11th International Conference on Fast Sea Transportation
FAST 2011, Honolulu, Hawaii, USA, September 2011

Solving Nonlinear Wave-Body Interaction Problems with the Pre-Corrected Fast Fourier Transform (pFFT) Method

Sheguang Zhang, Kenneth Weems, and Woei-Min Lin

Ocean Sciences R&D Division, SAIC, Bowie, Maryland, USA

ABSTRACT

This paper presents the solutions of nonlinear wave-body interaction problems using the pre-corrected Fast Fourier Transform (pFFT) method. The advantages and limitations of the pFFT method for different applications are investigated through the comparison of the solution accuracy and algorithm efficiency to the conventional direct method. Our study shows that the significant advantage of the pFFT method over the direct method is its savings in CPU time and/or core memory when solving nonlinear problems or linear problems with a very large number of unknowns. The dependency of such savings on a few controlling parameters is also discussed.

KEY WORDS

pFFT method, potential flow panel codes, LAMP, wave-body interaction, ship hydrodynamics.

1.0 INTRODUCTION

In the early stages of ship design, the evaluation of seakeeping and maneuvering performance usually involves analyzing many combinations of state and environmental variables, such as mode of motions, speed, sea states, wave headings, etc. Thus, the number of simulations required can be very large. Moreover, when nonlinear and/or high-speed effects must be considered, time-domain simulations with high temporal and spatial resolutions are needed. The complexity of such simulations significantly increases the computation load in terms of turn-around time and memory capacity. Even with the rapid growth in computation technology in both hardware and software in recent decades, solving these nonlinear and/or large scale wave-body interaction problems is still non-trivial and quite demanding.

The most commonly used approaches for the numerical simulation of seakeeping and maneuvering of complex configurations are based on the potential flow solution of the wave-body interaction problem with boundary element methods, also referred to as "panel methods." In panel methods, the unknowns are typically associated with the source/dipole strengths on a set of panels distributed over the wetted portion of the hull and a local portion of the free surface. The number of the unknowns, N, is proportional to the number of panels used. A set of linear algebraic equations for these unknowns is posed by applying a body boundary condition on each hull panel and a free surface boundary condition on each free surface panel. The computation speed and memory requirements for panel methods are generally dominated by two parts of this procedure. The first part is evaluating the panel influence coefficients to set up the linear algebraic equation system, and the second part is solving the equation system to get the source/dipole strength on each panel.

Three types of methods are typically used to obtain the solutions to the equation system: direct methods, iterative methods, and accelerated iterative methods. These methods are briefly discussed below, with special attention to their computation requirements.

In direct methods, the coefficient matrix of order N is explicitly constructed, which requires a computation effort of order $O(N^2)$ and memory space of order $O(N^2)$. In solving the equation system, a Gauss elimination scheme is adopted, which is an $O(N^3)$ effort (e.g. LU decomposition of $O(N^3)$ and back-substitution of $O(N^2)$). The solution by direct methods can be as accurate as machine precision. However, direct methods will experience a dramatic increase in computation effort as N becomes large, and they will be constrained by available core memory. As a result, direct methods are usually restricted to solving small- to medium-sized problems, typically for $N < 10^4$. The direct methods do have a distinct advantage for linear hydrodynamic problems in which the panels, and therefore the influence coefficients, do not change with time, and the equation setup and LU decomposition need be performed only once, with just a back-substitution required at each time step.

In the iterative methods, such as the Gauss-Seidel method, the complete coefficient matrix must still be formed, so they are subject to the same memory size constraint as the direct methods. The effort required for solving the matrix using the iterative method is $O(N^2)$, compared to $O(N^3)$ for the direct methods. For linear problems, this may not be a significant advantage, because it still suffers the same memory constraint as the direct method and its iteration speed is usually slower than the back-substitution in the direct method.

The third solution method is an accelerated iterative method, which incorporates the merits of both the direct and iterative methods to speed up the solution process and minimize the memory usage while maintaining the solution accuracy at an acceptable level. One accelerated iterative method is the pFFT method. In the pFFT method, the construction of the full coefficient matrix of $O(N^2)$ is reduced to a coefficient matrix of much smaller O(N) size; the solution effort is reduced from the direct methods' $O(N^3)$ or the iterative

methods' $O(N^2)$ to only a size of O(NlogN). These inherent advantages in CPU and memory savings for larger N make pFFT an ideal candidate to solve large-scale linear ($N>10^5$) or medium- to large-scale ($N>10^3$) nonlinear wave-body interaction problems.

The pFFT algorithm was originally developed by Phillips & White (1997) for electrostatic analysis of complicated 3-D semiconductor structures. Later the method was successfully extended to solving Laplace potential problems in different fields such as the analysis of electromagnetic waves (Nie et al. 2002), magnetoencephalography (Tissari & Rahola 2003), nonlinear wave simulations for large structures (Kring et al. 1999), and nonlinear wave-body interactions with a higher order panel method (Yan & Liu 2011).

