

A fast method for nonlinear three-dimensional free-surface waves

BY CHRISTOPHE FOCHESATO AND FRÉDÉRIC DIAS*

Centre de Mathématiques et de Leurs Applications, Ecole Normale Supérieure de Cachan, 61 avenue du Président Wilson, 94235 Cachan Cedex, France

An efficient numerical model for solving fully nonlinear potential flow equations with a free surface is presented. Like the code that was developed by Grilli *et al.* (Grilli *et al.* 2001 *Int. J. Numer. Methods Fluids* **35**, 829–867), it uses a high-order three-dimensional boundary-element method combined with mixed Eulerian–Lagrangian time updating, based on second-order explicit Taylor expansions with adaptive time-steps. Such methods are known to be accurate but expensive. The efficiency of the code has been greatly improved by introducing the fast multipole algorithm. By replacing every matrix–vector product of the iterative solver and avoiding the building of the influence matrix, this algorithm reduces the computing complexity from $O(N^2)$ to nearly $O(N)$, where N is the number of nodes on the boundary. The performance of the method is illustrated by the example of the overturning of a solitary wave over a three-dimensional sloping bottom. For this test case, the accelerated method is indeed much faster than the former one, even for quite coarse grids. For instance, a reduction of the complexity by a factor six is obtained for $N=6022$, for the same global accuracy. The acceleration of the code allows the study of more complex physical problems and several examples are presented.

Keywords: boundary-element method; surface-water waves; nonlinear waves; fast multipole algorithm; numerical wave tank

1. Introduction

For several decades, fully nonlinear potential flow equations have been used to model water waves. The governing equation is Laplace’s equation, subject to nonlinear boundary conditions. The numerical simulation of water waves is a vast topic. The main progress witnessed in the last decade is the advent of time-dependent three-dimensional calculations, due to the rapidly increasing power of computers. At the same time, two-dimensional calculations have also matured, producing either highly accurate results or calculations over impressively large grids. The reader is referred for example to the review by Dias & Bridges (in press). The main approaches for solving exactly or approximately the time-dependent three-dimensional water-wave equations can be classified into three categories: the methods that solve model equations such as nonlinear Schrödinger type equations or Boussinesq type equations, the high-order spectral methods and the methods based on boundary integrals. The main advantage of

* Author for correspondence (dias@cmla.ens-cachan.fr).

using model equations is that significantly larger spatial and temporal scales can be considered (see Fuhrman *et al.* 2004 for the state of the art on such methods). The main disadvantage is that the ranges of validity are limited. The main advantage of high-order spectral methods, which are based on series expansions, is that they are computationally efficient and robust when the series converges (see Bateman *et al.* 2001 for the state of the art on such methods). The main disadvantage is that the series do not always converge and numerical instabilities develop. The main advantage of boundary integral equation methods (BIEM) is that they are accurate and can describe highly nonlinear waves. However, they are expensive and, as clearly said by Nishimura (2002), they have been considered losers in large problems for that reason. Recent developments on fast multipole-accelerated boundary integral methods have revealed that the discretized equation for BIEM may possibly be solved with $O(N)$, at least in integral equations for Laplace's equation. Here N is the number of unknowns introduced to discretize the boundary integral equation (BIE). Surprisingly, the BIEM community in applied mechanics seems to remain rather indifferent to such developments (see again Nishimura 2002). In the water-wave community, there is even more scepticism. When the present work was initiated 3 years ago, several researchers had some doubts on the feasibility of the proposed numerical wave tank, probably due to earlier failures by various teams. It is true that the fast multipole algorithm (FMA) does not exist as a black-box implementation and requires significant work to be inserted in any new context.

The main results of the present paper are the derivation and the validation of a rapid boundary-element method (BEM) that will make large-scale problems in three-dimensional free-surface waves more tractable. The method recently developed by Fructus *et al.* (2005), which is also based on an integral formulation but makes an extensive use of Fourier transforms, is an interesting alternative but is restricted so far to flat bottoms.

The capability of BEM to describe accurately nonlinear water waves in three dimensions has been demonstrated by several authors, such as Romate & Zandbergen (1989), Broeze *et al.* (1993), Grilli *et al.* (2001), Xue *et al.* (2001) and Guyenne & Grilli (2006). Arbitrary waves can be generated over a non-uniform bottom with various boundary conditions. The method used for example by Grilli *et al.* (2001) combines a high-order BEM with a mixed Eulerian–Lagrangian temporal scheme, based on explicit Taylor expansions. The key features of this model will be reviewed in §2.

The main drawback of this kind of discretization is the necessity to assemble and solve full linear systems, as opposed for example to finite-difference methods or finite-element methods, which lead to banded coefficient matrices. With the use of an iterative solver such as GMRES (generalized minimal residual), both the assembling and solving steps are $O(N^2)$. There are two major ways to improve the model efficiency. The first one consists in sharing the work and taking advantage of parallel computers (e.g. Wang *et al.* 1995). The second way is to speed up the computation of matrix–vector products with a fully populated matrix. A decomposition of the matrix can be used through particular transforms, such as fast Fourier transforms for irregular grids (Potts & Steidl 2003) or wavelet transforms (Alpert *et al.* 1993). Another idea is to use hierarchical algorithms. Several methods share the same idea: \mathcal{H} -matrices, including the panel clustering method (Hackbush 1999), mosaic skeleton

matrices (Goreinov *et al.* 1997) and the FMA all give a hierarchical block sparse structure to the approximate matrix. The first two are algebraic techniques, whereas the FMA requires knowledge of the functions involved in the approximation of the interaction kernel by an expansion in separable functions.

First developed by Greengard & Rokhlin (1987) for the N -body problem, the FMA allows a faster computation of all pairwise interactions in a system of N particles, in particular the interactions governed by Laplace's equation. So, it is well suited to our problem and we chose to apply this technique. The idea of the algorithm is based on the fact that the interaction strength decreases with distance, so that far away points can be placed in groups to contribute at one collocation point. A hierarchical subdivision of space automatically gives distance criteria to distinguish close interactions from far ones. In this last case, the kernel is approximated by an expansion where both spatial variables are separated. For Laplace's equation, the interactions are represented by Green's functions which can be expanded into spherical harmonics. The main components of the algorithm are given in §3 and for a more detailed description the reader is referred to Greengard (1988) or Beatson & Greengard (1997).

The outcome turned out to be very efficient, especially in two dimensions. In three dimensions, optimal implementations have been more difficult to obtain and several improvements have been suggested. Indeed, there is a loss in efficiency if one wants great accuracy in three dimensions, in particular for non-uniform distributions of particles. We do not attempt here to list all the work on the method. We just indicate that Cheng *et al.* (1999) have designed a better version with new translation techniques. However, for a reasonable accuracy, former FMA gives sufficiently good results compared to direct evaluation of interactions in order to be incorporated in various numerical models (see Nishimura 2002 for a review). We can also mention some work on a generalization of the algorithm so that it becomes kernel-independent (Anderson 1992). This kind of research should lead to black-box implementations of the FMA. Nevertheless, applications often require specific implementations in order to get the best efficiency.

The fast algorithm can be used alone to solve Laplace's equation, but it can also be associated with an integral representation of this equation. The discretization then leads to a linear system with matrix–vector products of an iterative solver that can be accelerated by the FMA. Rokhlin (1985) applied this idea to the equations of potential theory. After Rokhlin, the technique has spread in various areas for different kinds of discretization, in particular for BEM (Grama *et al.* 1999; Darve 2000).

