Fast Multipole Algorithms for Solving Three-Dimensional Potential Integral Equations," *Proceedings of the Colorado Conference on Iterative Methods*, Breckenridge, Colorado, April 1994

COMPARING PRECORRECTED-FFT AND FAST MULTIPOLE ALGORITHMS FOR SOLVING THREE-DIMENSIONAL POTENTIAL INTEGRAL EQUATIONS

J. WHITE*, J. R. PHILLIPS† AND T. KORSMEYER‡

1. Introduction. Mixed first- and second-kind surface integral equations with $\frac{1}{r}$ and $\frac{\partial}{\partial n}\frac{1}{r}$ kernels are generated by a variety of three-dimensional engineering problems. For such problems, Nyström type algorithms can not be used directly, but an expansion for the unknown, rather than for the entire integrand, can be assumed and the product of the singular kernal and the unknown integrated analytically. Combining such an approach with a Galerkin or collocation scheme for computing the expansion coefficients is a general approach, but generates dense matrix problems. Recently developed fast algorithms for solving these dense matrix problems have been based on multipole-accelerated iterative methods [1, 2, 3], in which the fast multipole algorithm is used to rapidly compute the matrix-vector products in a Krylov-subspace based iterative method. Another approach to rapidly computing the dense matrix-vector products associated with discretized integral equations follows more along the lines of a multigrid algorithm [4], and involves projecting the surface unknowns onto a regular grid, then computing using the grid, and finally interpolating the results from the regular grid back to the surfaces.

In this paper, we describe a precorrected-FFT approach to replacing the fast multipole algorithm for accelerating the dense matrix-vector product associated with discretized potential integral equations. The precorrected-FFT method, sketchily described below, is an order *nlogn* algorithm, and is asymptotically slower than the order n fast multipole algorithm. However, initial experimental results indicate the method may have significant constant factor advantages for a variety of engineering problems.

2. Problem Formulation. For exposition in this short paper, we consider only a simplified discretization applied to a first kind formulation, though the techniques generalize. The approach used is to formulate the exterior Dirichlet problem using a single layer charge density denoted σ. It then follows that σ must satisfy the integral equation

$$\psi(x) = \int_{surfaces} \sigma(x') \frac{1}{\|x - x'\|} da', \qquad x \in surfaces. \tag{1}$$

where $\psi(x)$ is the known surface potential, da' is the incremental conductor surface area, x, $x' \in \mathbf{R}^3$, and $\|x\|$ is the usual Euclidean length of x given by $\sqrt{x_1^2 + x_2^2 + x_3^2}$.

* Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. (white@rle-vlsi.mit.edu)
† Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology Cambridge, MA 02139. (jphill@rle-vlsi.mit.edu)
‡ Dept. of Ocean Engineering Massachusetts Institute of Technology Cambridge, MA 02139. (xmeyer@f-c.mit.edu)

A standard approach to numerically solving (1) for σ is to use a piece-wise constant Galerkin scheme. That is, the surfaces are discretized into n small panels or tiles, and it is assumed that on each panel i, a charge, q_i, is uniformly distributed. Then for each panel, an equation is written which relates the known potential at the center of that i-th panel, denoted $\overline{p}_i$, to the sum of the contributions to that potential from the n charge distributions on all n panels. The result is a dense linear system,

$$Pq = \overline{p} \tag{2}$$

where $P \in \mathbf{R}^{n \times n}$, q is the vector of panel charges, $\overline{p} \in \mathbf{R}^n$ is the vector of known panel potentials, and

$$P_{ij} = \frac{1}{a_i a_j} \int_{panel_i} \int_{panel_j} \frac{1}{4\pi\epsilon_0 \|x - x'\|} \, dada', \tag{3}$$

where a_i and a_j are the areas of the i-th and j-th panel.

The dense linear system of (2) can be solved to compute panel charges from a given set of panel potentials, and the electrostatic capacitances can be derived by summing the panel charges. If Gaussian elimination is used to solve (2), the number of operations is order n^3. Clearly, this approach becomes computationally intractable if the number of panels exceeds several hundred. Instead, consider solving the linear system (2) using a Krylov-subspace style iterative method. The dominant costs in such an algorithm will be calculating the n^2 entries of P using (3) before the iterations begin, and performing n^2 operations to compute Pq on each iteration. Described below is a precorrected-FFT based algorithm which, through the use of approximate grid projections, avoids forming most of P and reduces the cost of forming Pq to order $nlogn$ operations.