In order to speed up the panel method calculation and to make efficient use of the memory resources, a CPM (Constant Panel Method) based pFFT algorithm originally developed by Yan et al. (2006) was implemented in LAMP (the Large Amplitude Motion Program). LAMP is a time domain numerical simulation tool for the motions and loads of ships or other marine vehicles in waves and includes both linear and nonlinear formulations of the wave-body hydrodynamic problem (Weems et al. 2000). The LAMP implementation of the pFFT method was first made under the ONR HSSL (High-Speed Sealift) program in a collaborative effort with MIT (Massachusetts Institute of Technology) and has since been extended to hydrodynamic calculations involving multiple ships (LAMP_Multi) under the ONR T-Craft program. In this paper, we consider the LAMP-based application of the pFFT method for solving ship hydrodynamic problems, especially using body nonlinear formulation.

The paper is organized as follows: Section 2 gives a brief overview of the pFFT algorithm based on CPM, Section 3 describes the implementation procedures, Section 4 presents and compares the results between pFFT and the direct methods, and conclusions are given in Section 5.

2.0 OVERVIEW OF THE pFFT METHOD

Since more detailed descriptions and derivations of the pFFT method can be found in papers by Phillips & White (1997), Yan et al. (2006), and Yan & Liu (2011), only a brief overview of the method is given in this section.

The starting point of the pFFT description for ship potential flow hydrodynamic applications is the boundary integral equation (BIE) which can be written as

$$\alpha\Phi(\vec{p}) + \iint_S \left[\Phi(\vec{q})G_n(\vec{p},\vec{q}) - \Phi_n(\vec{q})G(\vec{p},\vec{q})\right]dS(\vec{q}) = 0 \quad (1)$$

where Φ and Φ_n are the velocity potential and its derivative in the direction of the surface outward normal n; $\vec{p}$ and $\vec{q}$ are the field and source points, respectively; $G = 1/r$ and $G_n = \partial G/\partial n$ with $r = |\vec{p} - \vec{q}|$ are the Rankine source Green function and its normal derivative; α is the solid angle; and S is the boundary surface that includes the wetted body surface S_b and the free surface S_f.

The panel method is based on the discretized form of Eq. (1), created by subdividing the boundary S into N surface panels and by approximating Φ (dipole strength) and Φ_n (source strength) on S with polynomials in terms of their nodal values at the panel collocation points. For low order methods such as the CPM, the surface S is composed of N flat panels, and Φ and Φ_n are piecewise constants with the collocation point at the panel center. With the CPM, Eq. (1) can be written as

$$\alpha\Phi(\vec{p}) + \sum_{j=1}^{N_b}\Phi_j\iint_{S_j} G_n(\vec{p},\vec{q})dS_j - \sum_{j=1}^{N_f}\Phi_{nj}\iint_{S_j} G(\vec{p},\vec{q})dS_j = -\sum_{j=1}^{N_f}\Phi_j\iint_{S_j} G_n(\vec{p},\vec{q})dS_j + \sum_{j=1}^{N_b}\Phi_{nj}\iint_{S_j} G(\vec{p},\vec{q})dS_j \quad (2)$$

where the surface integration is carried out on a panel-by-panel basis over N_b panels on S_b and over N_f on a local portion of S_f. Φ_j and Φ_{nj} are the velocity potential and its normal derivative at the center of panel j. It can be seen that as the field point $\vec{p}$ is chosen at the N collocation points, Eq. (2) produces four matrix-vector product terms: two on each side. Conventionally, this leads to a linear algebraic equation system with N unknowns:

$$\mathbf{Ax} = \mathbf{b} \quad (3)$$

where vector **x** contains the unknowns such as Φ on S_b and Φ_n on S_f.; the coefficient matrix **A** is dense and requires $O(N^2)$ effort to setup and $O(N^2)$ memory to save; the right-hand-side vector **b** is the product of the influence coefficients and the known values of Φ_n on S_b and Φ on S_f. Eq.(2) can be solved with the direct method of $O(N^3)$ effort, including both decomposition and back-substitution, or with the iterative method of $O(N^2)$.

The pFFT method, although an iterative method in nature, follows a different route in solving Eq. (2) by accelerating the solution process and reducing the memory usage. The fundamental reason that pFFT can achieve such computation efficiency is that the influence on a given field point due to the singularity distributions (source or dipole) on a far field panel can be approximated by a set of concentrated sources/dipoles distributed on a 3-D grid surrounding the far field panel. Taking advantage of this nature, the pFFT method divides the total influence on a field point into two parts: the far field part that is associated with the singularity distributions on the source panels at the far field and the near field part that is due to source panels nearby. A criterion can be set to determine the far and near field panels by balancing the solution accuracy and efficiency. The key features of the pFFT are that (a) it directly computes the matrix-vector product in Eq. (2) for the far field influence without the need to explicitly construct the NxN matrix **A**, thereby reducing the memory burden, and (b) these matrix-vector products are computed on a uniform 3-D grid system with the FFT algorithm,

which requires operations of O(NlogN) instead of the much larger $O(N^2)$ for the iterative or $O(N^3)$ for the direct when N is large.