Applications in fluid mechanics seem to be rather sparse compared to other fields. However, there are several possibilities to use the FMA for incompressible flows: potential flow equations as first suggested by Rokhlin (1985), but also Stokes flow (Gómez & Powert 1997; Mammoli & Ingber 2000; Kropinski 2001), vortex methods (Pringle 1994; Scorpio & Beck 1996) and the possibility to come as a component of Navier–Stokes solvers via generalized Helmholtz decomposition (Brown *et al.* 2003). Water-wave computations with multipole-accelerated codes also exist. Korsmeyer *et al.* (1993) applied the FMA with a BEM through a Krylov-subspace iterative algorithm. Following Rokhlin's ideas, they designed a modified multipole algorithm for the equations of potential theory. First developed for electrostatic analysis, their code has been generalized to become

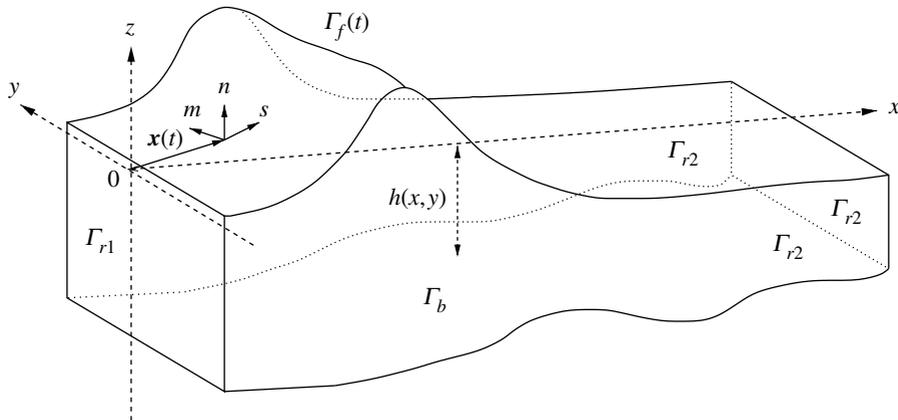


Figure 1. Computational domain. The free surface $\Gamma_f(t)$ is defined at each time-step by the position vector $\mathbf{x}(t)$. Lateral boundaries are denoted by Γ_{r1} and Γ_{r2} . The bottom Γ_b is defined by $z = -h(x, y)$. Use is made of the Cartesian coordinate system (x, y, z) and of the local curvilinear coordinate system (s, m, n) defined at the point $\mathbf{x}(t)$ of the boundary.

a fast Laplace solver, which subsequently has been used for potential flows. They got an efficient model but the global accuracy was limited by the use of low-order elements. [Scorpio & Beck \(1996\)](#) studied wave forces on bodies with a multipole-accelerated desingularized method, and thus did not use boundary elements to discretize the problem. Neither did [Graziani & Landrini \(1999\)](#) who used the Euler–McLaurin quadrature formula in their two-dimensional model. We show in §4 how the FMA can be inserted in the numerical wave tank designed by [Grilli *et al.* \(2001\)](#) in order to get a more efficient tool. Numerical experiments are carried out in §5 to show the accuracy and the improved efficiency of the new model. Future possibilities of the model are discussed. Finally, the conclusion sums up the characteristics of the accelerated model.

2. The numerical wave tank

The main aspects of the numerical wave tank are presented in this section. We focus on the points dealing with building the fast algorithm. Details of the method can be found in [Grilli *et al.* \(2001\)](#) or in [Fochesato \(2004\)](#). The goal is to compute gravity surface waves under the assumptions that the fluid is incompressible, inviscid and that the flow is irrotational. The numerical method follows the general scheme developed by [Longuet-Higgins & Cokelet \(1976\)](#): a linear boundary-value problem for the velocity potential is solved at each time-step and the nonlinear free-surface boundary conditions are used to update in time the geometry and the velocity potential.

(a) Mathematical formulation

The computational domain is shown in [figure 1](#). Equations for fully nonlinear potential flows with a free surface are summarized later. The fluid velocity is expressed as $\mathbf{u} = \nabla\phi$ with $\phi(x, y, z; t)$ the velocity potential. The continuity

equation in the fluid domain is Laplace's equation for the velocity potential:

$$\Delta\phi = 0. \quad (2.1)$$

The three-dimensional free space Green's function is defined as

$$G(\mathbf{x}, \mathbf{x}_l) = \frac{1}{4\pi|\mathbf{r}|}, \quad \frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{x}_l) = -\frac{1}{4\pi} \frac{\mathbf{r} \cdot \mathbf{n}}{|\mathbf{r}|^3}, \quad (2.2)$$

where $|\mathbf{r}| = |\mathbf{x} - \mathbf{x}_l|$ is the distance from the source point \mathbf{x} to the collocation point \mathbf{x}_l (both are on the boundary) and \mathbf{n} is the normal vector pointing out of the fluid. The notation $\partial G/\partial n$ represents the normal derivative, i.e. $\partial G/\partial n = \nabla G \cdot \mathbf{n}$. Green's second identity transforms Laplace's equation (2.1) into a BIE on the boundary $\Gamma = \Gamma_{r1} \cup \Gamma_{r2} \cup \Gamma_b \cup \Gamma_f$ of the fluid domain:

$$\alpha(\mathbf{x}_l)\phi(\mathbf{x}_l) = \int_{\Gamma(t)} \left\{ \frac{\partial \phi}{\partial n}(\mathbf{x})G(\mathbf{x}, \mathbf{x}_l) - \phi(\mathbf{x})\frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{x}_l) \right\} d\Gamma, \quad (2.3)$$

where $\alpha(\mathbf{x}_l)$ is proportional to the solid exterior angle made by the boundary at the collocation point \mathbf{x}_l .

The kinematic and dynamic boundary conditions on the free surface are written in a mixed Eulerian–Lagrangian form:

$$\frac{D\mathbf{x}}{Dt} = \nabla\phi, \quad (2.4)$$

$$\frac{D\phi}{Dt} = -gz + \frac{1}{2}\nabla\phi \cdot \nabla\phi, \quad (2.5)$$

where \mathbf{x} is the position vector of a fluid particle on the free surface, g the acceleration due to gravity and D/Dt the material derivative. The lateral boundary $\Gamma_{r1}(t)$ is either fixed or moving. In the first case, the potential is specified on the free surface in order to determine the initial perturbation. In the second case, waves are generated by a wave maker with specified motion and velocity. An absorbing piston can also be used at the opposite boundary. Along the fixed parts of the boundary, including the bottom that can be defined by an arbitrary shape, the no-flow condition $\partial\phi/\partial n = 0$ is prescribed.

(b) Time integration

There are several possibilities for time integration. For example, Xue *et al.* (2001) used a fourth-order Adams–Bashforth–Moulton integrator coupled with a fourth-order Runge–Kutta scheme. As in Grilli *et al.* (2001), second-order explicit Taylor series expansions are used to find the new position and the potential on the free surface at time $t + \delta t$. Zeroth-order coefficients are given by the geometry and the solution of the BIE (2.3) at time t . First-order coefficients are then directly obtained from the boundary conditions (2.4) and (2.5). The use of second-order terms leads to a better accuracy of the time scheme, but it requires more information than is available. Indeed, the corresponding coefficients depend on the spatial and temporal derivatives of the velocity potential. On one hand, tangential derivatives are computed by high-order interpolation on a sliding 5×5 grid (Fochesato *et al.* 2005). On the other hand, the pairs $(\partial\phi/\partial t, \partial^2\phi/\partial t \partial n)$ are computed by solving another integral equation

on the boundary similar to (2.3). There are therefore two different linear systems to solve at each time-step. In the following, we only describe the case of the velocity potential, since the numerical solution is identical for both.

The time-step δt is adapted at each time as a function of the minimum distance between two nodes on the free surface and a constant Courant number C_0 . [Grilli et al. \(2001\)](#) found an optimal value of roughly 0.45 for C_0 . The time-stepping scheme presents the advantage of being explicit, and the use of spatial derivatives along the free surface provides a better stability of the computed solution.

(c) *Spatial discretization*

The integral equations are solved by BEM. The boundary is discretized into N collocation nodes and M high-order elements are used to interpolate in between q of these nodes. Within each element, the boundary geometry and the field variables are discretized using polynomial shape functions $N_j(\xi, \eta)$, where (ξ, η) denote the intrinsic coordinates of the reference element:

$$\mathbf{x}(\xi, \eta) = \sum_{j=1}^q N_j(\xi, \eta) \mathbf{x}_j, \tag{2.6}$$

$$\phi(\xi, \eta) = \sum_{j=1}^q N_j(\xi, \eta) \phi_j, \quad \text{and} \quad \frac{\partial \phi}{\partial n}(\xi, \eta) = \sum_{j=1}^q N_j(\xi, \eta) \frac{\partial \phi_j}{\partial n}. \tag{2.7}$$

Here, $j = 1, \dots, q$ denotes a node within an element. The chosen elements are the so-called middle-interval-interpolation elements, which provide C_2 continuity in between elements. They are quadrilaterals with cubic shape functions defined using the four nodes of the element and also all the neighbouring nodes in each direction, for a total of $q=16$ nodes.

