An implementation of the Fast Multipole Algorithm for wave interaction problem on sparse arrays of floating bodies

Bruno Borgarino, A. Babarit, Pierre Ferrant

An implementation of the fast multipole algorithm for wave interaction problems on sparse arrays of floatin bodies

Bruno Borgarino · Aurelien Babarit · Pierre Ferrant

Abstract This paper describes an implementation of the fast multipole algorithm using the free-surface Green's function for ocean water waves. Its aim is to investigate different parameters of the fast multipole algorithm in order to efficientl carry out computations on sets of unknowns that are very inhomogeneously distributed in space. Some limits of the algorithm for this specifi case are pointed out. Those limits are essentially due to slow convergence of the multipole expansion of the Green's function. Eventually, a simplifie algorithm for this specifi application is described. The performance of the different algorithms is evaluated based on the computational time they require.

Keywords Fast multipole algorithm · Free-surface Green's function · Multipole expansion · Sparse arrays · Wave energy conversion

1 Introduction

The origin of this study is the need for modeling large arrays of wave energy converters of point absorber type using boundary element methods (BEM). Point absorbers are floatin bodies whose dimensions are small compared with the incident wavelength. The motions of these bodies in waves can be calculated by computing the flui velocity potential on their surfaces. However, the solution of such problems on arrays of numerous bodies is so far a numerical bottleneck. Indeed, the surface of each point absorber is typically modeled by a few hundred fla panels. As the industrial fiel of wave energy is still in development, an array would be made of 10–50 wave energy converters. This leads to large numbers of panels, meaning that very large linear systems (involving full, nonsymmetric matrixes) have to be solved. For each wave period within the range of interest, several problems need to be solved considering the whole array: up to six radiation problems per body (one for each degree of freedom), and one diffraction problem per incident wave direction. To perform these many large simulations, acceleration methods are mandatory.

The fast multipole algorithm [1] appears to be a perfectly adapted solution for such study, as its working principle is to compute interactions between well-separated groups of particles (in the following, the term "particles" refers to the centers of the panels discretizing the wet surfaces of the floatin bodies). The implementation of a fast

B. Borgarino (✉) · A. Babarit · P. Ferrant
LMF, Ecole Centrale de Nantes, 1 rue de la Noë, Nantes CEDEX 03, France
e-mail: bruno.borgarino@ec-nantes.fr

multipole algorithm into an iterative solver permits the complexity of the solution to be diminished from $O(N_{\text{panels}}^2)$ to $O(N_{\text{panels}})$.

It is worth mentioning a method based on the idea of grouping the influenc of the floatin bodies (instead of grouping the particles) developed in [2]. In this reference, the interaction method described in [3] is applied to very large arrays of floatin cylinders. A hierarchical method resembling the FMA is used to group the influenc of several bodies close to each other, translate it, and distribute it to the target bodies. This method thus uses BEM for computing the hydrodynamic characteristics of a single body. In the present paper, however, the objective is to apply the BEM directly to the whole array, such computations needing to be accelerated by the FMA.

Regarded as one of the ten most important algorithms of the 20th century, the FMA has been widely used for study of gravitational, electrostatic, and electromagnetic interactions in many-particle systems, although applications in the hydrodynamics fiel remain marginal. In [4], Fochesato et al. used the FMA in a numerical wave tank to study the propagation of fully nonlinear waves over a complex bathymetry. This method meshes the boundaries of the problem and thus uses simple Rankin sources (with Green's function $G = 1/r$). Other methods only mesh the wet surfaces of the floatin bodies, using specifi Green's functions respecting the boundary conditions on the free surface and on the seabed [5,6]. These functions have to be expressed in the multipole expansion formulation to be used in the FMA.

For the constant depth case, the Green's function is described as series of terms containing the modifie Bessel function of the second kind K_0 [5]. Using Graf's addition theorem, the multipole expansion has been derived by [7]. Combining a higher-order boundary element method (HOBEM) and a FMA using this expansion, the hydrodynamic responses of a very large floatin structure (VLFS) have been investigated. In [8], the results of the combination of the constant panel method (CPM) or the HOBEM and the FMA are compared with analytical solutions for a floatin box and a floatin cylinder. In [9], the hydrodynamic interactions between three closely spaced ships have been studied. Recently, an expansion for the free-surface Green's function has been developed for the infinit water depth case [10], and applied to the case of a VLFS. This formulation is appropriate for describing a wave farm, which would ideally be situated in large depth, to avoid energy losses in the incident waves due to bathymetry effects. In case of shallow water, it is still possible to consider a complex seabed, represented as an independent, nonmoving body.

This paper describes an implementation of the fast multipole algorithm in Aquaplus, in-house diffraction/radiation software developed over 30 years at LMF [11]. The firs step is the extension of Utsunomiya's formulations [10] to enable their use in a 3D fast multipole algorithm. Some convergence difficultie for the multipole expansion are then underlined. An open-source algorithm (DPMTA) using the previously developed formulations is then integrated into Aquaplus. Finally, a simplifie version of the fast multipole algorithm, more suited to the specifi problem of wave energy converter arrays, is presented. Performance in terms of computation time, accuracy, and memory needs is investigated.