The pFFT procedure includes the following steps:

- Creation of a 3-D uniform grid system that encloses the entire solution domain, i.e. the body surface S_b and the local free surface S_f. This builds a background grid system as a platform on which the FFT can be performed. For CPM, the grid points for each 3-D cell are at the cell vertices, e.g. 2^3=8 grid points per cell. The total number of grids is denoted as N_g, which will determine the FFT operation of $O(N_g \log N_g)$. In this paper we roughly assumed $O(N_g) \sim O(N)$.
- Projection of the product $\iint_{S_j} \Phi G_n dS_j$ and $\iint_{S_j} \Phi_n G dS_j$ on panel j, j=1, 2, ..., N, in Eq. (2) onto the grids of the 3-D cell that encloses the panel. With respect to each field point $\vec{p}$, the kernels G and G_n are first represented in terms of shape functions and their nodal values at the vertices of the 3-D cell, e.g.

$$G_n(\vec{p},\vec{q}) = \sum_{k=1}^{8} H_k(\vec{q}) G_n(\vec{p},\vec{q}_k) \tag{4}$$

  where k is the index of the eight grids at the cell corners and $H_k(\vec{q})$ is the k^{th} shape function of a polynomial form. Then, the influence on $\vec{p}$ from the source panel j is made equivalent to that from the concentrated sources at the grid points of the projecting cell:

$$\Phi_j \iint_{S_j} G_n(\vec{p},\vec{q}) dS_j = \sum_{k=1}^{8} G_n(\vec{p},\vec{q}_k) Q_k \tag{5}$$

  where $Q_k = \Phi_j \iint_{S_j} H_k(\vec{q}) dS_j$ is the equivalent point source on grid k. Eq. (5) establishes the relation between the singularity strength on panel j and those on grid k, k=1, 2, ..., 8. The process is completed by carrying out the projection through N field points for $\vec{p}_i$, i=1, 2, ..., N. Note that only the projection of the influence at the far field panels are relevant in the pFFT calculation.
- Convolution performed on the 3-D grids with FFT.

  The matrix-vector products projected on the 3-D grids are in discrete convolution form, which can be computed using FFT algorithms. The LAMP implementation of the pFFT method uses FFTW, a highly efficient and free subroutine library developed at MIT and available through www.fftw.org.
- Interpolation of the matrix-vector product on the 3-D grids back onto the panel collocation points. This is a reverse process of the interpolation, e.g. the right hand side of Eq.(5) is used to represent the LHS term $\Phi_j \iint_{S_j} G_n dS_j$. The interpolation also applies to the similar term for $\Phi_{nj} \iint_{S_j} G dS_j$.
- Direct calculation and pre-correction for the near field influence coefficients. The previous three steps – projection, convolution, and interpolation – are mainly for the far field influence computations that take the majority of the computation effort. For the near field influences, which require a more accurate scheme, a direct panel-to-panel calculation is needed. This is the same algorithm as the conventional direct method but at a much reduced scale, since the number of near field influences, N_n, that need direct calculation for a given field point is typically much less than N. The computation effort is $O(N_n^2)$ which is roughly O(N) for a typical configuration. In the near field, these directly computed panel-to-panel influences will be used and the FFT computed grid-to-panel influences by the interpolation will be eliminated. This completes the pre-correction phase before the iteration starts.

These steps are built into the GMRES (Generalized Minimum Residual) iterative method and executed at every step within the iteration loop. A pre-conditioner treatment can be introduced to further accelerate the iteration.

3.0 NUMERICAL IMPLEMENTATION

3.1 Generation of the 3-D Background Grids

A prerequisite of running the pFFT method is setting up a 3-D background grid. The background grid forms a 3-D box that encloses all surface panels in the solution domain, e.g. those on the body surface S_b and the local portion of the free surfaces S_f. Fig. 1 shows a sample 3-D Cartesian grid that has 40x21x7 grid lines in the longitudinal, lateral, and vertical directions, with a uniform grid distribution in each of the three directions. The box is subdivided into 3-D cells. The nominal size the 3-D cell is denoted as h, which is usually larger than the nominal panel size. In the present implementation, all grid points are located at the 3-D cell corners.

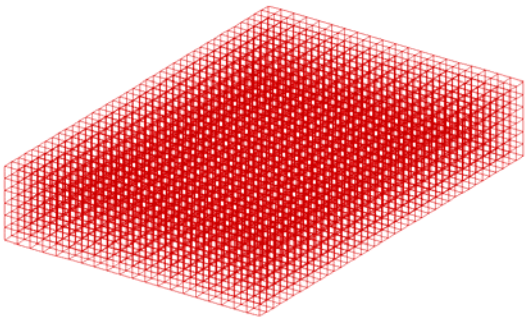

Fig. 1. Sample 3-D background grid

3.2 Incorporation of pFFT Algorithm with LAMP

The pFFT algorithm was implemented by integrating MIT's sample pFFT algorithm into LAMP's existing direct solution method. The major effort is to compute the matrix-vector products on the LHS and RHS of Eq. (2). Products that involve the far field influences are performed by FFT on the 3-D background grids following the procedures of the projection, convolution, and interpolation; products that involve the near field influences are directly calculated with LAMP's existing utilities. The FFT is computed using the free FFTW library. The iteration utility is GMRES with preconditioned accelerators.