The integrals on the boundary are converted into a sum on the elements, each one being calculated on the reference element $\Gamma_{\xi, \eta}$. The curvilinear change of variables $\mathbf{x} \rightarrow (\xi, \eta)$ leads to a Jacobian matrix \mathbf{J}^i for the i th element. The discretized form of the integrals in (2.3) can thus be written as

$$\int_{\Gamma} \frac{\partial \phi}{\partial n}(\mathbf{x}) G(\mathbf{x}, \mathbf{x}_l) d\Gamma = \sum_{i=1}^M \left\{ \int_{\Gamma_{\xi, \eta}} \sum_{j=1}^q \frac{\partial \phi}{\partial n}(\mathbf{x}_j) N_j(\xi, \eta) G(\mathbf{x}(\xi, \eta), \mathbf{x}_l) |\mathbf{J}^i(\xi, \eta)| d\xi d\eta \right\},$$

$$\int_{\Gamma} \phi(\mathbf{x}) \frac{\partial G}{\partial n}(\mathbf{x}, \mathbf{x}_l) d\Gamma = \sum_{i=1}^M \left\{ \int_{\Gamma_{\xi, \eta}} \sum_{j=1}^q \phi(\mathbf{x}_j) N_j(\xi, \eta) \frac{\partial G}{\partial n}(\mathbf{x}(\xi, \eta), \mathbf{x}_l) |\mathbf{J}^i(\xi, \eta)| d\xi d\eta \right\}.$$

The associated discretized BIE leads to a sum on the N boundary nodes,

$$\alpha(\mathbf{x}_l) \phi(\mathbf{x}_l) = \sum_{j=1}^N \left(K_{lj}^D \frac{\partial \phi}{\partial n}(\mathbf{x}_j) - K_{lj}^N \phi(\mathbf{x}_j) \right), \tag{2.8}$$

where $l = 1, \dots, N$ and K_{lj}^D, K_{lj}^N are Dirichlet and Neumann global matrices, respectively. According to the boundary type on which lies the current node, the potential, or its normal derivative, is either specified by a boundary condition or is an unknown of the problem.

The matrices are built with the numerical computation of the integrals on the reference element. When the collocation node does not belong to the integrated element, a standard Gauss–Legendre quadrature method is used. When it belongs to the element, special methods are needed to take into account the singularities in the evaluation of Green’s functions. Some of the regular integrals can be nearly singular because of the accumulation of nodes. An adaptive technique that is based on recursive subdivisions is needed.

The matrices are then modified to take into consideration two specific aspects of the solution. First, the rigid mode technique is used to avoid a purely geometric calculation of solid angles at nodes of the discretized boundary. Besides, it improves the conditioning of the linear system. Numerically, it consists of imposing

$$\alpha(\mathbf{x}_l) + K_{ll}^N = - \sum_{j=1(\neq l)}^N K_{lj}^N, \quad l = 1, \dots, N, \tag{2.9}$$

so that diagonal terms are directly substituted in the discretized system. Finally, it avoids the computation of the singular diagonal coefficient of the Neumann matrix K_{ll}^N . The second point is related to edges. Indeed, boundary conditions and normal directions are in general different on intersecting parts of the boundary, such as the free surface (or the bottom) and lateral boundaries. Consequently, edges are represented by two or three numerical nodes in the model. Therefore, more equations are needed for each additional node. They are obtained from the continuity of the potential. Only the free surface has specified potential values, represented below with horizontal bars. On the other boundaries it is unknown:

- (i) $\phi(\mathbf{x}_{n_2}) = \overline{\phi(\mathbf{x}_{n_1})}$ if n_1 belongs to the free surface,
- (ii) $\phi(\mathbf{x}_{n_2}) = \phi(\mathbf{x}_{n_1})$ if n_1 belongs to lateral or bottom boundaries,

where n_1 is the main node number for which a BIE is solved (the main node is the one which appears first in the mesh numbering), and n_2 is either a double or a triple node number. The continuity of $\partial\phi/\partial n$ is imposed similarly. From the implementation point of view, lines in the matrix corresponding to the BIE for multiple nodes are replaced by these compatibility conditions.

Let Γ^D and Γ^N denote the boundaries with, respectively, Dirichlet and Neumann conditions. The linear system can finally be described as follows:

- (i) If $\mathbf{x}_l \in \Gamma^D$ and is a simple node or the first of multiple nodes,

$$\sum_{j \in \Gamma^N} K_{lj}^N \phi(\mathbf{x}_j) - \sum_{j \in \Gamma^D} K_{lj}^D \frac{\partial \phi}{\partial n}(\mathbf{x}_j) = \sum_{j \in \Gamma^N} K_{lj}^D \frac{\partial \bar{\phi}}{\partial n}(\mathbf{x}_j) - \sum_{j \neq l}^{j \in \Gamma^D} K_{lj}^N \bar{\phi}(\mathbf{x}_j) + \sum_{j \in \Gamma}^{j \neq l} K_{lj}^N \bar{\phi}(\mathbf{x}_l). \tag{2.10}$$

- (ii) If $\mathbf{x}_l \in \Gamma^N$ and is a simple node or the first of multiple nodes,

$$\sum_{j \in \Gamma^N}^{j \neq l} K_{lj}^N \phi(\mathbf{x}_j) - \sum_{j \in \Gamma}^{j \neq l} K_{lj}^N \phi(\mathbf{x}_l) - \sum_{j \in \Gamma^D} K_{lj}^D \frac{\partial \phi}{\partial n}(\mathbf{x}_j) = \sum_{j \in \Gamma^N} K_{lj}^D \frac{\partial \bar{\phi}}{\partial n}(\mathbf{x}_j) - \sum_{j \in \Gamma^D} K_{lj}^N \bar{\phi}(\mathbf{x}_j). \tag{2.11}$$

(iii) If \mathbf{x}_l is a multiple node (it is necessarily on Γ^N in our case),

$$\phi(\mathbf{x}_l) = \overline{\phi(\mathbf{x}_{\text{main}(l)})} \text{ if } \mathbf{x}_{\text{main}(l)} \in \Gamma^D, \quad \phi(\mathbf{x}_l) - \phi(\mathbf{x}_{\text{main}(l)}) = 0 \text{ if } \mathbf{x}_{\text{main}(l)} \in \Gamma^N,$$

where $\mathbf{x}_{\text{main}(l)}$ denotes the location of the main node number.

The linear systems resulting from the two BIE are full and non-symmetric. Assembling the matrix as well as performing the integrations accurately are time-consuming tasks. They are done only once at each time-step, since the same matrix is used for both systems. Solving the linear system is another time-consuming task. Even with the GMRES algorithm with preconditioning, the computational complexity is $O(N^2)$, which is the same as the complexity of the assembling phase.

This is not sufficient for large problems. In order to accelerate the computations, we inserted the FMA. First, it reduces directly the complexity of the problem to nearly $O(N)$. Second, the matrix is no longer built. Far away nodes are placed in groups, so less time is spent in numerical integrations and memory requirements are reduced. Third, the hierarchical structure involved in the algorithm automatically gives the distance criteria for adaptive integrations. Finally, it gives a way to parallelize the method. The last two are left for future work and are not treated here. However, a serial implementation in the existing numerical wave tank leads to significant improvements as shown in §5.

3. The fast multipole algorithm

The fast multipole algorithm (FMA) of Greengard & Rokhlin (1987) provides a way to compute all pairwise interactions in large sets of particles. Rather than describing precisely the algorithm, we briefly present below the main components of the method (Beatson & Greengard 1997). Let us consider a sum of the form

$$u(\mathbf{x}_i) = \sum_{j=1}^N K(\mathbf{x}_i, \mathbf{y}_j) w_j, \quad (3.1)$$

where \mathbf{x}_i and \mathbf{y}_j are points in \mathbb{R}^3 and K an interaction kernel. Direct evaluations of such sums at N target points \mathbf{x}_i require $O(N^2)$ operations. The FMA deals with interacting systems represented by a kernel K that can be expressed as a far field expansion in which the influence of source and evaluation points is separated. It is valid for a point \mathbf{O} close to \mathbf{y} and far from \mathbf{x} :

$$K(\mathbf{x}, \mathbf{y}) \approx \sum_{k=1}^p \phi_k(\mathbf{O}\mathbf{x}) \psi_k(\mathbf{O}\mathbf{y}), \quad (3.2)$$

where the notation $\mathbf{O}\mathbf{x}$ means $\mathbf{x} - \mathbf{O}$. Thus, $u(\mathbf{x}_i)$ can be computed in two steps. First, one computes the moments,

$$M_k(\mathbf{O}) = \sum_{j=1}^N \psi_k(\mathbf{O}\mathbf{y}_j) w_j. \quad (3.3)$$

Then, one evaluates $u(\mathbf{x}_i)$ at each given point \mathbf{x}_i via the formula

$$u(\mathbf{x}_i) \approx \sum_{k=1}^p M_k(\mathbf{O}) \phi_k(\mathbf{O}\mathbf{x}_i). \quad (3.4)$$

Consequently, the amount of required work is $O(Np)$, where p is the number of terms kept in the summation (3.2). This estimate is based on the far field expansions and is therefore related to only one part of the distribution of particles. For particles that are close to each other, direct computations of the interactions are necessary. This is the principle of the method.