3. The precorrected FFT method. After a three dimensional problem has been discretized into panels, consider then subdividing the cube containing the problem into an $n \times n \times n$ array of small cubes so that each small cube contains only a few panels. Several sparsification techniques for P are based on the idea of directly computing only those portions of Pq associated with interactions between panels in neighboring cubes. The rest of Pq is then somehow approximated to accelerate the computation [7].

One approach to computing distant interactions is to exploit the fact that evaluation points distant from a cube can be accurately computed by representing the given cube's charge distribution using a small number of weighted point charges. In particular, if the point charges all lie on a uniform grid, then FFT can be used to compute the potential at these grid point due to the grid charges. More specifically, one method for approximating Pq in $nlogn$ operations has four steps:

- directly compute nearby interactions,
- project the panel charges onto a uniform grid of point charges,
- compute the grid potentials due to grid charges using an FFT, and
- interpolate the grid potentials onto the panels.

The difficulty, as will be made clearer below, is that the calculations using the FFT on the grid are not only approximate, but also nearly duplicate the calculation of the nearby

interactions. This can be precorrected, at almost no cost, by modifying the way the direct interaction is computed.

3.1. Projecting onto a grid. For panel charges contained within a given cube, the potentials at evaluation points distant from the given cube can be accurately computed by representing the given cube's charge distribution with a small number of appropriately weighted point charges on a uniform grid thoughout the cube's volume. For example, consider the cube embedded in the center of a $3 \times 3 \times 3$ array of cubes, and assume that the potential will be evaluated at points exterior to the 27 cube array. Then, since the potential satisfies Laplace's equation, the error in the point charge approximation over the entire exterior can be minimized by minimizing the potential error on the surface of the cube array.

The above observation suggests a scheme for computing the grid charges used to represent charge in a given cube a. First, test points are selected on the surface of the $3 \times 3 \times 3$ cube array which has cube a as its center. Then, potentials due to the grid charges are forced to match the potential due to the cube's actual charge distribution at the test points. Since such collocation equations are linear in the charge distribution, this projection operation which generates a subset of the grid charges, q_a^g, can be represented as a matrix, W_a, operating on a vector representing the panel charges in cube a, q_a. In particular, if there are G grid charges and A panels, then

$$q_g = W_a q_a = \begin{bmatrix} P^{gt} \\ 1 \ldots 1 \end{bmatrix}^{-1} \begin{bmatrix} P^{qt} \\ 1 \ldots 1 \end{bmatrix} q_a, \tag{4}$$

where $P^{gt} \in \Re^{(G-1)\times G}$ and

$$P^{gt}_{i,j} = \frac{1}{\|x_i^t - x_j^g\|}; \tag{5}$$

$P^{qt} \in \Re^{(G-1)\times A}$ and

$$P^{qt}_{i,j} = \int_{panel\ j} \sigma(x') \frac{1}{\|x_i^t - x'\|} da'. \tag{6}$$

Here x_i^t and x_j^g are the position of the i-th test point and the j-th grid point. The rows of ones in (4) insure that the sum of grid charges is equal to the net charge in the cube. For an alternative approach based more generally on matching multipole expansion coefficients, see [6].

3.2. Using the FFT. For a general three dimensional problem, consider subdividing a cube containing the entire problem domain into a $n \times n \times n$ array of small cubes. Then, the collocation approach above can be used to generate point charge approximations for charge distributions in every cube, effectively projecting the charge density onto a three-dimensional grid. For example, if the representative point charges are placed at the cube vertices, then the resulting charge distribution will be projected to a $(n+1) \times (n+1) \times (n+1)$ uniform grid. Fast multipole algorithms also effectively create a uniform grid by

constructing multipole expansions at the center of each cube, but due to sharing, the point charge approach can be more efficient. For example, a point charge at a cube vertex is used to represent charge in the eight cubes which share that vertex.