3.3 Classification of Near vs. Far Field Panels

It is necessary to classify the distance to a field point at which a source panel is considered as a near field panel.

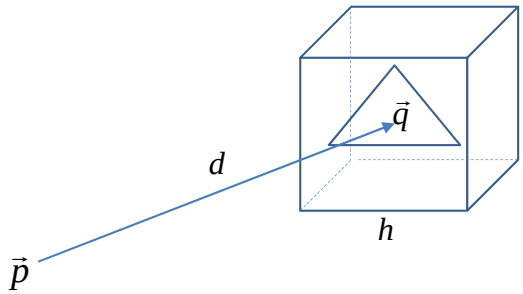

Fig. 2. Illustration of the field point, source panel, and 3-D background cell

In Fig. 2 we denote this distance as d and use a non-dimensional parameter d/h to describe the near field distance in the pFFT implementation. If the near field distance d is defined as quadruple the cell size h (i.e. d/h=4), then source panels that are located within 4h distance to a field point will be treated as near field panels and their influence to the field point will be calculated directly; source panels farther than 4h will be classified as far field panels and their influences will be calculated based on the pFFT algorithm.

4.0 NUMERICAL RESULTS

The results presented in this section illustrate the accuracy, efficiency, and some limitations of using the pFFT method. For accuracy comparisons, the solution obtained by the pFFT method is compared with the direct method; for efficiency comparisons, the pFFT and direct methods are executed in the similar computation environment for the same numerical test. The resulting CPU and memory usages are recorded and compared. The examples in the subsections address one issue at a time.

4.1 Flared Body with Prescribed Heave

Fig. 3 presents a flared body performing a prescribed large amplitude heave motion.

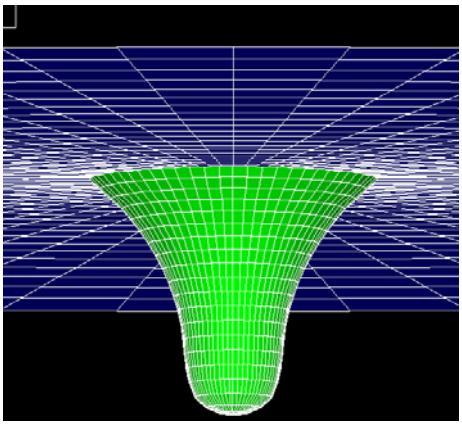

Fig. 3. A flared body in a prescribed heave motion

Because of the large amplitude motion, the body nonlinear approach is needed to capture the nonlinear hydrodynamic effect. The hydrodynamic forces computed with both the pFFT and direct methods are presented in Fig. 4.

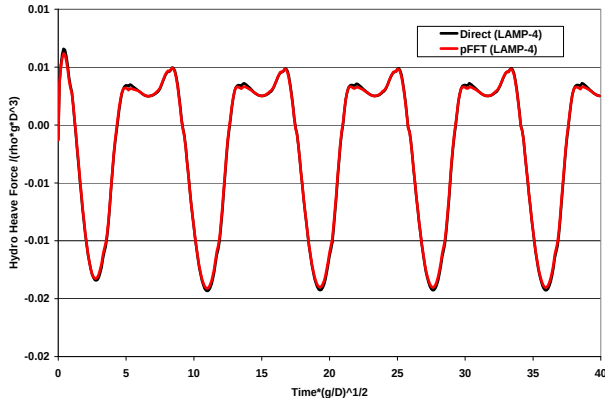

Fig. 4. The hydrodynamic have force on the flared body with d/h=1

The difference between the two methods is negligible, with the pFFT result (red) overlaying on the direct (black). In these calculations, the total number of panels is N=2800 with N_b=1200 and N_f=1600, while the total number of pFFT grid points is N_g=5880. By changing the near field distance ratio d/h, the solution errors are calculated and presented in Fig. 5.

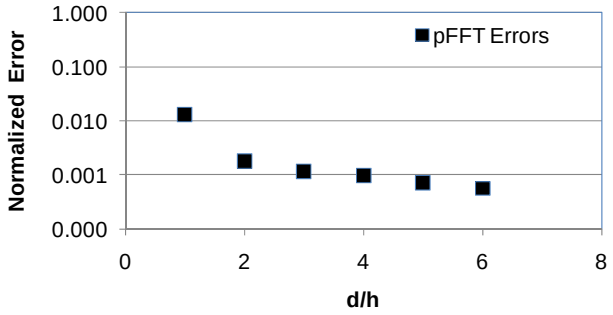

Fig. 5. The solution error vs. d/h

The solution error is the RMS error of the pFFT solution relative to that of the direct solution, $\sqrt{\sum_i (F_{iF} - F_{iD})^2 / n}$. It is normalized with respect to $\sqrt{\sum_i F_{iF}^2 / n + \sum_i F_{iD}^2 / n}$ and plotted in Fig. 5, where the summation is over the number of samples and the subscripts F and D refer to the pFFT and the direct methods. The closer the two solutions are to each other, the closer the normalized error approaches to 0; otherwise, the error approaches to 1. Fig. 5 shows that d/h=1 results in about 1% relative error while d/h=4 will reduce the error to 0.1%. With the number of panels fixed, larger d/h implies that more panels will be treated as the near field panels and their influences to the field point $\vec{p}$ will be calculated with the direct method. This means that the pFFT accuracy will approach the direct method as d/h gets larger but at a higher computation cost, therefore reducing the benefit of using pFFT.