However, in order to be really effective and achieve an almost linear complexity, this idea must be associated with an algorithmic treatment: a hierarchical subdivision of space. At the top level of the tree is a cube containing the whole set of particles. The cube is cut into eight children cubes to create the first level. The tree is built recursively until a number of levels is reached. Then, one can work mainly in terms of the cells of the octree rather than the particles. In fact, one considers the contributions of groups of particles belonging to a same cell. The regular partitioning automatically gives distance criteria to determine the cells for which it is possible to use the multipole expansions. Therefore, interaction lists are created for each cell. Interactions between well-separated pairs can then be computed by multipole expansions.

In that case, the machinery for far away particles must be initiated. The application of the above principle gives rise to an $O(N \log(N))$ implementation. In order to reduce the complexity even further, one has to apply another technique for the evaluation phase. Indeed the use of local expansions, similar to the multipole ones, makes it possible to place particles in groups for this evaluation phase. If we have an expansion for a point \mathbf{Q} close to \mathbf{x} ,

$$\phi_k(\mathbf{O}\mathbf{x}) \approx \sum_{l=1}^p \alpha_l(\mathbf{Q}\mathbf{x})\beta_l^k(\mathbf{O}\mathbf{Q}), \tag{3.5}$$

then the potential $u(\mathbf{x}_i)$ can be evaluated only with locally dependent terms:

$$u(\mathbf{x}_i) \approx \sum_{l=1}^p \alpha_l(\mathbf{Q}\mathbf{x}_i)L_l(\mathbf{Q}), \quad L_l(\mathbf{Q}) \approx \sum_{k=1}^p M_k(\mathbf{O})\beta_l^k(\mathbf{O}\mathbf{Q}). \tag{3.6}$$

In order to relate spatially both expansions, translation formulae are required. Each expansion depends on an origin which is the centre of the current cell. First, the moments (3.3) are computed at each cell of the finest level. Then, a translation formula allows to climb the tree, the multipole expansions being transported to the centre of the parent cell (\mathbf{O}'):

$$M_k(\mathbf{O}') = \sum_{l=1}^p M_l(\mathbf{O})\gamma_l^k(\mathbf{O}\mathbf{O}'), \tag{3.7}$$

where γ_l^k is obtained through the available expansion,

$$\psi_k(\mathbf{O}'\mathbf{y}) = \sum_{l=1}^p \psi_l(\mathbf{O}\mathbf{y})\gamma_l^k(\mathbf{O}\mathbf{O}'). \tag{3.8}$$

The contribution of a distant group of nodes to a local group is computed by the formula (3.6), which converts the multipole expansion into a local expansion L_l . The translation formula for local expansions allows to go down the tree. Back at the finest level, local expansions are evaluated giving the contribution of all distant nodes.

All these expansions depend on the kernel K . For Laplace’s equation, Green’s functions are expanded into spherical harmonics. It is important to note here that the FMA offers the advantage over other fast methods that any accuracy can be specified *a priori* through the choice of the number of terms p in the multipole approximation.

4. Application of the FMA to the BEM

The FMA could be used by itself to solve Laplace’s equation. But it is better to keep the BEM to discretize the problem since the accuracy of the BEM is well recognized. Then, the FMA comes as an additional approximation that makes the computations more efficient. What is difficult in the classical BEM, even with the GMRES algorithm, is to solve the full linear system that comes from the discretized integral equation involving Green’s function, and to assemble the resulting matrix. The idea is to replace every matrix–vector product in the GMRES algorithm by an evaluation with the FMA. This avoids assembling the matrix and reduces the complexity to nearly $O(N)$ (N is the number of nodes on the boundary of the domain).

(a) Expansions

As for the integration of Laplace’s equation, the application of the multipole algorithm in our case is based on expansions of Green’s function into spherical harmonics. Given an origin \mathbf{O} , a point \mathbf{x}_l far away from the origin and a source point \mathbf{x} near the origin such that $\mathbf{O}\mathbf{x} = (\rho, \alpha, \beta)$ and $\mathbf{O}\mathbf{x}_l = (r, \theta, \phi)$ in spherical coordinates, Green’s function can be expanded as

$$G(\mathbf{x}, \mathbf{x}_l) \approx \frac{1}{4\pi} \sum_{k=0}^p \sum_{m=-k}^k \rho^k Y_k^{-m}(\alpha, \beta) \frac{Y_k^m(\theta, \phi)}{r^{k+1}}. \tag{4.1}$$

Here, the functions $Y_k^{\pm m}$ are the spherical harmonic polynomials (omitting a scaling factor as in Greengard 1988),

$$Y_k^{-m}(\alpha, \beta) = \sqrt{\frac{(k - |m|)!}{(k + |m|)!}} P_k^{|m|}(\cos \alpha) e^{-im\beta}, \quad k \geq 0, \quad |m| \leq k, \tag{4.2}$$

defined by the associated Legendre polynomials,

$$P_k^m(x) = \frac{(-1)^m}{2^k k!} (1 - x^2)^{m/2} \frac{d^{k+m}(x^2 - 1)^k}{dx^{k+m}}, \quad k \geq 0, \quad 0 \leq m \leq k. \tag{4.3}$$

One has an expansion similar to (4.1) for $\partial G / \partial n(\mathbf{x}, \mathbf{x}_l)$. Consequently, the integral equation (2.3) occurring in the surface wave model for the pairs

$(\phi, \partial\phi/\partial n)$ (the pairs $(\partial\phi/\partial t, \partial^2\phi/\partial t \partial n)$ are treated similarly) becomes

$$\alpha(\mathbf{x}_i)\phi(\mathbf{x}_i) \approx \int_{\Gamma} \left\{ \frac{\partial\phi}{\partial n}(\mathbf{x}) \frac{1}{4\pi} \sum_{k=0}^p \sum_{m=-k}^k \rho^k Y_k^{-m}(\alpha, \beta) \frac{Y_k^m(\theta, \phi)}{r^{k+1}} - \phi(\mathbf{x}) \frac{1}{4\pi} \sum_{k=0}^p \sum_{m=-k}^k \frac{\partial}{\partial n} (\rho^k Y_k^{-m}(\alpha, \beta)) \frac{Y_k^m(\theta, \phi)}{r^{k+1}} \right\} d\Gamma. \quad (4.4)$$

The double sum can be taken outside of the integral so that

$$\alpha(\mathbf{x}_i)\phi(\mathbf{x}_i) \approx \frac{1}{4\pi} \sum_{k=0}^p \sum_{m=-k}^k M_k^m(\mathbf{O}) \frac{Y_k^m(\theta, \phi)}{r^{k+1}}, \quad (4.5)$$

where $M_k^m(\mathbf{O})$ is the moment at \mathbf{O} :

$$M_k^m(\mathbf{O}) = \int_{\Gamma} \left\{ \frac{\partial\phi}{\partial n}(\mathbf{x}) \rho^k Y_k^{-m}(\alpha, \beta) - \phi(\mathbf{x}) \frac{\partial}{\partial n} (\rho^k Y_k^{-m}(\alpha, \beta)) \right\} d\Gamma. \quad (4.6)$$

The expansions are valid at a given origin \mathbf{O} that represents the point at which the contributions are placed together. In the algorithm, it is the centre of the cell in which the expansion is known to be valid.

(b) *Near interactions*

For the cells that are near neighbours, the programme calls a routine for the BEM analysis. The integrations are the same as in the former model, except that they are not performed at each collocation node. The integrations on each element of the cells belonging to this interaction list are added and evaluated at each collocation node.

