11th Int'l. Workshop on Water Waves and Floating Bodies, Hamburg, 7/96

A Precorrected-FFT Algorithm for Accelerating Surface Wave Problems *

Tom Korsmeyer † Joel Phillips ‡ Jacob White §

1 Introduction

In analyzing the hydrodynamic interaction of a body and free-surface waves, boundary integral equation methods (BIEM) are currently more popular than volume methods like finite-element (FEM) because: (1) these are exterior problems, but BIEM with a suitable Green function often allows the computational surface to be the body surface alone; and (2) the complexity of the body geometry and the time dependence of the free-surface geometry may make surface grids easier to generate than volume grids.

Classical BIEM are computationally expensive because they generate dense linear systems, whereas FEM generate larger sparse systems. Using iterative methods, the computational cost (time expended and memory allocated) of solving an $N \times N$ dense system is at least order N^2, but can be as low as order N for an $N \times N$ sparse system. So in three dimensions, if n unknowns are required *per dimension*, the cost of BIEM will be n^4 and the cost of a volume method as low as n^3. Recently, there have been advances in the acceleration of BIEM which amount to sparsification techniques for the dense systems. The most widely implemented of these is multipole acceleration (*MA*), which has been shown to reduce the computational cost to order $(n^2 = N)$ [1, 2].

The MA algorithms have been instrumental in a significant shortening of the duration of the design cycle for many problems in low frequency electromagnetics, where the Green function is the Rankine free-space Green function, there are only piecewise-constant Dirichlet boundary conditions and engineers routinely contemplate problems with order 10^5 unknowns. However the MA algorithms have been slow to catch on in the hydrodynamics community where complicated Green functions may be used to satisfy free-surface and periodic boundary conditions, the need for knowing the fluid velocity as well as the potential has motivated the use of higher-order panels, and the Neumann/Dirichlet boundary intersections have motivated special treatments for some panels or nodes. While such problems may be treated with MA in principle, the implementation is daunting. Only in hydrodynamic problems cast as desingularized or vortex-dynamics formulations, using the fundamental free-space Green functions, are researchers benefiting from MA algorithms.

An algorithm which avoids some of these shortcomings is the precorrected-FFT method [3, 4]. In this abstract, the precorrected-FFT method will be described and indicators of its performance for hydrodynamic problems will be provided.

2 A Linearized Hydrodynamic Problem

Consider the familiar linearized frequency-domain radiation/diffraction problem for a body with surface S_b, upon which there is a unit normal vector $\hat{n}$, in the semi-infinite fluid domain V, under the free surface S_f. A potential $\psi(\vec{x};\omega)$ is to be found which satisfies the field equation

$$\Delta\psi(\vec{x};\omega) = 0 \qquad \vec{x} \in V, \tag{1}$$

*This work is supported by Advanced Research Projects Agency contract N00174-93-C-0035 and DABT63-94-C-0053, National Science Foundation contracts MIP-8858764 A02 and ECS-9301189, The Joint Industry Project *Wave Effects on Offshore Structures*, F.B.I. contract J-FBI-92-196, and grants from Digital Equipment Corporation and I.B.M.

†Dept. of Ocean Engineering, Massachusetts Institute of Technology (xmeyer@caf.mit.edu)

‡Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (jphilli@rle-vlsi.mit.edu)

§Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (white@rle-vlsi.mit.edu)

$\vec{x} = (x, y, z) \in \mathbf{R}^3$ with $z = 0$ the plane of S_f. $\psi(\vec{x}; \omega)$ must also satisfy the boundary conditions

$$(-\omega^2 + g\frac{\partial}{\partial z})\psi = 0 \qquad \vec{x} \in S_f, \quad \hat{n} \cdot \nabla\psi = f(\vec{x}) \qquad \vec{x} \in S_b, \tag{2}$$

and a radiation condition. This boundary-value problem may be recast as a boundary-integral equation to be solved on S_b by Green's theorem and the Green function

$$G^f(\vec{x}; \vec{\xi}, \omega) = G(\vec{x}; \vec{\xi}) + \tilde{G}(\vec{x}; \vec{\xi}, \omega) \tag{3}$$