2 Methods

2.1 The boundary element problem

The water is modeled as inviscid and incompressible. The flui velocity is the gradient of a potential ϕ. The motion amplitude of the bodies compared with the wavelength and the wave steepness are supposed to be small. As a consequence, the free-surface condition (Eq. 3) can be expressed linearly. The water depth is supposed to be infinite The corresponding boundary problem is the following (see [12]):

$$\Delta\phi = 0 \quad \text{in the entire flui domain,} \tag{1}$$

$$\frac{\partial\phi}{\partial n} = \vec{V}_i \cdot \vec{n} \quad \text{on the surface } S_i \text{ of body } i, \tag{2}$$

$$\frac{\partial^2\phi}{\partial t^2} + g\frac{\partial\phi}{\partial z} = 0 \quad \text{on the free surface,} \tag{3}$$

where g is the gravitational acceleration and $\vec{V}_i$ is the velocity on the surface of body i. In the small motions hypothesis, the wet surface is supposed to be constant. Definin

$$\Phi = \mathrm{Re}[\phi \mathrm{e}^{-\mathrm{i}\omega t}], \tag{4}$$

$$\phi(P) = -\frac{1}{4\pi} \iint\limits_{\sum_i S_i} \sigma(Q) G(Q, P) \mathrm{d}S(Q), \tag{5}$$

and applying Green's second formula leads to the following integral equation [11]:

$$\frac{\sigma(P)}{2} - \frac{1}{4\pi} \iint\limits_{\sum_i S_i} \sigma(Q) \frac{\partial G(Q, P)}{\partial n_\mathrm{P}} \mathrm{d}S(Q) = \vec{V}_\mathrm{P} \cdot \vec{n}_\mathrm{P}, \tag{6}$$

with ω the angular frequency and $\sigma(Q)$ the source density at a source point $Q(\zeta, \eta, \xi)$, assuming a source distribution. $P(x, y, z)$ is the fiel point, and $\vec{n}_\mathrm{P}$ and $\vec{V}_\mathrm{P}$ are the panel normal and the flui velocity at P. All coordinates are define in a system centered at $O(x_\mathrm{C}, y_\mathrm{C}, 0)$; the free surface is situated at $z = 0$. This method, which uses Eqs. 5 and 6, is preferred to a direct method (in which the unknown is the potential), as it gives better accuracy when computing velocities. Such a method is used in the hydrodynamic software WAMIT [13] as well.

In the CPM, the surfaces of the bodies are represented by a total of N_panels fla panels, on which the unknowns σ are constant. Equation 6 has to be solved at the center M_j of each panel, leading to a $N_\mathrm{panels} \times N_\mathrm{panels}$ linear system:

$$\mathbf{K}\sigma = \frac{\partial \varphi}{\partial n}, \tag{7}$$

$$\varphi = \mathbf{S}\sigma, \tag{8}$$

with ($i, j \leq N_\mathrm{panels}$, S_{M_j} the surface of panel j):

$$\varphi_j = \phi(M_j), \tag{9}$$

$$\mathbf{K}_{ij} = \delta_{ij} \frac{1}{2} + \iint\limits_{S_{M_j}} \frac{\partial G(M_i, M_j)}{\partial n_{M_j}} \mathrm{d}S(M_i), \tag{10}$$

$$\mathbf{S}_{ij} = \iint\limits_{S_{M_j}} G(M_i, M_j) \mathrm{d}S(M_i). \tag{11}$$

The solution of Eq. 7 by GMRes involves matrix–vector products (MVPs) $\mathbf{K}r^{(l)}$, with $r^{(l)}$ the residual at iteration l. The contribution of the FMA will be to speed up the evaluation of these products, each product then requiring $O(N_\mathrm{panels})$ complexity, instead of $O(N_\mathrm{panels}^2)$. Similarly, the FMA can be used to compute the MVP $\mathbf{S}\sigma$ in Eq. 8 to obtain the potential on each panel. For the infinit water depth case, the Green's function which satisfie the boundary element problem has the following expression [5] in the frequency domain:

$$G(Q, P) = \underbrace{\frac{1}{r} + \frac{1}{r_1}}_{G_1} + \underbrace{\int\limits_0^\infty \frac{2\nu}{k - \nu} \mathrm{e}^{k(z+\zeta)} J_0(kR) \mathrm{d}k}_{G_2} \tag{12}$$

with ν the wavenumber, J_n the nth-order Bessel function of the firs kind, $r = \sqrt{(x-\xi)^2 + (y-\eta)^2 + (z-\zeta)^2}$, $r_1 = \sqrt{(x-\xi)^2 + (y-\eta)^2 + (z+\zeta)^2}$, and $R = \sqrt{(x-\xi)^2 + (y-\eta)^2}$. The coordinates are $Q(r_\xi, \alpha_\xi, \theta_\xi)$ and $P(r_x, \alpha_x, \theta_x)$ in the spherical system, and $Q(R_\xi, \alpha_\xi, \zeta)$ and $P(R_x, \alpha_x, z)$ in the cylindrical system, both systems being centered at $\mathbf{C}(x_\mathrm{C}, y_\mathrm{C}, z_\mathrm{C})$ (Fig. 1). The integral in Eq. 12 is in the sense of Cauchy principal value.

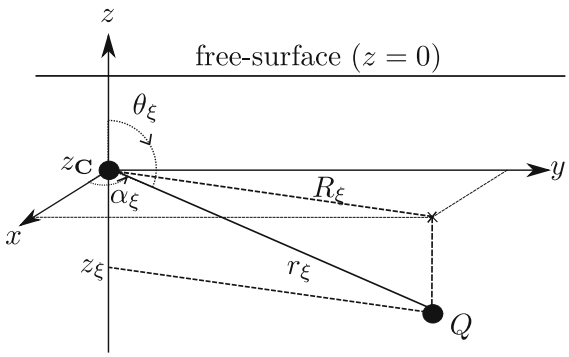


Fig. 1 Coordinates of the source point E in the spherical and cylindrical systems centered at **C**

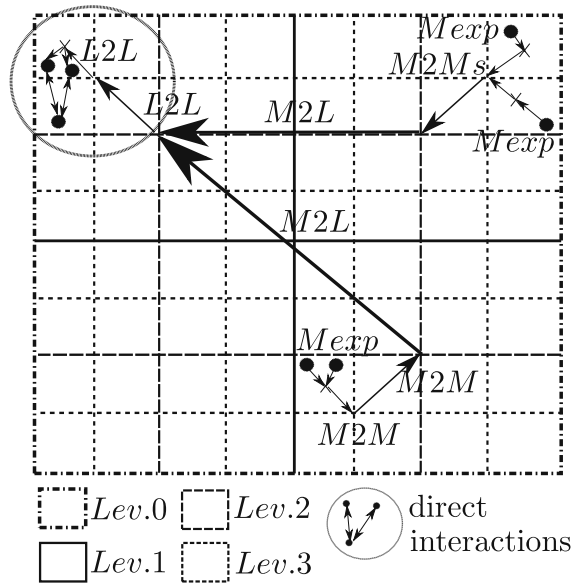


Fig. 2 Principle of the FMA, illustrated in 2D: the surface is meshed by a quad-tree. $N_{\mathrm{L}} = 4$