(c) *Far away interactions*

First, a BEM analysis has to be performed to compute the moments

$$M_k^m(\mathbf{O}) = \sum_{i=1}^M \left\{ \int_{\Gamma_{\xi, \eta}} \sum_{j=1}^q \frac{\partial\phi}{\partial n}(\mathbf{x}_j) N_j(\xi, \eta) \rho^k Y_k^{-m}(\alpha, \beta) |\mathbf{J}^i(\xi, \eta)| d\xi d\eta \right\} - \sum_{i=1}^M \left\{ \int_{\Gamma_{\xi, \eta}} \sum_{j=1}^q \phi(\mathbf{x}_j) N_j(\xi, \eta) \frac{\partial}{\partial n} (\rho^k Y_k^{-m}(\alpha, \beta)) |\mathbf{J}^i(\xi, \eta)| d\xi d\eta \right\}. \quad (4.7)$$

In this case, $\rho^k Y_k^{-m}(\alpha, \beta)$ plays the role of $G(\mathbf{x}, \mathbf{x}_i)$. But the numerical integrations are similar. The main question is to know whether there is a singularity. Unlike Green's function, the spherical harmonics are not singular. However, the normal derivative of the spherical harmonics brings in an apparent singularity that could generate numerical errors. [Fochesato \(2004\)](#) showed that this singularity can be removed by writing the spherical harmonics differently. The boundary-element discretization only appears in the computation of the moments. Therefore, the translation of a multipole expansion, the conversion to a local expansion, the translation of a local expansion and the local expansion itself are all the same as in the original work by [Greengard \(1988\)](#) (see [Fochesato 2004](#) for more details).

(d) *Storage and specific points*

All the ingredients of the former numerical code that were related to the discretization matrix must be adapted. For example, the storage of coefficients that are needed for multiple use is now done cell by cell. The rigid mode technique was applied to the former model in order to compute the factors $\alpha(\mathbf{x}_l)$ of the BIE. Recall that it improves the conditioning of the system and avoids the integration of the singular diagonal coefficient of the Neumann matrix. Since the matrix is no longer available, the implementation must be adapted. In the former model, the technique was applied *a priori* by substituting lines according to equation (2.9). Now we apply it after the matrix–vector product, by correcting the resulting vector. The sum of the off-diagonal terms is seen as a matrix–vector product with the constant vector composed of ones. Then the vector is multiplied by $\phi(\mathbf{x}_l)$, which is either a specified value (Dirichlet condition) or an estimated value (Neumann condition) since an iterative method is used. Therefore, the rigid mode technique appears in the linear systems (2.10) and (2.11) as

$$+ \left(\sum_{j \in \Gamma}^{j \neq l} K_{lj}^N \times 1 \right) \bar{\phi}(\mathbf{x}_l) \text{ on the RHS,} \quad - \left(\sum_{j \in \Gamma}^{j \neq l} K_{lj}^N \times 1 \right) \phi(\mathbf{x}_l) \text{ on the LHS,}$$

depending on the location of the node \mathbf{x}_l . The new product is performed by the fast algorithm, using the stored Neumann coefficients as seen earlier.

The treatment of the edges also led to a modification of the coefficient matrix. The implementation of double and triple nodes was done *a priori*, replacing lines by the compatibility conditions. In the adapted fast BEM, these continuity relations impose values of the resulting vector. So, once the solution of the matrix–vector product is obtained by the FMA, a routine explicitly acts on the resulting vector to force the continuity of the potential at double/triple nodes.

Finally, note that the GMRES algorithm, which is partly at the intersection between the BEM and the FMA, is necessarily adapted. First, calls to a specific routine replace the matrix–vector products. But since the matrix is no longer available, the preconditioning must be adapted too. SSOR (symmetric successive over-relaxation) preconditioning is still used but the matrix is different. The FMA is well suited to a geometric strategy for the sparse approximate matrix (Carpentieri *et al.* 2001). The near interacting coefficients of the matrix are still available and can be used for that purpose. Therefore, we store these coefficients in a sparse matrix during the near interaction phase in order to use the former preconditioning.

5. Results

Preliminary results obtained with the original model without the FMA were given by Grilli *et al.* (2001). The original numerical wave tank was further validated by Grilli *et al.* (2002) and Guyenne & Grilli (2006), with an emphasis on physical implications of the results. Grilli *et al.* (2002) compared favourably model results on tsunami generation by underwater landslides to laboratory experiments, while Guyenne & Grilli (2006) investigated the surface and internal kinematics of the shoaling and overturning of solitary waves. In this section, results obtained with the new algorithm are compared with those obtained with

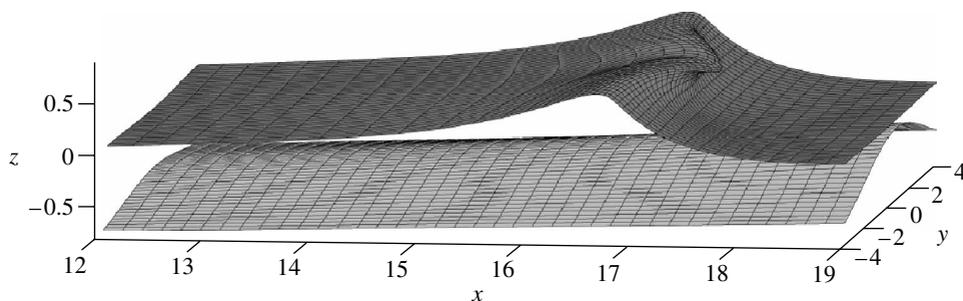


Figure 2. Overturning of a steep solitary wave with a plunging jet. The solitary wave comes from the far field and propagates over a sloping ridge. The discretization used in the three-dimensional numerical wave tank is $60 \times 40 \times 4$ elements, respectively, along the x , y and vertical directions (i.e. 6022 nodes).

the classical BEM. New results are presented as well. Our aim is to show that the same global accuracy is achieved with much less computing time.

The application considered by Grilli *et al.* (2001) is used first to perform the comparative study. It involves the overturning of a solitary wave over a sloping bottom with a transverse modulation that focuses the energy and leads to a plunging jet. The model has been shown to reproduce with high accuracy the initiation of the breaking jet. Therefore, it is natural to use that example to test the accelerated method. The maximum depth h_{\max} is used as unit length. The unit time is $\sqrt{h_{\max}/g}$. The numerical wave tank is of length 19 and width 4 (or 8). It is bounded by solid boundaries. The initial wave is a steep solitary wave of amplitude 0.6. Figure 2 shows a zoom of the wave overturning on the three-dimensional bottom. The accuracy of the new solution is then compared to that of the former solution (without FMA). Moreover, the global stability is monitored in order to get the smallest number of multipole terms that is needed to get the same breaking jet as before. The computing times are also reported for several grids. The computations have been realized on an Intel Pentium 4 processor. Its main characteristics are 2.2 GHz for the CPU and 1 Go of memory.

For the application under consideration, the various parameters of the BEM code are fixed. The number of integration points is 10 by direction in order to ensure sufficient accuracy to obtain the breaking jet. Two parameters of the FMA must be specified with great care: the degree of the multipole expansion and the number of levels in the hierarchical subdivision. The first one determines the accuracy of the new approximation, but also influences the computing time. The second one mainly acts on the efficiency. One expects the algorithm to be more efficient for more subdivided domains. But then the manipulation of the tree structure becomes a big task and reduces the benefits of the algorithm. For most of our computations, five levels turned out to be optimal. For very coarse meshes, four levels were used and for larger systems, a sixth or higher level was required.

(a) Accuracy

As shown in §4, the FMA is based on truncated multipole expansions. They are another approximation in the model and the first comparison consists in verifying that the new results approach the former ones when the number of

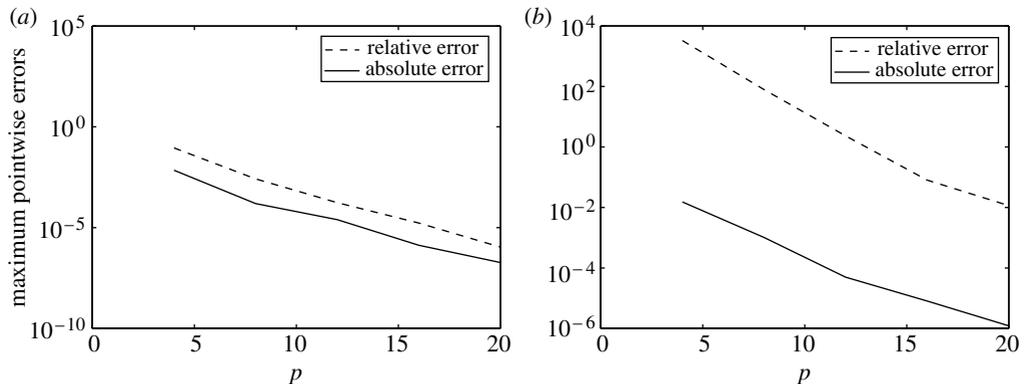


Figure 3. Maximum errors between the solutions with and without using the FMA versus the number of terms p in the multipole expansions. (a) 150 nodes, (b) 1422 nodes.

multipole terms increases. For that purpose we compute the first linear system for two grids (i.e. two different levels of accuracy). The first one is a very coarse grid with four elements in each direction ($4 \times 4 \times 4$ grid) for a total of 150 nodes. Then, a more realistic grid is used: $40 \times 10 \times 4$ with 1422 nodes. The consistency of the multipole approximation inside the numerical wave tank is checked by comparing point by point the solution obtained with both methods. The maximum relative and absolute errors are plotted as a function of the number of multipole terms in figure 3.