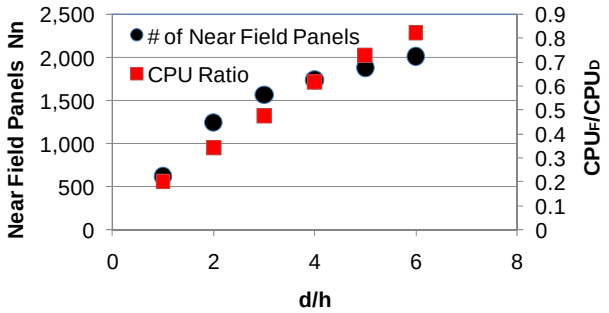

Fig. 6. The number of near field panels and the ratio of CPU pFFT to CPU direct vs. d/h

Fig. 6 plots the number of near field panels and the CPU ratios as a function of d/h. It shows that as d/h increases, more source panels are treated as near field panels (N_n) and the computation cost increases at the same time. In this example, the direct method used 2800 panels and 107 minutes CPU for a 400 time step simulation.

It can be seen that to take advantage of the pFFT, it is necessary to balance the computation accuracy shown in Fig. 5 and the efficiency shown in Fig. 6. For example, if 1% normalized error is acceptable, then d/h=1 can be used with the pFFT computation time only about 20% of the direct method.

4.2 Submerged Slender Body beneath the Free Surface

This example examines the computation effort spent on different components of the solution process. The case is an ellipsoid translating beneath the free surface at Fn=0.2. The background grids are constructed as described in Section 3.1 and remain fixed while the number of panels varies. Only one time step is executed using the direct and the pFFT methods. The results are listed in Table 1 and Table 2, respectively.

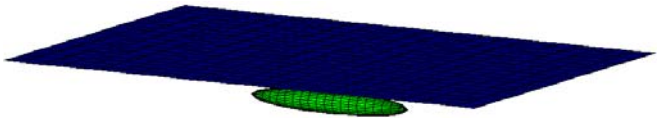

Fig. 7. An ellipsoid translating beneath the free surface at Fn=0.2

Table 1. CPU and memory usage – Direct Method

# of Panels	CPU (mm:ss)				Memory (mb)	
N	Setup	Decomp	Bksub	Total	VIRT	RES
2000	00:00.4	00:05.1	00:00.0	00:05.7	1774	53
4000	00:01.4	00:46.0	00:00.0	00:48.0	1744	161
8000	00:07.2	06:06.1	00:00.1	06:14.9	1744	561
12000	00:23.2	21:04.5	00:00.2	21:31.1	1744	1100

In Table 1, the total CPU time for a one step run with the direct method is composed of three parts: setup of the full coefficient matrix, LU decomposition, and back-substitution. It can be seen that 98% of the total CPU time is used for the LU decomposition, 2% for the matrix setup, and less than 1% for the back-substitution. The memory usage is measured either by the size of the virtual image (VIRT) or by the resident memory (RES). It can be seen that VIRT, which is mainly determined by the size of the

compiled executable, remains fixed as panel number N increases while RES increase with N.

For the pFFT method, the corresponding CPU and memory data are listed in Table 2.

Table 2. CPU and memory usage – pFFT Method

# of Panels		CPU (mm:ss)			Memory (mb)	
N	Nn	Setup_near	GMRES	Total	VIRT	RES
2000	670	00:01.6	00:00.2	00:03.6	1377	75
4000	1337	00:06.1	00:00.5	00:10.5	1377	170
8000	2676	00:23.8	00:01.8	00:39.9	1377	445
12000	4030	00:53.2	00:03.7	01:29.8	1377	792

With d/h=4, it can be seen that the number of near field panels N_n is about a third of the total panels N. The resulting total CPU time is composed of the setup time for the near field influence, GMRES iterations, and other pFFT executions that are not recorded as standalone entries. Of the total CPU time, the near field influence setup takes about 60%, GMRES 5%, and the remaining pFFT procedures take the rest at about 35%.

Comparing the two tables for the one-step execution, the total CPU ratio of the pFFT to the direct ranges from 60% for smaller N (=2000) to 7% for larger N (=12000). In other words, the larger the N, the greater the CPU savings with the pFFT method over the direct method. However, it should be noted that these CPU comparisons are for a single time step execution only. For body-linear problems using the direct method, the coefficient matrix is set up and the LU decomposition is completed at the first time step, and subsequent time steps involve only the back-substitution, it can be very efficient and faster than the pFFT iteration schemes. Therefore, the pFFT method may not be faster in total computational speed for body-linear problems.