2.2 Role of the fast multipole algorithm

The solution of Eq. 6 by GMRes generates MVPs $\mathbf{K}r^{(l)}$. These operations are time consuming when N_{panels} is large. This is mainly due to the building of $\mathbf{K}$ for the firs MVP (at the firs iteration of the GMRes). $\mathbf{K}$ remains the same in the following MVPs, as it only depends on the wavelength and the positions of the panels. In the FMA, the contribution of neighboring panels and more distant panels are expressed separately:

$$\mathbf{K}r^{(l)} = (\mathbf{K}_{\mathrm{far}}r^{(l)})_{\mathrm{FMA}} + \mathbf{K}_{\mathrm{near}}r^{(l)}. \tag{13}$$

In Eq. 13, the MVP $(\mathbf{K}_{\mathrm{far}}r^{(l)})_{\mathrm{FMA}}$ is directly evaluated by FMA without having to build the matrix $\mathbf{K}_{\mathrm{far}}$. This MVP applies to interactions occurring between groups of panels that are well separated from each other (the vast majority of the interactions). The matrix $\mathbf{K}_{\mathrm{near}}$ has to be explicitly built, $\mathbf{K}_{\mathrm{near}}r^{(l)}$ being computed directly afterwards.

2.3 Principle of the fast multipole algorithm

Liu and Nishimura [14] provide a good introduction to the enhancement of BEM software using the FMA. The mathematical foundations of this algorithm as well as its principles are fully developed in [1,15]. Variations of this algorithm can be found in [16,17].

The FMA splits the physical simulation space into cells, building a hierarchical oct-tree: the elementary cell is split into height children cells, themselves split into eight cells, and so on (3D case). If the simulation space is only a surface, then a quad-tree is used, each cell having four children (2D case, Fig. 2). The total number of cells depends on the number of decomposition levels N_{L}. The FMA differentiation between close (or direct) cell interactions and far cell interactions is based on the distance between cells compared with the cell size. Figure 2 illustrates in 2D the steps the FMA goes through to express the influenc of a group of panels on another group. Level 1 cells are the children of the level 0 cell, and so on. Once the tree has been built, the evaluation of the far-fiel interactions is carried out using the following process:

– The influenc of the panels is expressed at the center of each cell and added together (*Mexp* in Fig. 2). This step involves computing the moments of the Green's function multipole expansion (see Eq. 19).

– The sum of these moments, corresponding to a "grouped influence of panels, is translated to the target cell. Specifi operators are used: M2M and L2L (moment to moment and local to local) for short-distance transfers, and moment to local (M2L) for long-distance transfers. The hierarchical relations between cells (parent–child–sibling) determine which translations have to be used.
– The influenc of the far group of panels is then split on the panels belonging to the target cell (see Eq. 35).

This method is suited to problems involving a large number of unknowns, because most of the computation effort is shared between the particles. Indeed, adding a particle only means computing an extra multipole expansion and an extra distribution [this explains the $O(N_{\text{panels}})$ tendency when N_{panels} is large enough]. In the direct method, interactions would have been computed between the extra particle and all the other particles.

3 Multipole expansion of the free-surface Green's function

In this part, formulations of the multipole expansion of the infinit depth Green's function are considered. Formulations from [10] are extended. The original formulations implied that the expansion center **C** had to be on the free surface. As a consequence, these formulations had to be used in a 2D algorithm: only the horizontal plane of the free surface was meshed by a hierarchical quad-tree. The translation operations mentioned in Sect. 2.3 were then purely horizontal, the expansion centers being situated on the free surface. The following formulations permit $z_{\text{C}} \neq 0$, enabling their use in a 3D algorithm. In such a case, the complete volume of the simulation domain can be meshed by a hierarchical oct-tree, allowing expansion centers at diverse depths. Some restrictions for the far-fiel formulations present in [10] are proven to be unnecessary.

3.1 Multipole expansion

3.1.1 Near-fiel G_1

The expression for the near fiel can be derived directly from [1]. Greengard gives the following expression for the multipole expansion of the potential in the case of Coulombic interactions:

$$\frac{1}{r} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{M_n^m}{r_x^{n+1}} Y_n^m(\theta_x, \alpha_x), \tag{14}$$

and the moments of the expansion at a source point Q are $(M_n^m \times r^{(l)}(Q))$, with

$$M_n^m = r_\xi^n Y_n^{-m}(\theta_\xi, \alpha_\xi), \tag{15}$$

where Y_n^m are the spherical harmonics of degree n. Theses harmonics can be expressed using associated Legendre functions $P_n^{|m|}$ as

$$Y_n^m(\theta, \alpha) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos\theta) \mathrm{e}^{\mathrm{i}m\alpha}. \tag{16}$$