In each case, we get an almost straight line on a logarithmic scale; thus the convergence is nearly exponential. However, we note that the relative errors are greater than the absolute ones, particularly for the finer grid. The new approximation deals with far away interactions. That way, small coefficients of the influence matrix are approximated with less accuracy than the larger ones. Consequently, small values which are very close to zero lead to large relative errors. Only absolute errors are relevant here, because the values of interest are of order unity and we are not specifically aiming at getting high accuracy in the tail of the solitary wave where the values are close to zero. This confirms the consistency of the accelerated method with regard to the classical one.

The fast algorithm comes as an additional approximation in the model. What is important is not only that the accelerated method matches the former one, but also that the global accuracy and numerical stability properties remain the same. The only available diagnostic tool consists of following the evolution of volume and energy as a function of time. Indeed these quantities must be constant since there are no sources or sinks. The relative errors with regard to their initial value give a good indication on the accuracy and the stability of the computed solution. Big errors or oscillatory variations must lead to a loss in accuracy.

As shown in figure 2, the overturning of a solitary wave is simulated on a $60 \times 40 \times 4$ grid and eight multipole terms are used in the FMA. Figure 4 shows the evolution of the volume and energy relative errors for both methods. The curves are very close. The computations are done in two steps. First, the shoaling of the solitary wave is obtained on the whole domain. Then, a regular regridding is applied at $t \approx 6$ on a reduced domain around the wave. That way, the spatial discretization is finer and is allowed to go further in the development of the

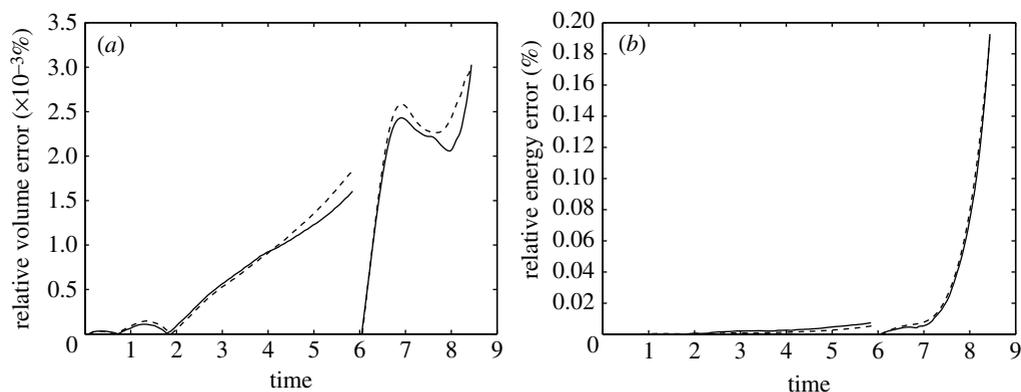


Figure 4. Relative numerical errors (%) on (a) volume versus time, (b) energy versus time, for both methods (with FMA in solid line and without FMA in dashed line). The discretization is $60 \times 40 \times 4$ elements (6022 nodes), and $p=8$ in the FMA. The jumps are due to regriding.

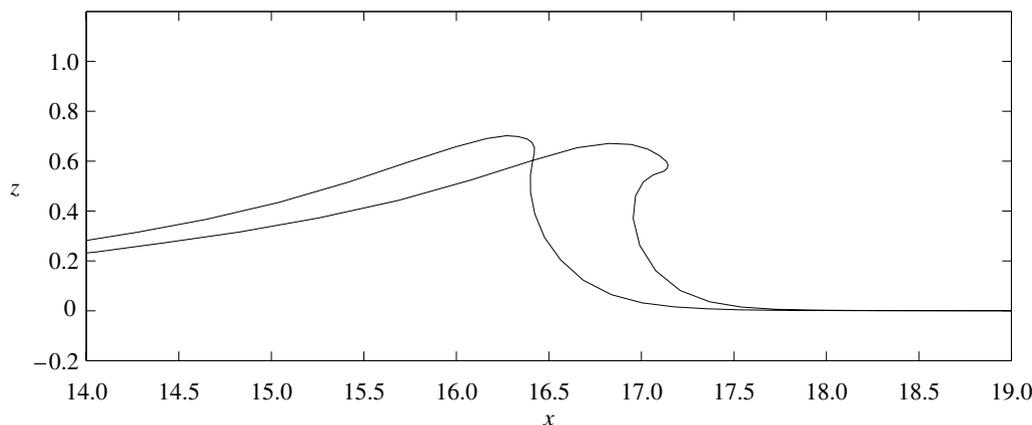


Figure 5. Profiles of the free surface in the middle of the wave tank ($y=0$) at two different times. The results of both methods (with FMA in solid line and without FMA in dashed line) are shown. There is no difference within graphical accuracy. The grid is $60 \times 40 \times 4$ and the value $p=8$ is used in the FMA.

plunging jet. This regriding explains the jump in the volume and energy curves. The gradual increase of these errors corresponds to the formation of the plunging wave, which has propagated along the basin. To confirm this conclusion, figure 5 shows the slices in the middle of the tank for both solutions at two different times, after the regriding. Within graphical accuracy, there is no difference between them.

Since the accelerated model is able to accurately simulate highly nonlinear wave phenomena, let us now consider the efficiency. The first question to answer is whether one can get such solutions with fewer multipole terms. That is to say, once the accuracy of the BEM code is given, what is the smallest value for p which leads to the same global accuracy? For this $60 \times 40 \times 4$ grid, one can keep only six terms in the multipole expansions instead of eight. The errors on the

Table 1. Comparison of computing time with and without the use of the fast multipole algorithm as a function of the number of nodes. (The value $p=8$ is used in the FMA.)

grid	number of nodes	time with FMA (s)	time without FMA (s)
$40 \times 10 \times 4$	1422	45	41
$50 \times 20 \times 4$	2862	120	390
$50 \times 40 \times 4$	5102	250	1258
$60 \times 40 \times 4$	6022	329	1852

volume and the energy are of the same order but they do not follow exactly the same curves as in figure 4. The plunging jet is obtained, but there is a difference in the wave elevation and the back of the wave is slightly oscillatory. With finer grids, the accuracy of the BEM is improved. If we keep the same parameter values for the FMA, the errors due to the multipole approximation may become dominant. So it may be necessary to modify them in order to get the same global accuracy.

(b) Efficiency

Suppose that the best parameter set for the FMA is known. The computing times for various discretizations are recorded in table 1. The comparisons are made for the computations until the first linear system is solved. The values for one time-step are not exactly the same. Indeed, if the computing time for one time-step is measured as the ratio between the total computing time and the number of time-steps, one obtains for instance 49 s per time-step with the FMA versus 62 s without for the grid with 1422 nodes, and 320 versus 1835 s for the grid with 6022 nodes. So even for the coarse grid with 1422 nodes, the FMA gives better values. For the $60 \times 40 \times 4$ grid with 6022 nodes, a reduction of the computing time by a factor close to six is obtained.

These results have been obtained for some prescribed values of the BEM parameters, such as the number of integration points. Using less integration points reduces the time spent in the regular integrations. This is true for both methods but, since with the FMA less integrals are computed, the difference in time is less significant. However, the reduction of the complexity is conserved. For any value of the integration parameters, it is almost linear for large N (see figure 6 where the computing times are plotted versus the number of nodes for five integration points per direction instead of 10). The complexity of the accelerated method becomes linear above roughly 4000 nodes.

(c) Discussion and future improvements

The memory requirements of the former BEM code make it practically impossible to increase further the number of nodes to discretize the boundary. Thus the comparison stops here. However, the new model avoids the storage of most of the $N \times N$ matrices. In fact, the only one that remains is the preconditioning matrix. It is a sparse matrix and an adapted storage is under investigation. Anyhow more memory is available than before to do computations over larger grids.