with

$$G(\vec{x}; \vec{\xi}) = \frac{1}{r} \quad \text{and} \quad \tilde{G}(\vec{x}; \vec{\xi}, \omega) = \frac{1}{r'} + \frac{2\nu}{\pi}\int_0^\infty dk \frac{e^{k(z+\zeta)}}{k - \nu} J_o(kR) \tag{4}$$

in which r is the Euclidean distance between $\vec{x}$ and $\vec{\xi}$, r' is that distance between $\vec{x}$ and the image point ($\vec{\xi}$ reflected about S_f), $\nu = \frac{\omega^2}{g}$, R is the horizontal distance between $\vec{x}$ and $\vec{\xi}$, and J_o is the zeroth-order Bessel function. A collocation procedure combined with discretization of the body surface into N constant strength planar panels leads to the dense linear system

$$Dp = Pf \tag{5}$$

where $p, f \in \mathbf{R}^N$ are the vectors of unknown panel potentials and known panel Neumann boundary conditions respectively, and $D, P \in \mathbf{R}^{N \times N}$ are given by

$$D_{ij} = \frac{1}{a_j}\int_{panel_j} d\vec{\xi}\, \hat{n}_j \cdot \nabla G^f(\vec{x}_i; \vec{\xi}, \omega), \quad \text{and} \quad P_{ij} = \frac{1}{a_j}\int_{panel_j} d\vec{\xi}\, G^f(\vec{x}_i; \vec{\xi}, \omega) \tag{6}$$

where a_j and $\hat{n}_j$ are the area and the unit vector normal for the j-th panel, and $\vec{x}_i$ is the i-th collocation point.

Typical radiation/diffraction programs solve (5) using either direct factorization or an iterative procedure. Direct factorization grows in computational cost like N^3, and so is too expensive for problems with more than a few hundred panels. The computational costs of iterative procedures grow like N^2, so such methods can be used for problems with up to a few thousand panels. However, to analyze a structure like a platform supported by an array of cylinders, order 10^5 panels may be needed. It is hopeless to use a standard iterative procedure to solve such a problem, since storing the associated matrix would require order 10^2 gigabytes.

3 The Precorrected FFT Algorithm

When an iterative procedure is used to solve (5), only matrix-vector products are required. To see this, consider that most iterative procedures solve (5) by guessing a p^0, evaluating the resulting residual

$$r^0 = Pf - Dp^0, \tag{7}$$

and then updating the guess by computing $p^1 = F(p^0, r^0)$, where the function F depends on the specifics of the iterative method. It is possible to use the fact that iterative methods only require matrix-vector products to both: avoid the N^2 cost of explicitly forming and storing P and D, and to reduce the time of computing $Pf - Dp^0$ to typically order $N \log N$. Such an approach, the precorrected-FFT algorithm, is presented below. Only the computation of Dp, which is a potential, will be described, as the computation of Pf can be handled identically. Also, for simplification, let $G^f \to G$; this loss of generality will be rectified once the basic concepts of the algorithm have been outlined.

To develop a faster approach to computing Dp, after discretizing the problem into N panels, consider subdividing the problem domain into an array of small cubic cells so that each cell contains only a few panels. If the problem were homogeneous (which it is not, in general) there would be order N cells. Several sparsification techniques for D are based on the idea of directly computing only those portions of Dp

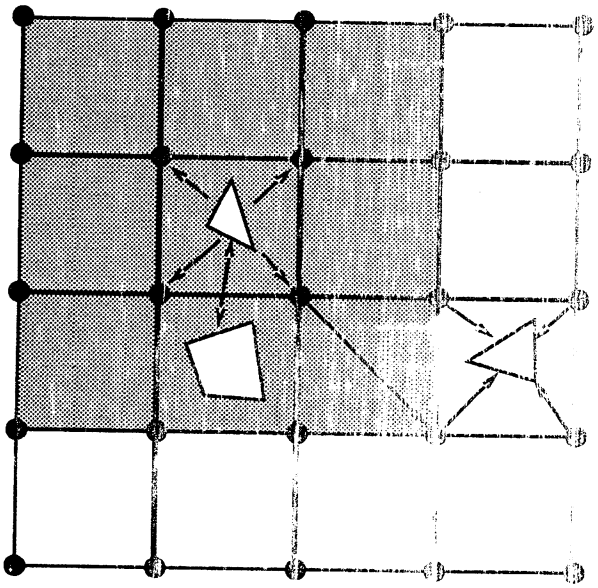

Figure 1: A 2-D schematic of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly; interactions between distant panels are computed using the grid