Once the charge has been projected to a grid, computing the potentials at the grid points due to the grid charges is a three-dimensional convolution. We denote this as

$$\psi_g(i,j,k) = \sum_{i',j',k'} h(i-i', j-j', k-k') q_g(i',j',k'). \tag{7}$$

where i,j,k and i',j',k' are triplets specifying the grid points, ψ_g is the vector of grid point potentials, q_g is the vector of grid point charges, and $h(i-i', j-j', k-k')$ is the inverse distant between grid points i,j,k and i',j',k'. As will be made clear below, $h(0,0,0)$ can be arbitrarily defined, and is set to zero. The above convolution can then be computed in *nlogn* time using the Fast Fourier Transform.

Once the grid potentials have been computed, they can be interpolated to the panels in each cube using the transpose of W_a [4]. Therefore, projection, followed by convolution, followed by interpolation, can be represented as

$$\psi_{fft} = W^t \hat{H} W q, \tag{8}$$

where q is the vector of panel charges, ψ_{fft} is an approximation to the panel potentials, W is the concatenation of the $W_a's$ for each cube, and $\hat{H}$ is the Toeplitz matrix representing the convolution in (7).

3.3. Precorrecting. In ψ_{fft} of (8), the portions of Pq associated with neighboring cube interactions have already been computed, though this close interaction has been badly approximated in the projection/interpolation. Before computing a better approximation, it is necessary to remove the contribution of the inaccurate approximation. In particular, denote as $P_{a,b}$ the portion of P associated with the interaction between neighboring cubes a and b, denote the potential at grid points in cube a due to grid charges in cube b as $\hat{H}_{a,b}$, and denote ψ_a and q_b as the panel potentials and charges in cubes a and b respectively. Then

$$\psi_a = \psi_{a_{fft}} + \left(P_{a,b} - W_a^t \hat{H}_{a,b} W_b^t)\right) q_b \tag{9}$$

will be a much better approximation to ψ_a.

Assuming that the Pq product will be computed many times in the inner loop of an iterative algorithm,

$$P_{a,b}^{cor} = \left(P_{a,b} - W_a^t \hat{H}_{a,b} W_b^t)\right) \tag{10}$$

will be expensive to initially compute, but will cost no more to subsequently apply than $P_{a,b}$.

4. Conclusions. Using the above notation, the precorrected FFT algorithm for computing Pq can be briefly described as two steps. First compute

$$\psi_{fft} = W^t H W q \tag{11}$$

using the FFT to diagonalize H. Then, for each cube a, compute

$$\psi_a = \sum_{b \in neighbors} P^{cor}_{a,b} q_b + \psi_{a_{fft}}; \tag{12}$$

Numerical experiments indicate that using a uniform grid with an average of eight grid points associated with each cube (making $G = 27$ in (4)) results in potentials outside nearest neighbors which have errors similar to those produced by evaluating second-order multipole expansions (which have nine coefficients) outside second nearest neighbors. Assuming a homogenous distribution of eight panels per cube, this implies that for commensurate accuracy, the precorrected FFT method has a complexity of $(216 + K_{fft} log n) * n$ and the fast multipole algorithm has a complexity of at least $2700n$. So, although the fast multipole algorithm is asymptotically faster, it is only of practical significance for extremely large n.

It should be noted that the above result is NOT general. The fast multipole algorithm retains its linear-time behavior even in the arbitrarily inhomogenous case [5]. The precorrected FFT method is only suitable when the distribution can be made to look homogenous. More detailed experiments are required to better understand these pragmatic issues.

REFERENCES

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

[2] A. Greenbaum, L. Greengard, and G. B. McFadden, "Laplace's equation and the dirchlet-neumann map in multiply connected domains," tech. rep., Courant Institute, New York, March 1991.

[3] 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.

[4] A. Brandt, "Multilevel Computations of Integral Transforms and Particle Interactions with Oscillatory Kernals," *Computer Physics Communications*, no. 65, pp. 24-38, 1991.

[5] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, "Multipole Accelerated Preconditioned Iterative Methods for Three-Dimensional Potential Integral Equations of the First Kind," *To Appear, SIAM J. on Sci. and Stat. Comp.*

[6] C. L. Berman, "Grid-Multipole Calculations," *IBM Research Report*, no. RC 19068(83210), 1993.

[7] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems.* Cambridge, Massachusetts: M.I.T. Press, 1988.