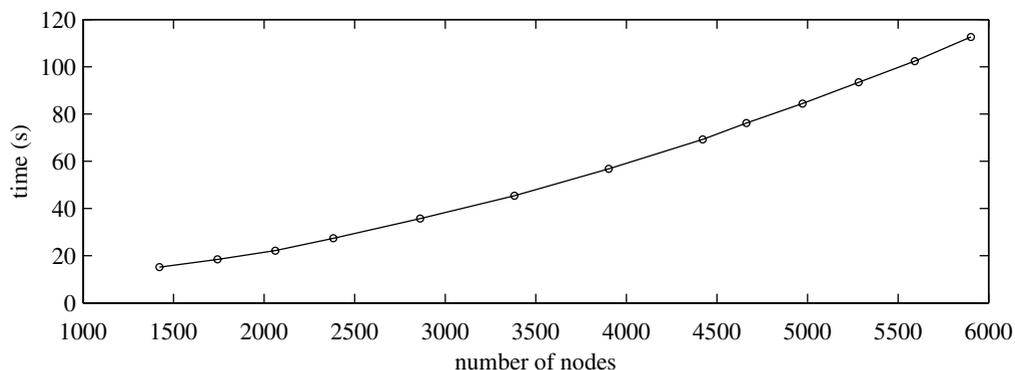


Figure 6. Evolution of the computing time versus the number of nodes.

The error analysis of the classical FMA is well known. But to our knowledge, a rigorous analysis of the BEM for a mixed Dirichlet–Neumann BIE with a piecewise smooth boundary has yet to be given (see [Beale *et al.* 1996](#); [Beale 2000](#); [Hou & Zhang 2002](#) for convergence results on BIEM for time-dependent, three-dimensional, doubly periodic water waves). The use of particular techniques makes the analysis even more complicated. So it is not appropriate to use the analysis of the FMA for the choice of the number of multipole terms, since we have no access *a priori* to the error made in the BEM approximation. We must apply empirical techniques that are application dependent. However, it will be interesting to be able to adjust each successive approximation to the global order of accuracy of the method: this will reduce the computing time for the desired accuracy.

Examples of computations performed with the new model are shown in [figures 7 and 8](#). The motivation for the example shown in [figure 8](#) was given by the Gold Coast artificial reef designed for coastal protection and surfing ([Black 2000](#)). All the computations presented in this paper correspond to scalar implementations. In fact, the old model has a vectorized implementation for which the computing time is reduced on a supercomputer. The new model could also be vectorized, but since the set of nodes has been divided into cells, the vector operations concern smaller arrays. Therefore, a vectorized implementation would be interesting only for a sufficiently large number of nodes by cell.

The FMA has been constantly improved since its discovery. But it is not obvious that much further gain can be expected for the level of accuracy needed in the numerical wave tank. However, a parallel implementation can bring a significant speed-up for any grid.

6. Conclusion

The three-dimensional numerical wave tank developed by [Grilli *et al.* \(2001\)](#) and validated by [Guyenne & Grilli \(2006\)](#) solves potential flow equations with a free-surface, and can describe accurately physical aspects of nonlinear waves over a complex bottom topography and wave focusing. However, more complex phenomena require finer and finer discretizations so that the $O(N^2)$ complexity of the code, where N is the number of nodes on the boundary of the domain, is too

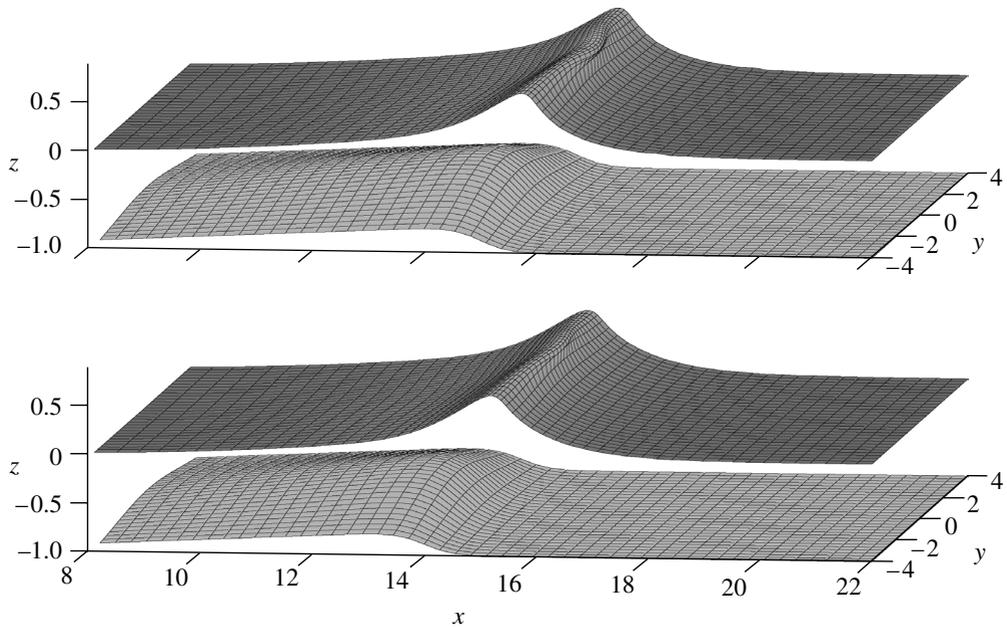


Figure 7. The shoaling of a solitary wave on a slope that stops abruptly leads to various types of breaking depending on the location of the bar. The upper panel shows a plunging wave with less inertia than in the case of a regular slope (see figure 2). For a bar located at a smaller value of x (lower panel), one gets instead an overturning wave with a very small jet. It is nearly a spilling wave. The discretizations are $80 \times 30 \times 4$ elements for 6142 nodes (335 s per time-step).

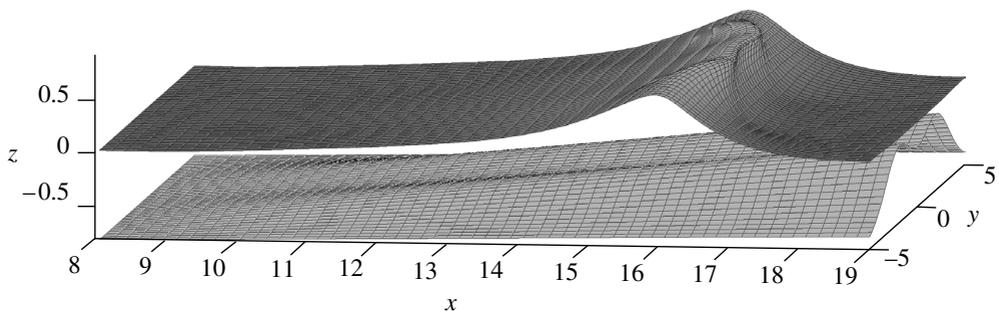


Figure 8. Overturning of a solitary wave over a sloping bottom with two humps in the transverse direction. As compared to the single hump case (figure 2), more elements are needed in the y -direction. Here, $60 \times 50 \times 4$ elements are used for 7342 nodes (400 s per time-step).

restrictive. To relax this constraint, the FMA has been combined with the BEM through the use of an iterative solver. This rapid algorithm has been designed to compute faster the mutual interactions of a set of particles. For far away particles, contributions can be grouped together through the use of multipole expansions. Moreover, a hierarchical subdivision of space leads to a real reduction of the complexity of the problem. The inclusion of the algorithm in the numerical wave tank means replacing each matrix–vector product by a call to the FMA. However, this is a non-trivial task and several failures have been reported in the past. The BEM analysis for well-separated nodes leads to new

integrals for which great care is taken in order to avoid apparent singularities. Particular features of the model must also be taken into account for the implementation, such as the rigid mode technique, the double nodes and the need to store some information that was included in the influence matrix. The resulting combined methods give a more efficient model even for coarse discretizations. Several improvements are still possible. The available distance criteria from the hierarchical subdivision can be used to apply adaptive integrations. Since the accurate numerical integrations are the main time-consuming task, appropriate choices of the number of nodes for the quadrature according to the distance between nodes should improve the method even further. Finally, even though the FMA is particularly efficient for a large number of nodes, the use of a parallel version will be a valuable improvement for any size of discretization. That way, finer grids and larger domains could be considered to give new insights into free-surface problems.

The authors would like to thank S. Grilli for valuable discussions on this topic. The first author acknowledges support from the Délégation Générale de l'Armement (DGA). The computations were performed in part on computers located at IDRIS (Institut du Développement et des Ressources en Informatique Scientifique, France), and were funded by the CNRS (Centre National de la Recherche Scientifique).