In Eqs. 14 and 15, the condition $r_\xi < r_x$ is necessary to ensure that the errors due to the sum truncations remain bounded. As a consequence, **C** has to be close to Q, leading to $z_{\text{C}} \leq 0$. The expression for $1/r_1$ can be easily deduced from above. $Q'(\xi, \eta, -\zeta)$ is the symmetrical of Q with respect to the free surface. To satisfy the previous condition, a new expansion center $\mathbf{C}'(x_{\text{C}}, y_{\text{C}}, -z_{\text{C}})$ needs to be define above the free surface. Definin by $(r'_\xi, \theta'_\xi, \alpha'_\xi)$ and $(r'_x, \theta'_x, \alpha'_x)$ the coordinates of Q' and P in a system centered at $\mathbf{C}'$, we have: $\cos\theta'_\xi = -\cos\theta_\xi$, $r'_\xi = r_\xi$, $\alpha'_\xi = \alpha_\xi$, $\alpha'_x = \alpha_x$. Knowing that

$$P_n^m(-\cos\theta_\xi) = (-1)^{n+m} P_n^m(\cos\theta_\xi), \tag{17}$$

we can express the multipole expansion of the near fiel between Q and P as

$$G_1 = \sum_{n=0}^{\infty} \sum_{n=-m}^{m} M_n^m (n-|m|)! \left(\frac{P_n^{|m|}(\cos\theta_x)}{r_x^{n+1}} + (-1)^{n+m} \frac{P_n^{|m|}(\cos\theta'_x)}{(r'_x)^{n+1}} \right) \mathrm{e}^{\mathrm{i}m\alpha_x}, \tag{18}$$

with

$$M_n^m = r_\xi^n \frac{P_n^{|m|}(\cos\theta_\xi)}{(n+|m|)!} \mathrm{e}^{-\mathrm{i}m\alpha_\xi}. \tag{19}$$

The obtained formulation is very similar to Utsunomiya's. Some slight differences are pointed out below:

- In the case of bodies with deep drafts, the multipole expansion of G_1 would not converge if Q and Q' share a common expansion center, as in [10]. As a consequence, $\mathbf{C}'$ had to be introduced here.
- Using Greengard's theory shows that Eq. 18 is valid whatever the relative position of Q and P, since $r_\xi < r_x$. A condition $z - \zeta < 0$ is therefore not necessary to ensure convergence.

The convergence of this expansion will be faster the smaller $\mathbf{r}_{G_1} = r_\xi / r_x$ is.

3.1.2 Far-fiel G_2

This part follows Utsunomiya's steps. Applying Graf's addition theorem on Bessel functions in G_2 leads to

$$G_2 = \int_0^{\infty} \frac{2\nu}{k-\nu} \mathrm{e}^{k(\zeta - z_\mathrm{C})} \mathrm{e}^{k(z+z_\mathrm{C})} \times \sum_{m=-\infty}^{\infty} J_m(kR_\xi) J_m(kR_x) \mathrm{e}^{-\mathrm{i}m\alpha_\xi} \mathrm{e}^{\mathrm{i}m\alpha_x} \mathrm{d}k. \tag{20}$$

Extending a formulation from [18], Utsunomiya sets, $\forall m$,

$$\mathrm{e}^{k(\zeta - z_\mathrm{C})} J_m(kR_\xi) = \epsilon_m \sum_{n=|m|}^{\infty} r_\xi^n \frac{P_n^{|m|}(\cos\theta_\xi)}{(n+|m|)!} k^n, \tag{21}$$

with

$$\epsilon_m = \begin{cases} 1 & \text{if } m \geq 0, \\ (-1)^m & \text{if } m < 0. \end{cases} \tag{22}$$

Considering that $J_{-m}(kR_\xi) = (-1)^m J_m(kR_\xi)$, then, $\forall m$,

$$J_m(kR_x) = \epsilon_m J_{|m|}(kR_x). \tag{23}$$

Substituting Eqs. 21 and 23 into Eq. 20, and noting that

$$\sum_{m=-\infty}^{\infty} \sum_{n=|m|}^{\infty} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n}, \tag{24}$$

leads to

$$G_2 = 2\nu \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \mathrm{e}^{\mathrm{i}m\alpha_x} \int_0^{\infty} \frac{k^n}{k-\nu} \mathrm{e}^{kz'_x} J_{|m|}(kR_x) \mathrm{d}k, \tag{25}$$

with $z'_x = z - (-z_\mathrm{C}) < 0$. $(r'_x, \theta'_x, \alpha'_x)$ and (R'_x, α'_x, z'_x) are the spherical and cylindrical coordinates of F in systems centered at $\mathbf{C}'$, with $R'_x = R_x$, $\alpha'_x = \alpha_x$. We defin the last integral as G_3:

$$G_3 = \int_0^{\infty} \frac{k^n}{k-\nu} \mathrm{e}^{kz'_x} J_{|m|}(kR_x) \mathrm{d}k. \tag{26}$$

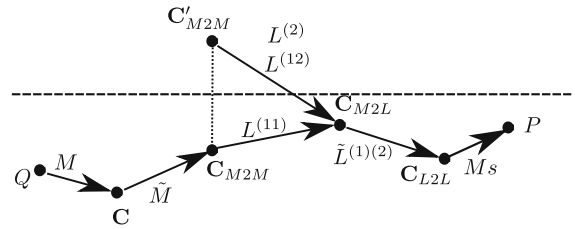


Fig. 3 Notations of the multipole and local expansion coefficient depending on the translation (operators identifie by their center $\mathbf{C}_{...}$)

From [10]:

$$G_3 = \begin{cases} \displaystyle\sum_{p=1}^{n-|m|} \nu^{p-1}(-1)^{m+n-p}(n-p-|m|)!\frac{P_{n-p}^{|m|}(\cos\theta_x)}{(r'_x)^{n-p+1}} + \nu^{n-|m|}\int_0^\infty \frac{1}{k-\nu}\mathrm{e}^{kz'_x}J_{|m|}(kR_x)k^{|m|}\mathrm{d}k \quad \text{if } n > |m|, \\ \displaystyle\int_0^\infty \frac{1}{k-\nu}\mathrm{e}^{kz'_x}J_{|m|}(kR_x)k^{|m|}\mathrm{d}k \quad \text{if } n = |m|, \end{cases} \tag{27}$$

in which the last integral define as G_4 can be evaluated by Eq. 28, for $m > 0$, as

$$G_4 = \frac{2}{\pi}\left(\frac{-\nu^m\pi^2\mathrm{e}^{\nu z'_x}}{4}[(-1)^m\mathbf{H}_{-\mathbf{m}}(\nu R_x) - Y_m(\nu R_x)] + \frac{\sqrt{\pi}(2R_x)^m\Gamma(m+\frac{1}{2})}{2}\int_0^{z'_x}\frac{\mathrm{e}^{\nu(z-s)}}{(s^2+R_x^2)^{m+\frac{1}{2}}}ds\right)$$
$$-\pi\nu^m\mathrm{i}\mathrm{e}^{\nu z'_x}H_m^{(2)}(\nu R_x), \tag{28}$$