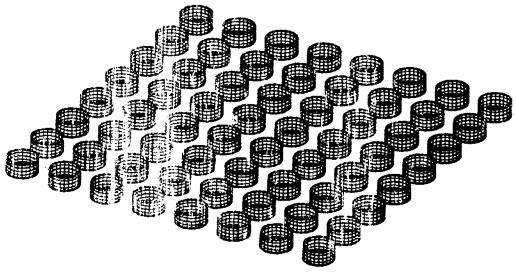

Figure 2: An [illegible] array of cylinders, each with 240 panels.

associated with interactions between panels in neighboring cells, with the rest of Dp somehow approximated to accelerate the computation. The FMA algorithm mentioned above is one example.

Another approach to computing distant interactions is to exploit the fact that at evaluation points distant from a cell an element of Dp can be computed accurately by representing the cell's singularity distribution using a small number of weighted point singularities. Figure 1 summarizes the approximate computation of Dp consisting of the following four steps:

1. Project the panel singularity distributions onto a uniform grid of point singularities (the "grid singularities").

2. Compute Dp for evaluation points at the grid points (the "grid potential") due to grid singularities.

3. Interpolate the grid potential onto the panels.

4. Directly compute the nearby interactions.

There are several possible approaches to computing the the grid singularity strengths so as to represent the panel potentials accurately. One effective approach is to require that the potential of the grid singularities representing a panel singularity distribution match the exact potential of the panel singularity distribution at carefully chosen test points. As shown in [3], a good choice of test points are Gaussian quadrature points on a surrounding sphere twice the diameter of the cube. Empirical results indicate that a $3 \times 3 \times 3$ array of grid singularities per unit cell will approximate the potential well enough to insure that the solution to (5) is accurate to 0.1%, provided that the grid singularities are used to approximate the panel potentials at least one cell distance away.

Once the grid singularities have been determined, the grid potential must be computed. The potential at a grid point $\vec{x}$ is the sum of the contributions from all of the grid singularities. Since the free-space Green function $G(\vec{x};\xi)$ depends only on the relative distance between the points $\vec{x}$ and $\vec{\xi}$, the regular grid allows the computation of the grid potential to be carried out by a three-dimensional space-invariant discrete convolution. Since this convolution is space-invariant, it can be computed using the FFT in order $M \log M$ operations, where M is the number of grid points. Or, from a linear algebra point of view, one can note

that the matrix mapping the grid singularities to the grid potential is a block-Toeplitz matrix which can be embedded in a larger block-circulant matrix, and this block-circulant matrix can be diagonalized by the discrete Fourier transform.

Once the grid potential has been computed, it must be interpolated to find the potential at the collocation points. It is easily shown that an accurate interpolation operator can be determined using the transpose of the test-point based projection operator described above.

The combination of projection, FFT-accelerated convolution, and interpolation can be used to approximately compute Dp in order $N \log N$ operations, provided the density of panels in space is relatively uniform. Unfortunately, in this approximation, the portions of Dp associated with neighboring cell interactions is not accurate. This inaccurate near-by representation must be removed and replaced with the exact direct calculation. It is possible to construct a "precorrected" direct interaction operator for the panels in two cells a and b, $D^{cor}_{a,b}$, which consists of the direct interaction operator for these neighboring cells, but with the errors introduced by the grid singularities exactly subtracted out. When used in conjunction with the grid singularity representation, $D^{cor}_{a,b}$ results in the exact calculation of the interactions between panels which are close. $D^{cor}_{a,b}$ is expensive compute initially, but costs no more to apply than the uncorrected $D_{a,b}$.