References

- Alpert, B., Beylkin, G., Coifman, R. & Rokhlin, V. 1993 Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Sci. Comput.* **14**, 159–184. (doi:10.1137/0914010)
- Anderson, C. R. 1992 An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comput.* **13**, 923–947. (doi:10.1137/0913055)
- Bateman, W. J. D., Swan, C. & Taylor, P. H. 2001 On the efficient numerical simulation of directionally spread surface water waves. *J. Comput. Phys.* **174**, 277–305. (doi:10.1006/jcph.2001.6906)
- Beale, J. T. 2000 A convergent boundary integral method for three-dimensional water waves. *Math. Comput.* **70**, 977–1029.
- Beale, J. T., Hou, T. Y. & Lowengrub, J. S. 1996 Convergence of a boundary integral method for water waves. *SIAM J. Numer. Anal.* **33**, 1797–1843. (doi:10.1137/S0036142993245750)
- Beatson, R. K. & Greengard, L. 1997 A short course on fast multipole methods. In *Wavelets, multilevel methods and elliptic PDEs* (ed. M. Ainsworth, J. Levesley, W. Light & M. Marletta), pp. 1–37. New York: Oxford University Press.
- Black, K. 2000 Artificial surfing reefs for erosion control and amenity: theory and application. In *Proc. Int. Coastal Symposium (ICS 2000), Rotorua, New Zealand, 2000*.
- Broeze, J., van Daalen, E. F. G. & Zandbergen, P. J. 1993 A three-dimensional panel method for nonlinear free surface waves on vector computers. *J. Comput. Mech.* **13**, 12–28.
- Brown, M. J., Mammoli, A. A. & Ingber, M. S. 2003 Parallel multipole implementation of the generalized Helmholtz decomposition for solving viscous flow problems. *Int. J. Numer. Methods Eng.* **58**, 1617–1635. (doi:10.1002/nme.830)
- Carpentieri, B., Duff, I. S. & Giraud, L. 2001 Robust preconditioning of dense problems from electromagnetics. In *Numerical analysis and its applications*. Springer Lecture Notes in Computer Science, no. 1988, pp. 170–178. Berlin: Springer.
- Cheng, H., Greengard, L. & Rokhlin, V. 1999 A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.* **155**, 468–498. (doi:10.1006/jcph.1999.6355)
- Darve, E. 2000 The fast multipole method: numerical implementation. *J. Comput. Phys.* **160**, 195–240. (doi:10.1006/jcph.2000.6451)

- Dias, F. & Bridges, T. J. In press. The numerical computation of freely propagating time-dependent irrotational water waves. *Fluid Dyn. Res.*
- Fochesato, C. 2004 Modèles numériques pour les vagues et les ondes internes. Ph.D. thesis, Ecole Normale Supérieure de Cachan, France.
- Fochesato, C., Grilli, S. & Guyenne, P. 2005 Note on non-orthogonality of local curvilinear coordinates in a three-dimensional boundary element method. *Int. J. Numer. Methods Fluids* **48**, 305–324. (doi:10.1002/fld.838)
- Fructus, D., Clamond, D., Grue, J. & Kristiansen, Ø. 2005 An efficient model for three-dimensional surface wave simulations. Part I. Free space problems. *J. Comput. Phys.* **205**, 665–685. (doi:10.1016/j.jcp.2004.11.027)
- Fuhrman, D. R., Madsen, P. A. & Bingham, H. B. 2004 A numerical study of crescent waves. *J. Fluid Mech.* **513**, 309–341. (doi:10.1017/S0022112004000060)
- Gómez, J. E. & Powert, H. 1997 A multipole direct and indirect BEM for 2D cavity flow at low Reynolds number. *Eng. Anal. Bound. Elem.* **19**, 17–31. (doi:10.1016/S0955-7997(97)00021-0)
- Goreinov, S. A., Tyrtshnikov, E. E. & Yeregin, A. Yu. 1997 Matrix-free iteration strategies for large dense linear systems. *Numer. Linear Algebra Appl.* **4**, 273–294. (doi:10.1002/(SICI)1099-1506(199707/08)4:4<273::AID-NLA97>3.0.CO;2-T)
- Gramma, A., Kumar, V. & Sameh, A. 1999 Parallel hierarchical solvers and preconditioners for boundary element methods. *SIAM J. Sci. Comput.* **20**, 337–358. (doi:10.1137/S1064827596313322)
- Graziani, G. & Landrini, M. 1999 Application of multipoles expansion technique to two-dimensional nonlinear free-surface flows. *J. Ship Res.* **43**, 1–12.
- Greengard, L. 1988 *The rapid evaluation of potential fields in particle systems*. Cambridge, MA: MIT Press.
- Greengard, L. & Rokhlin, V. 1987 A fast algorithm for particle simulations. *J. Comput. Phys.* **73**, 325–348. (doi:10.1016/0021-9991(87)90140-9)
- Grilli, S., Guyenne, P. & Dias, F. 2001 A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom. *Int. J. Numer. Methods Fluids* **35**, 829–867. (doi:10.1002/1097-0363(20010415)35:7<829::AID-FLD115>3.0.CO;2-2)
- Grilli, S., Vogelmann, S. & Watts, P. 2002 Development of a 3D numerical wave tank for modeling tsunami generation by underwater landslides. *Eng. Anal. Bound. Elem.* **26**, 301–313. (doi:10.1016/S0955-7997(01)00113-8)
- Guyenne, P. & Grilli, S. T. 2006 Numerical study of three-dimensional overturning waves in shallow water. *J. Fluid Mech.* **547**, 361–388. (doi:10.1017/S0022112005007317)
- Hackbush, W. 1999 A sparse matrix arithmetic based on \mathcal{H} -matrices. I. Introduction to \mathcal{H} -matrices. *Computing* **62**, 89–108. (doi:10.1007/s006070050015)
- Hou, T. Y. & Zhang, P. 2002 Convergence of a boundary integral method for 3-D water waves. *Discret. Cont. Dyn. Syst. B* **2**, 1–34.
- Korsmeyer, F. T., Yue, D. K. P. & Nabors, K. 1993 Multipole-accelerated preconditioned iterative methods for three-dimensional potential problems. In *Proc. BEM. 15, Worcester, MA, USA, 1993*.
- Kropinski, M. C. A. 2001 An efficient numerical method for studying interfacial motion in two-dimensional creeping flows. *J. Comput. Phys.* **171**, 479–508. (doi:10.1006/jcph.2001.6787)
- Longuet-Higgins, M. S. & Cokelet, E. D. 1976 The deformation of steep surface waves on water. I. A numerical method of computation. *Proc. R. Soc. A* **350**, 1–26.
- Mammoli, A. A. & Ingber, M. S. 2000 Parallel multipole BEM simulation of two-dimensional suspension flows. *Eng. Anal. Bound. Elem.* **24**, 65–73. (doi:10.1016/S0955-7997(99)00039-9)
- Nishimura, N. 2002 Fast multipole accelerated boundary integral equation methods. *Appl. Mech. Rev.* **55**, 299–324. (doi:10.1115/1.1482087)
- Potts, D. & Steidl, G. 2003 Fast summation at nonequispaced knots by NFFTs. *SIAM J. Sci. Comput.* **24**, 2013–2037. (doi:10.1137/S1064827502400984)
- Pringle, G. 1994 Numerical study of three-dimensional flow using fast parallel particle algorithms. Ph.D. thesis, Napier University, Edinburgh.

- Rokhlin, V. 1985 Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.* **60**, 187–207. (doi:10.1016/0021-9991(85)90002-6)
- Romate, J. E. & Zandbergen, P. J. 1989 Boundary integral equation formulations for free-surface flow problems in two and three dimensions. *Comput. Mech.* **4**, 267–282. (doi:10.1007/BF00301385)
- Scorpio, S. & Beck, F. 1996 A multipole accelerated desingularized method for computing nonlinear wave forces on bodies. In *Proc. 15th Int. Conf. on Offshore Mechanics and Arctic Engineering, Florence, Italy, 1996*.
- Wang, P., Yao, Y. & Tulin, M. P. 1995 An efficient numerical wave tank for nonlinear water waves, based on the multi-subdomain approach with BEM. *Int. J. Numer. Methods Fluids* **20**, 1315–1336. (doi:10.1002/flid.1650201203)
- Xue, M., Xü, H., Liu, Y. & Yue, D. K. P. 2001 Computations of fully nonlinear three-dimensional wave–wave and wave–body interactions. Part 1. Dynamics of steep three-dimensional waves. *J. Fluid Mech.* **438**, 11–39. (doi:10.1017/S0022112001004396)