with $\mathbf{H}$ the Struve function, $H^{(2)}$ the Hankel function of second kind, Y the Bessel function of second kind, and Γ the Gamma function (see also Appendix A). Note that

$$\mathrm{Im}(G_2) = -2\pi\nu\sum_{n=0}^{\infty}\sum_{m=-n}^{n}(\nu r_\xi)^n\frac{P_n^{|m|}(\cos\theta_\xi)}{(n+|m|)!}\mathrm{e}^{-\mathrm{i}m\alpha_\xi}\mathrm{e}^{\nu z'_x}J_m(\nu R_x)\mathrm{e}^{\mathrm{i}m\alpha_x}. \tag{29}$$

Equation 29 shows that the term $\mathbf{r}_{G_2} = \nu r_\xi$ has to be as small as possible to get good convergence of the expansion of G_2. The moment of the multipole expansion is the same for G_1 and G_2, see Eqs. 19 and 25.

3.2 Translation operators

The following formulas can be demonstrated using the addition theorem on Bessel functions on J_m terms in the expansions of G_1 and G_2 (refer to [10] for another demonstration of G_1 expansion, where J_m terms appear). The notations of the translation coefficient are detailed in Fig. 3. For simplicity's sake, only one M2M is represented, even if several M2M can follow each other when the hierarchical tree is climbed up. This warning applies to the L2L as well (the tree being climbed down).

3.2.1 Moment to moment

Let us have M the original multipole expansion coefficients and $\tilde{M}$ the translated ones. $(r_{x_M}, \alpha_{x_M}, \theta_{x_M})$ are the coordinates of $\mathbf{C}$ seen from $\mathbf{C}_{\mathrm{M2M}}$. Then,

$$\tilde{M}_n^m = \sum_{t=0}^{n}\sum_{s=-t}^{t} M_{n-t}^{m-s}\epsilon_{m-s}\epsilon_m \underbrace{\epsilon_s r_{x_M}^t \frac{P_t^{|s|}(\cos\theta_{x_M})}{(t+|s|)!}\mathrm{e}^{-\mathrm{i}s\alpha_{x_M}}}_{\mathrm{Coeff}_{\mathrm{M2M}_t^s}} . \tag{30}$$

3.2.2 Moment to local

The local expansion resulting from the M2L translation can be expressed as

$$L = L^{(1)} + L^{(2)}. \tag{31}$$

As Fig. 3 shows, the M2L for $L^{(1)}$ depends on the original center of expansion (here $\mathbf{C}_{\mathrm{M2M}}$) and on its symmetrical $\mathbf{C}'_{\mathrm{M2M}}$ with respect to the free surface. The coordinates of the center of the multipole expansion and its symmetrical seen from the local expansion point ($\mathbf{C}_{\mathrm{M2L}}$) are subscripted by x_{ML} and x'_{ML}. $z_{x'_{\mathrm{ML}}}$ is the difference of depth between $\mathbf{C}'_{\mathrm{M2M}}$ and $\mathbf{C}_{\mathrm{M2L}}$,

$$\begin{aligned} L_t^{s(1)} = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} \epsilon_s\epsilon_m \tilde{M}_n^m \Bigg(& \underbrace{\epsilon_{s-m}[(n+t)-|s-m|]!\frac{P_{n+t}^{|s-m|}(\cos\theta_{x_{\mathrm{ML}}})}{r_{x_{\mathrm{ML}}}{}^{n+t+1}}\mathrm{e}^{-\mathrm{i}(s-m)\alpha_{x_{\mathrm{ML}}}}}_{\mathrm{Coeff}_{\mathrm{M2L}(11)_{n+t}^{s-m}}} \\ & \underbrace{+\epsilon_{s-m}[(n+t)-|s-m|]!(-1)^{t+n+(s-m)}\frac{P_{n+t}^{|s-m|}(\cos\theta_{x'_{\mathrm{ML}}})}{r_{x'_{\mathrm{ML}}}^{n+t+1}}\mathrm{e}^{-\mathrm{i}(s-m)\alpha_{x_{\mathrm{ML}}}}}_{\mathrm{Coeff}_{\mathrm{M2L}(12)_{n+t}^{s-m}}} \Bigg). \end{aligned} \tag{32}$$

The calculation of $L^{(2)}$ remains as in [10]:

$$L_t^{s(2)} = 2\nu\sum_{n=0}^{\infty}\sum_{m=-n}^{n} \epsilon_s\epsilon_m\tilde{M}_n^m \underbrace{\epsilon_{s-m}\mathrm{e}^{-\mathrm{i}(s-m)\alpha_{x_{\mathrm{ML}}}}\int_0^{\infty}\frac{k^{n+t}}{k-\nu}\exp\left(-k\left|z_{x'_{\mathrm{ML}}}\right|\right)J_{|s-m|}(kR_{x_{\mathrm{ML}}})\mathrm{d}k}_{\mathrm{Coeff}_{\mathrm{M2L}(2)_{n+t}^{s-m}}} . \tag{33}$$

Note that a M2L translation cannot be computed between two vertically aligned points. This would lead to undefine quantities $Y_m(0)$ when $m > 0$ in Eq. 33. The relative position of multipole and local expansion centers has to be carefully chosen.

