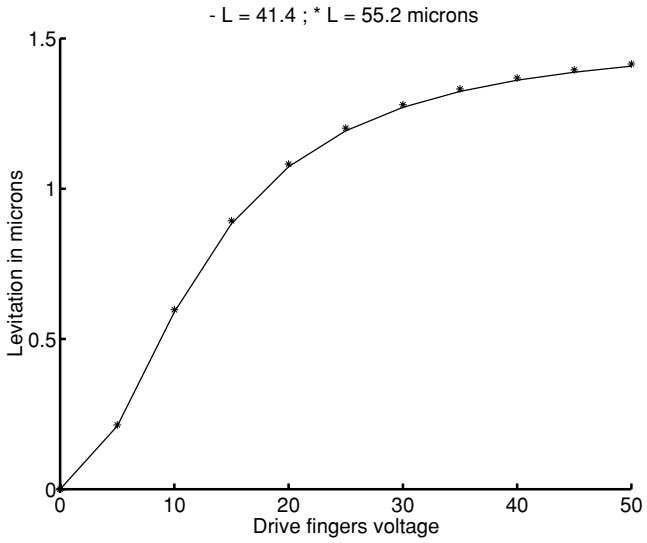

Fig. 8: Levitation.

References

[1] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM J. Sci. Statist. Comput., 7(3):105-126, 1986.

[2] J. Barnes and P. Hut. A hierarchical $O(N\log N)$ force-calculation algorithm. Nature, 324:446-449, 1986.

[3] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325-348, 1987.

[4] R. W. Hockney and J. W. Eastwood, *Computer simulation using particles*. New York: Adam Hilger, 1988.

[5] V. Rokhlin, "Rapid solution of integral equation of classical potential theory," *J. Comput. Phys.*, vol. 60, pp. 187-207, 1985.

[6] W. Hackbusch and Z. P. Nowak, "On the Fast Matrix Multiplication in the Boundary Element Method by Panel Clustering," Numer. Math. 54, pp. 463-491, 1989.

[7] D. T. Borup and O. P. Gandhi, "Calculation of high-resolution SAR distributions in biological bodies using the FFT algorithm and conjugate gradient method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 33, pp. 417-419, 1985.

[8] K. Nabors, J. White, "A Fast Multipole Algorithm for Capacitance Extraction of Complex 3-D Geometries" *Proc. Custom Int. Circuits Conf.*, San Diego, California, May 1989, p21.7.1-21.7.4.

[9] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-D capacitance extraction program," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, pp. 1447-1459, November 1991.

[10] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM J. Sci. Statist. Comput.*, 15(3):713-735, 1994.

[11] M. Kamon, M. J. Tsuk, and J. White, "FastHenry, A Multipole-Accelerated 3-D Inductance Extraction Program," *Proceedings of the 30th Design Automation Conference*, Dallas, June 1993.

[12] N. R. Aluru and J. White, "A Fast Integral Equation Technique for Analysis of Microflow Sensors Based on Drag Force Calculations," International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, Santa Clara, April 1998, pp. 283-286.

[13] M. Bachtold, J.G. Korvink, H. Baltes, "The Adaptive, Multipole-Accelerated BEM for the Computation of Electrostatic Forces," Proc. CAD for MEMS, Zurich, 1997, pp. 14.

[14] L. Greengard, V. Rokhlin, "A New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions," Acta Numerica, 1997, pp. 229-269.

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

[16] S. Kapur and J. Zhao,"A fast method of moments solver for efficient parameter extraction of MCMs" Design Automation Conference, 1997 pp. 141-146.

[17] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Comm. Pure Appl. Math.*, XLIV:141-183, 1991.

[18] W. Shi, J. Liu, N. Kakani, and T. Yu, "A Fast Hierarchical Algorithm for 3-D Capacitance Extraction" *Proceeding of the 29th Design Automation Conference*, San Francisco, CA, June, 1997, pp. 212-217.

[19] J. Tausch and J. White "Precondition and Fast Summation Techniques for First-Kind Boundary Integral Equations" Third IMACS International Symposion on Iterative Methods in Scientific Computation, Jackson Hole WY, Jul 9-12, 1997

[20] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comp. Phys.*, 86:414-439, 1990.

[21] C.C. Lu and W. C. Chew. Fast algorithms for solving hybrid integral equations. *IEEE Proceedings-H*, 140(6):455-460, December 1993.

[22] V. Rokhlin. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Applied and Computational Harmonic Analysis*, 1:82-93, 1993.

[23] J. R. Phillips and J. K. White, "Efficient capacitance extraction of 3D structures using generalized pre-corrected FFT methods," in Proceedings IEEE 3rd topical meeting on electrical performance of electronic packaging, November 1994.

[24] J. R. Phillips, E. Chiprout, and D. D. Ling, "Efficient full-wave electromagnetic analysis via model-order reduction of fast integral transforms," Proceedings of the 33rd Design Automation Conference, Las Vegas, NV, June 1996.

[25] L. Berman, "Grid-multipole calculations", IBM Research Report, 1993, no. RC 19068(83210).

[26] W. Ye, J. Kanapka, X. Wang and J. White, "Efficiency and Accuracy Improvements for FastStokes, A Precorrected-FFT Accelerated 3-D Stokes Solver," To appear International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Juan, April 1999

[27] D. Ramaswamy, N. R. Aluru and J. White, "A Mixed Rigid/Elastic Formulation for Efficient Analysis of Electromechanical Systems," International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, Santa Clara, April 1998, pp. 304-307.

Received in Cambridge, MA, USA, 20th July 1999

Paper 1/01765