Resident memory usage increases as N becomes larger for both methods. However, it increases at a slower rate with the pFFT method than with the direct method. This is shown in Fig. 8, which plots the ratio of the RES pFFT to RES direct.

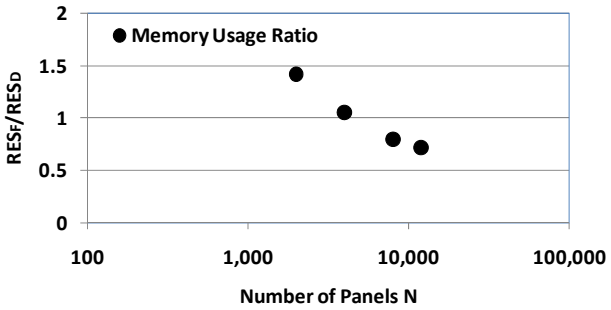

Fig. 8. The memory usage ratio vs. number of panels

From Fig. 8, it can be expected that the pFFT method can reduce the required RES size by 50% or more for $N>10^4$. Note that the results in this example are based on the fixed d/h=4. If d/h value can be lowered without sacrificing too much in solution accuracy, the pFFT savings in CPU and memory can be further improved.

4.3 T-Craft Moving in Head Seas at 30 knots

As described in Section 4.2, the pFFT method is not necessarily faster for a body-linear problem with many time steps. This limitation for the pFFT method, and therefore a merit for the direct method, is illustrated in an example calculation for the T-Craft moving in head seas at 30 knots for 4000 time steps.

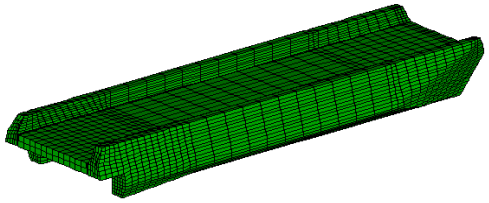

Fig. 9. T-Craft hull geometry

The T-Craft is modeled as a SES (Surface Effect Ship) with length of 2.4m and scale ratio 1:30. The LAMP calculation includes a dynamic calculation of the air cushion pressure that is applied to both the underside of the hull surface and the portion of the free surface beneath the cushion.

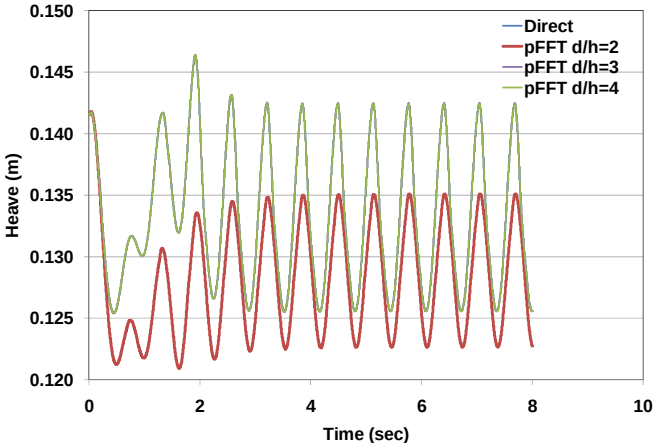

Fig. 10. Heave time history: direct (blue) vs. pFFT with d/h=2 (red), =3(purple), =4(green)

Fig. 10 shows the heave time history (model scale) from calculations using the direct method and the pFFT method with three d/h values (h~0.25m). As far as convergence and accuracy are concerned, the pFFT method with d/h=3 and 4 are nearly identical to the direct result, but the d/h=2 calculation is not acceptable because fewer panels are included as near field panels and their influences to a field point are not evaluated accurately. This implies that a minimum d/h value of 3 is needed for this configuration. The CPU time comparison in Table 3 shows that the direct method is fastest.

Table 3. CPU and memory comparisons

Method	# of Panels		CPU (min)	Memory (mb)
	N	Nn	Total	RES
pFFT d/h=2	5162	274	161	159
pFFT d/h=3	5162	503	206	179
pFFT d/h=4	5162	805	344	204
Direct	5162	n/a	40	253

While the direct method does need more CPU time to set up the coefficient matrix and perform the LU decomposition

during the first time step, subsequent time advancement requires only back-substitution (see Table 1). In pFFT, the matrix-vector products for the far field panels still need to be calculated every time step whether the problem is body-linear or body-nonlinear, so the pFFT and GMRES takes longer than the back-substitution (see Table 2).

In order to benefit from the pFFT method, the ship hydrodynamic problems should be either (a) body-nonlinear problems of medium to large size where $N>10^3$ or (b) body-linear problems of very large size (e.g. $N>10^5$) with which the direct method could not run because of machine memory constraints.