3.2.3 Local to local (L2L)

Let us have L the original local expansion coefficients and $\tilde{L}$ the translated ones. $(r_{x_{\mathrm{L}}}, \alpha_{x_{\mathrm{L}}}, \theta_{x_{\mathrm{L}}})$ are the coordinates of $\mathbf{C}_{\mathrm{M2L}}$ seen from $\mathbf{C}_{L2L}$,

$$\tilde{L}_u^v = \sum_{t=v}^{\infty}\sum_{s=-t}^{t}\frac{\epsilon_u}{\epsilon_s}L_t^s \underbrace{\epsilon_{u-s}(-1)^{(t-v)+(u-s)}r_{x_{\mathrm{L}}}^{t-v}\frac{P_{t-v}^{|u-s|}(\cos\theta_{x_{\mathrm{L}}})}{[(t-v)-|u-s|]!}\mathrm{e}^{-\mathrm{i}(u-s)\alpha_{x_{\mathrm{L}}}}}_{\mathrm{Coeff}_{\mathrm{L2L}_{t-v}^{u-s}}} . \tag{34}$$

3.2.4 Computing the Green's function from local expansion coefficient

Finally, the evaluation of G is given by

$$G(Q,P) = \sum_{v=0}^{\infty}\sum_{u=-v}^{v}\tilde{L}_v^u \underbrace{(r_x)^v\frac{P_v^{|u|}(\cos\theta_x)}{(v+|u|)!}\mathrm{e}^{\mathrm{i}u\alpha_x}}_{Ms_v^u}, \tag{35}$$

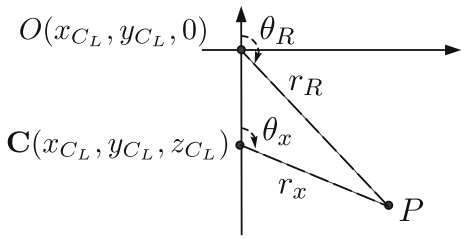


Fig. 4 Coordinate system when $z_C \neq 0$

with $(r_x, \alpha_x, \theta_x)$ the coordinate of the fiel point P seen from the local expansion center and $\tilde{L}_v^{u(1)(2)}$ the local moments define in Eq. 34. Ms_v^u has a similar form to the moment of the expansion (Eq. 19) except that it refers to the fiel point instead of referring to the source point. All these operators can be applied to the Green's function and to its normal derivatives.

3.3 Normal derivatives

According to [7], the normal derivatives at the source point can be obtained from the gradient of the moments. In Aquaplus the derivation occurs at the fiel point P. As a consequence, the Ms_n^m coefficient are differentiated instead of the M_n^m coefficient as in [7]:

$$\frac{\partial Ms_n^m}{\partial n_{\mathrm{P}}} = \left(\frac{\partial Ms_n^m}{\partial r_{\mathrm{R}}} \vec{e}_{r_{\mathrm{R}}} + \frac{1}{r_{\mathrm{R}}} \frac{\partial Ms_n^m}{\partial \theta_{\mathrm{R}}} \vec{e}_{\theta_{\mathrm{R}}} + \frac{1}{r_{\mathrm{R}} \sin\theta_{\mathrm{R}}} \frac{\partial Ms_n^m}{\partial \alpha_x} \vec{e}_{\alpha_{\mathrm{R}}} \right) \cdot \vec{n}_{\mathrm{P}}, \tag{36}$$

with $(\vec{e}_{r_{\mathrm{R}}}, \vec{e}_{\theta_{\mathrm{R}}}, \vec{e}_{\alpha_{\mathrm{R}}})$ the orthonormal base of the spherical system centered at $O(x_{C_{\mathrm{L}}}, y_{C_{\mathrm{L}}}, 0)$ (in this section, C_{L} refers to the last local expansion center). The Green's function normal derivative is then simply given by

$$\frac{\partial G(Q, P)}{\partial n_{\mathrm{P}}} = \sum_{v=0}^{\infty} \sum_{u=-v}^{v} \tilde{L}_v^u \frac{\partial Ms_u^v}{\partial n_{\mathrm{P}}}. \tag{37}$$

Referring to Eq. 35, the partial derivatives with respect to α and r are trivial when $z_{C_{\mathrm{L}}} = 0$. Derivatives with respect to θ call for the derivatives of P_n^m (obtained by recurrence relationships). For $z_{C_{\mathrm{L}}} \neq 0$, one needs to compose the derivatives in order to calculate the gradient at the fiel point F (see notation in Fig. 4). This leads to

$$\frac{\partial}{\partial \alpha_{\mathrm{R}}} Ms_n^m = \frac{\partial}{\partial \alpha_x} Ms_n^m = \mathrm{i}m(r_x^n) \frac{P_n^{|m|}(\cos\theta_x)}{(n+|m|)!} \mathrm{e}^{\mathrm{i}m\alpha_x}, \tag{38}$$

$$\frac{\partial}{\partial r_{\mathrm{R}}} Ms_n^m = \left[r_x^n \frac{\partial \cos\theta_x}{\partial r_{\mathrm{R}}} \frac{\partial P_n^{|m|}(\cos\theta_x)}{\partial \cos\theta_x} + P_n^{|m|}(\cos\theta_x) \frac{\partial r_x}{\partial r_{\mathrm{R}}} n\, r_x^{n-1} \right] \frac{\mathrm{e}^{\mathrm{i}m\alpha_x}}{(n+|m|)!}, \tag{39}$$

$$\frac{\partial}{\partial \theta_{\mathrm{R}}} Ms_n^m = \left[r_x^n \frac{\partial \cos\theta_x}{\partial \theta_{\mathrm{R}}} \frac{\partial P_n^{|m|}(\cos\theta_x)}{\partial \cos\theta_x} + P_n^{|m|}(\cos\theta_x) \frac{\partial r_x^n}{\partial \theta_{\mathrm{R}}} \right] \frac{\mathrm{e}^{\mathrm{i}m\alpha_x}}{(n+|m|)!}, \tag{40}$$

(the missing coefficient are given in Appendix B).

3.4 Validation

These formulations have been extensively tested and compared with Aquaplus original formulations, as well as analytical results. Details can be found in [19]. As explained in Sects. 3.1.1 and 3.1.2, $\mathbf{r}_{G_1} = r_\xi / r_x$ and $\mathbf{r}_{G_2} = \nu r_\xi$ have to be small enough to ensure good convergence of the expansion with a limited number of terms (truncation order N_{p}). Table 1 shows that, the higher N_{p}, the greater the fl xibility on $\mathbf{r}_{G_1}$ and $\mathbf{r}_{G_2}$. When these limit values are passed, the error increases brutally on G_2, and more smoothly on G_1.