In solving potential integral equations, it is sometimes useful to reduce the size of the problem by formulating a Green function which accounts for the special geometry of a system, thereby removing part of the problem domain from consideration. For example, the use of the free-surface Green function $G^f(\vec{x};\vec{\xi})$ eliminates the need to explicitly solve for unknowns on the free surface. Unfortunately, the precorrected-FFT method described above cannot be used for such a Green function because calculating the grid potential from the grid singularities will no longer be space-invariant. However, the free-surface Green function does have a structure that can be exploited by convolution with a modified form of the FFT. Note that, at any given frequency, equation (3) may be written

$$G^f(\vec{x};\vec{\xi},\omega) = G(x-\xi, y-\eta, z-\zeta) + \tilde{G}(x-\xi, y-\eta, z+\zeta). \tag{8}$$

The difficulty for the precorrected-FFT method is that the second term depends on $z+\zeta$, a general difficulty in problems with planar interfaces. The matrix mapping the grid singularities to the grid potentials in this case is then a sum of a matrix with block-Toeplitz structure, corresponding to the first term of (8), and a matrix with block-Hankel structure corresponding to the second term of (8). The Toeplitz-like part of the matrix corresponds to the discrete convolution with the free-space Green function, and can be treated directly with the FFT as described above. Because a Hankel matrix is related to a Toeplitz matrix via a permutation matrix which is simple to compute, multiplication by a Hankel matrix may also be done with order $N \log N$ operations via the FFT. Furthermore, the permutation matrix may be represented in Fourier space so that multiplication of a vector by the sum of a Hankel and Toeplitz matrix can be performed using a single forward and inverse FFT pair. Thus at each iteration, this type of Green function may be incorporated by multiplication in Fourier space by a diagonal matrix and a permutation matrix, requiring negligible additional computation time.

4 Algorithm Performance

A preliminary indication of the performance of the precorrected-FFT algorithm for free-surface problems may be found by solving a canonical single-layer formulation (with no physical significance). For this problem only the body surface is considered, with the Dirichlet boundary condition $\psi(\vec{x}) = 1$, $\vec{x} \in S_b$ set, and the source strength which produces this potential is found by solving the first-kind integral equation. The kernel of this equation is simply the free-space Green function $G(\vec{x};\vec{\xi})$. The geometry considered is an array of cylinders as shown in Figure 2. The cylinders are discretized with 240 panels each and problems with increasing numbers of total panels are devised by adding cylinders to the array. The solutions for three cylinder arrays are computed and the computational cost is reported in Table 4. The allocated memory increases with N as expected. The time expended per iteration appears to increase at a slightly slower rate (and clearly a slower rate than $N \log N$) because overhead is being amortized over a greater number of iterations as N increases.

Number of Panels	Number of Iterations	Memory Allocation	CPU Time
3840	10	10	42
15360	14	40	182
61440	19	161	788

Table 1: Performance of the precorrected-FFT code using the free-space Green function $G(\vec{x};\vec{\xi})$ on an IBM 590 workstation. Memory allocation is reported in megabytes and CPU time is reported in seconds.

References

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

[2] 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. Comp.*, 15(3):713–735, 1994.

[3] J. Phillips, Error and Complexity Analysis for a Collocation-Grid-Projection plus Precorrected-FFT Algorithm for solving Potential Integral Equations with Laplace or Helmholtz Kernels, Colorado Conference on Multigrid Methods, April, 1995.

[4] K. NABORS, J. PHILLIPS, F. T. KORSMEYER, AND J. WHITE, Multipole and precorrected-FFT accelerated iterative methods for solving surface integral formulations of three-dimensional Laplace problems. *Domain-based Parallelism and Problem Decomposition Methods in Science and Engineering*, D. E. Keyes, Y. Saad, and D. G. Truhlar, eds., SIAM, Philadelphia, 1995.

Discussion

- **W. Schultz:** This method looks like the vortex-in-cell method. In what way is it similar and different?
- **T. Korsmeyer:** There are quite a few methods for the solution of the Poisson, Laplace, or Helmholtz equations which use the FFT to accounts for long range particle or boundary element interactions (see [1]). The distinction between methods is often in how the short range interactions are accounted for. For the method presented here, we believe the projection, interpolation, and pre-correction operations are unique.

References

[1] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles.* Adam Hilger, New York, 1988.