4.4 ONR Tumblehome Hull Model 5613 with Large Amplitude Roll

This section presents a typical application scenario in which the pFFT method is necessary and beneficial. In this case, the body-nonlinear effect is important and the differences between the body-linear and body-nonlinear can be significant. We consider an ONR Tumblehome hull form (Model 5613-1) as shown in Fig. 11.

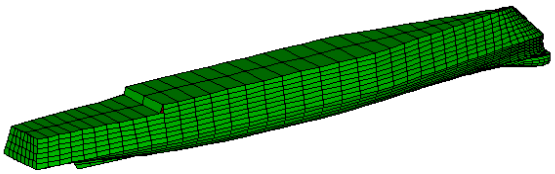

Fig. 11. ONR Tumblehome 5613-1 hull geometry

The ship is traveling at Fn=0.3 with prescribed large amplitude heave and roll in a regular beam sea ($\lambda/L=1.0$, $H/\lambda=1/20$). The ship length is L=154m. The heave and roll amplitudes are 7.7 m and 18 degrees (full scale), respectively, with the same frequency as the incident wave frequency. Two runs are made – the body-linear and body-nonlinear – with both the direct and pFFT methods. The total number of panels N=2989. For the pFFT method, the total number of grids N_g=5880, the near field distance parameter d/h=4, and the tolerance for GMRES convergence is set to $1.e^{-4}$. With these parameters, the pFFT results converge to the direct with the normalized error ~0.001 for both the body-nonlinear and body-nonlinear runs, so the results of the two solution methods are nearly identical.

The difference between the body-linear and body-nonlinear runs is significant. The computed hydrodynamic pitch moment is compared in Fig. 12.

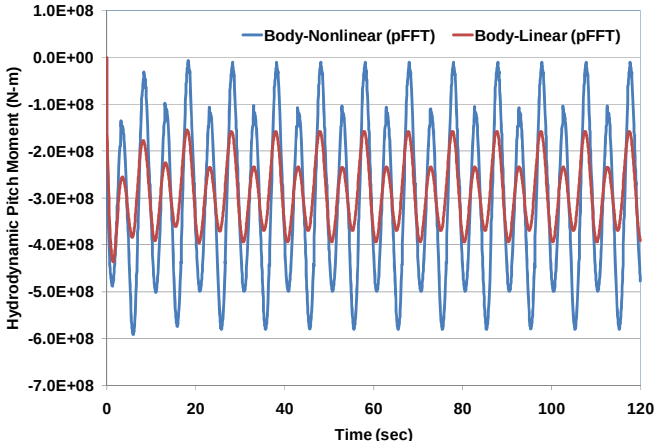

Fig. 12. The hydrodynamic pitch moment for the ONR Tumblehome hull with large forced heave and pitch motion in regular waves

These results indicate that the body-nonlinear calculation may be necessary for large amplitude motions. For the body-nonlinear simulations with 2000 time steps, the pFFT method took about 3 hours of CPU time and the direct method took about 20 hours. In this case, the pFFT achieves 85% CPU savings over the direct method while maintaining the solution error at a level of 0.001. The computation efficiency can be further enhanced by optimizing the near field distance ration d/h, for instance by using d/h=3 instead of 4.

4.5 T-Craft Passing a LMSR

This case involves two ships moving relative to each other in close proximity. The previous example in Section 4.4 is for a case in which both the body-linear and body-nonlinear approach can be adopted but the body-nonlinear approach is more appropriate to capture the physics associated with large amplitude motion and waves. In this multi-ship case when one is passing another, it is a body-nonlinear problem by nature regardless of how large the vertical motion and waves is. Because the relative position between the two ships in the lateral plane is changing with time, the influence coefficients between source panels and field points must be computed at every time step to reflect the body-nonlinearity of this relative motion.

To demonstrate the necessity and merit of using the pFFT method for such a problem, calculations were performed for the T-Craft SES passing close by a LMSR (Large, Medium Speed, Roll-on/Roll-off) heavy sealift ship in waves. In addition to requiring the body-nonlinear formulation due to the large relative motion, the problem requires a very large number of panels to model both ships and a fairly large local free surface domain. In the simulation, the T-Craft is initially positioned at the LMSR's stern while moving at a forward speed of 10 knots; the LMSR is moving in the same direction at 2 knots. The side-by-side separation between the two is 15 feet. Both vessels have 4-DOF, with constraints in sway and yaw to maintain ship heading. The incident waves are long-crested head seas at a high Sea State 4.

Fig. 13 plots the ship positions at the beginning and the end of passing as well as the free surface contours obtained by the pFFT.