Table 1 Maximum allowed values of $\mathbf{r}_{G_1}$ and $\mathbf{r}_{G_2}$ to ensure an error lower than 1 % on G_1 and $Im(G_2)$ (comparison between analytical solutions and multipole expansions)

	$N_\mathrm{p} = 4$	$N_\mathrm{p} = 8$	$N_\mathrm{p} = 12$
$\mathbf{r}_{G_1}$	0.2	0.4	0.5
$\mathbf{r}_{G_2}$	0.2	1.0	2.1

4 Implementation of the fast multipole algorithm

4.1 Implementation details

The open-source tool DPMTA (distributed parallel multipole tree algorithm [17]) has been implemented into Aquaplus. This tool offers a version of the FMA as described by [1], and a slightly modifie one permitting manual tuning of the proportions of far and near interactions by the use of a "multipole acceptance criterion" (mac). The DPMTA features several acceleration methods [fast Fourier transform (FFT), parallel implementation] which could be adapted to the hydrodynamics formulations in the future.

The DPMTA is dedicated to molecular dynamics and works with the following functions: $G = 1/r$ (electrostatic interactions) and $G = 1/r^6$ (Lennard–Jones interactions). The hydrodynamic formulations have been added. The coordinate system and the normalization routines have been modifie to take the free surface into account. The center of expansion within each cell has been moved away from the cell geometric center, to avoid purely vertical M2Ls between cells (see Eq. 33). The DPMTA and Aquaplus are compiled in the same executable, to avoid time-consuming data exchanges through files

This tool is called at each MVP. Considering Eqs. 19, 30, and 32–35, it is clear that many coefficient of the multipole formulations only need to be computed once. Indeed:

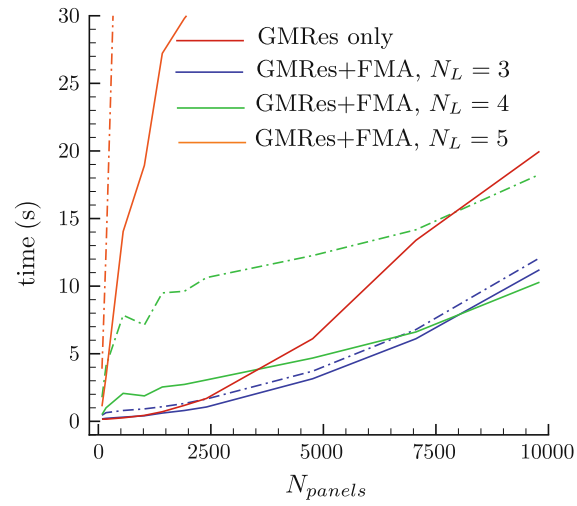
- M_n^m, $\mathrm{Coeff}_{\mathrm{M2M}}{}_n^m$, $\mathrm{Coeff}_{\mathrm{M2L(1)}}{}_n^m$, $\mathrm{Coeff}_{\mathrm{L2L}}{}_n^m$, and Ms_n^m depend only on the geometry of the problem. They can be shared between all iterations of radiation/diffraction problems for all wave periods.
- $\mathrm{Coeff}_{\mathrm{M2L(2)}}{}_n^m$ depend on the geometry and the wave period, and are shared from one iteration to another and one problem to another.

These coefficient are computed only the firs time they are needed, then stored in random access memory (RAM, using a virtual memory disk) and reused. This does not apply to the moments of the multipole and local expansions, as they are "weighted" by $r^{(l)}$ and change at each iteration (l).

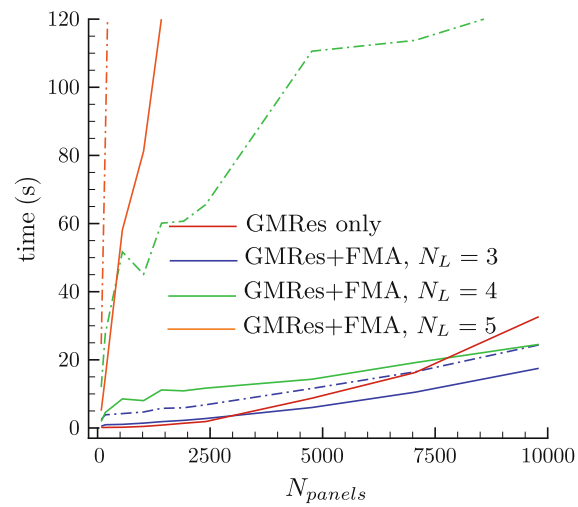
4.2 Performance

This section focuses on the computation time required for solving the heaving problem on a hemisphere modeled by 10^2 to 10^4 fla panels. Two parameters of the FMA are investigated: the truncation order ($N_\mathrm{p} = 5, 10$) and the number of levels in the hierarchical tree ($N_\mathrm{L} = 3, 4, 5$). The central processing unit (CPU) time is plotted depending on which solver is used. Figure 5a illustrates the time needed to perform the firs MVP in the GMRes, using either the GMRes alone or GMRes combined with the FMA. When the FMA is used, the shape of the time curve has two different slopes. Indeed:

- For a low number of panels, refinin the mesh means populating new cells of the hierarchical tree with new panels. As these cells interact together through translation operators, more operations are needed, and the CPU time rises rapidly.
- For a higher number of panels, most of the cells are already active. Adding a panel only means computing one extra moment (Eq. 19) and distribution (Eq. 35). The CPU time is then much less affected by the number of panels, having an $O(N)$ dependence.