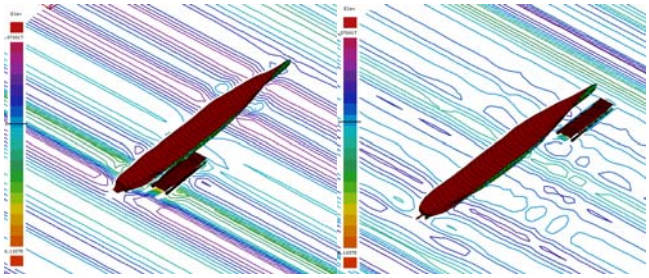

Fig. 13. T-Craft passing a LMSR in high Sea State 4: starting position (left) and ending position (right)

The time histories of the pitch motion of the two ships are plotted in Fig. 14. The results by the two solution methods are almost identical. However, because of the large panel number (N=11200) and body-nonlinearity, running LAMP with the direct method is prohibitively slow. In one test, the direct method required 37 hours but the pFFT method needed 3 hours, which is one order of magnitude faster than the direct.

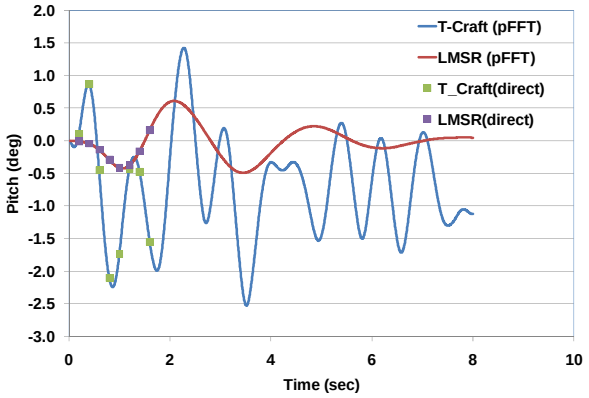

Fig. 14. Pitch motion of T-Craft and LMSR while passing in head seas – calculated using direct and pFFT methods

5.0 CONCLUSIONS

This paper presents the results from the application of a pre-corrected Fast Fourier Transform (pFFT) solution method for time-domain seakeeping calculations using the Large Amplitude Motions Program (LAMP). The results illustrate the advantages as well as limitations of using the pFFT method to solve the nonlinear wave-body interaction problems.

The most significant advantage of the pFFT method over the conventional direct method is its computation efficiency in solving the body-nonlinear problems of medium to large sizes ($N>10^4$). Because the solution accuracy and speed can be controlled through a number of parameters – such as iteration tolerance (e.g. $=1.e^{-4}$), near field distance d/h (e.g. =4) and number of 3-D grids – the pFFT method can be optimized for a specific application to maximize its benefit. For the body-nonlinear problems presented in this paper, the CPU time used by the pFFT method can be one order of magnitude smaller than the direct method while keeping the solution accuracy at an acceptable level.

The secondary advantage of the pFFT is the memory savings. This will become critical when the number of unknowns becomes large (e.g. $N>10^5$).

For body-linear transient problems, the back-substitution needed for the direct method at each time step is sufficiently faster, versus the pFFT accelerated iteration method for most configurations. For these cases, the pFFT method becomes necessary only when the problem size is sufficiently large that the memory required by the direct method exceeds the machine capacity.

REFERENCES

Nie, X.C., Li, L.W., & Yuan, N. (2002). "Precorrected-FFT Algorithm for Solving Combined Field Integral Equations in Electromagnetic Scattering." Journal of Electromagnetic Waves and Applications, **16**(8): 1171-1187.

Kring, D., Korsmeyer, T., Singer, J., Danmeier, D., & White, J. (1999). "Accelerated, Nonlinear Wave Simulations for Large Structures." Proceedings of the 7th International Conference on Numerical Ship Hydrodynamics, Nantes, France.

Phillips, J.R., & White, J.K. (1997). "A Precorrected-FFT Method for Electrostatic Analysis of Complicated 3-D Structures." IEEE Transactions on Computer-Aided Design on Integrated Circuits and Systems, **16**(10): 1059 -1072.

Tissari, S., & Rahola, J. (2003). "A precorrected-FFT Method to Accelerated the Solution of the Forward Problem in Magnetoencephalography." Physics in Medicine and Biology, **48**:523-541.

Weems, K.M., Lin, W.M., Zhang, S., & Treakle, T.E. (2000). "Time Domain Prediction for Motions and Loads of Ships and Marine Structures in Large Seas Using a Mixed-Singularity Formulation," Proceedings of the Fourth Osaka Colloquium on Seakeeping Performance of Ships (OC2000), Osaka, Japan, 272-280.

Yan, H., Liu, Y., & Yue, D.K.P. (2006). "An Efficient Computational Method for Nonlinear Three-Dimensional Wave-Wave and Wave-Body Interactions." Conference of Global Chinese Scholars on Hydrodynamics, Shanghai, China.

Yan, H., & Liu, Y. (2011). "An efficient high-order boundary element method for nonlinear wave–wave and wave-body interactions." Journal of Computational Physics, **230**: 402-424.

ACKNOWLEDGEMENTS

The funding for this study was provided by the Office of Naval Research (ONR) through the HSSL program managed by Dr. L. Patrick Purtell and the T-Craft program managed by Ms. Kelly Cooper. Their support is greatly appreciated. The authors would also like to thank Dr. Hongmei Yan and Dr. Yuming Liu from MIT who provided essential pFFT subroutines for implementation in LAMP.