(a) First MVP



(b) Full resolution

Fig. 5 CPU time with and without the FMA, with varying number of levels. *Solid line* $N_\mathrm{p} = 5$, *dashed line* $N_\mathrm{p} = 10$

The number of cells being directly linked to N_L ($N_\mathrm{cells} = 2^{N_\mathrm{L}-1}$), a higher N_panels will be necessary to reach the break point when N_L is high. As the translation operators between cells are sums truncated at truncation order N_p, the CPU time will be more dependent on N_p when N_L is high; see the differences between dashed lines ($N_\mathrm{p} = 10$) and solid lines ($N_\mathrm{p} = 5$) at the same number of levels in Fig. 5a. However, having a high N_L is a sensible choice when the number of panels is high, as it reduces the number of interactions computed by the direct method. This is the reason why:

– The curves $N_\mathrm{L} = 3$ and $N_\mathrm{L} = 4$ cross around $N_\mathrm{panels} = 7,500$;
– The shape of the curve $N_\mathrm{L} = 3$ is parabolic, similarly to the GMRes curve.

Figure 5a seems to show that the FMA provides a fair acceleration to the GMRes (the curves $N_\mathrm{L} = 3$ and $N_\mathrm{L} = 4$ cross the GMRes curve at, respectively, 1,000 and 3,500 panels). However, in Fig. 5b, considering the complete solution of the heaving problem, these "crossing points" occur at a higher number of panels. This is due to two opposite tendencies:

– The FMA strongly speeds up the GMRes at the firs MVP, by avoiding explicitly building **K** at the firs iteration.

– At the following MVPs, the GMRes is faster alone than combined with the FMA. Indeed, computing by the direct method, $\mathbf{K}r^{(l)}$ is very fast once $\mathbf{K}$ has been built, considering the relatively small number of particles (10^4). The same operation using the FMA requires more intermediate computations, and is slightly slower.

As a consequence, the FMA is an appropriate choice for a low number of MVPs, so that the time benefi of the firs iteration is not completely lost during the next iterations. This number is low when the convergence of the GMRes is fast, and when the number of problems to solve for each linear system (in our application, for each wave period) is low enough. This tool may be inappropriate for the case of point absorber arrays, considering 10–50 bodies and six degrees of freedom per body.

It is worth mentioning that Aquaplus is in itself a fast solution method, as most of the coefficient of $\mathbf{K}$ and $\mathbf{S}$ are interpolated into a table of precomputed values. Thus, the time benefi added by the FMA appears smaller here than in other results from the literature, in which the direct method fully computes G and $\partial G/\partial n$.

5 A simplifie FMA

5.1 Limits of the FMA for very sparse problems

In the case of a very sparse set of particles, the nonadaptive version of the FMA shows some limits, related to the hierarchical tree and the convergence of G_2. Let us consider the case of floatin bodies, having a typical diameter of 10 m and separated from each other by a few hundred meters, as would occur in an array of wave energy converters (WECs). To reduce the amount of direct interactions between panels, a very large number of cells is needed. This permits to have no more that one body per cell (the bodies then interacting through the FMA operators). As a consequence, a large number of decomposition levels is needed. A large N_L has two consequences which reduce the computational performance:

– Numerous intermediary calculations are needed, for climbing the tree up (M2M) and down (L2L), and between sibling cells (M2L), as the number of cells increases.
– Many empty cells are created. Though this does not influenc the output of the computation, it has an impact on its performance, depending on the software implementation. In FMA software designed for a homogeneous set of particles, memory is allocated to the cells, and interactions are computed between the cells, regardless of whether they contain particles or not.

The tree structure also impacts the convergence of G_2. For the lower bounds of the wave periods range, one must reduce the distance between a panel and the "last" multipole expansion center, before the M2L occurs (this would be the distance $r_\xi = (E\mathbf{C}_{\mathrm{M2M}})$ in Fig. 3). In this way, the term νr_ξ is low enough to ensure fast convergence of G_2. This leads to two requirements:

– Cells of the highest level (leaves) can contain only one body (or part of body). The center of expansion is the barycenter of the panels belonging to the cell, in order to reduce r_ξ.
– Only leaf-to-leaf interactions can be computed. Indeed, grouping the influenc of two bodies (by M2M) means computing the moments of multipole expansions at a point between these bodies. In such a situation the condition $\nu r_\xi \ll 1$ is no longer respected.

This last requirement goes against the main strength of the FMA, which is to maximize the grouping of particles by climbing the hierarchical tree as much as possible before computing long-distance interactions (M2L). Plus, these requirements cannot be satisfie efficiently

– The number of cells have to be extremely high, to ensure that a cell containing panels will only be grouped with an empty cell (to keep the local sets of panels dense)
– Or a very restrictive condition on the interactions has to be set up to permit only "very far" interactions. Doing so, the amount of interactions computed by FMA drops dramatically.

Such difficulties result from the origin of the FMA as a tool for gravitational or Coulombic interactions. The mechanisms differentiating far and near interactions only ensure that $\mathbf{r}_{\mathbf{G_1}}$ is small enough (leading to easy convergence of G_1, which is similar to $G = 1/r$).

5.2 Simplified algorithm

To overcome these problems without looking for an adaptive implementation of the FMA, a simplified algorithm is proposed (called FMAS in what follows), which does not use a tree (the DPMTA is no longer used). This algorithm is directly based on the mathematical principles from [1]. The interactions between panels are managed as follows: the near interactions occur between panels belonging to the same body, and the far interactions occur between panels belonging to different bodies. The far interactions involve only some of the operations previously described:

- For each panel, the moments of the multipole expansion are computed at the body barycenter (see Eq. 19).
- The body centers interact through the M2L transformation (Eqs. 32–33). There are $N_{\text{bodies}} \times (N_{\text{bodies}} - 1)$ M2L to compute.
- Then, for each panel of the target body, Eq. 35 is used.

As in the DPMTA, the simulation space is scaled to reduce cut-off errors before using the multipole formulations. Such an algorithm reduces the number of intermediary calculations (no M2M and L2L) and satisfies the requirements detailed in Sect. 5.1. However, the user has to make sure that the bodies are:

- Far enough from each other for fast convergence of G_1.
- Small enough compared with the wavelength for fast convergence of G_2.

For arrays of wave energy point absorbers, these conditions apply.

5.3 Results

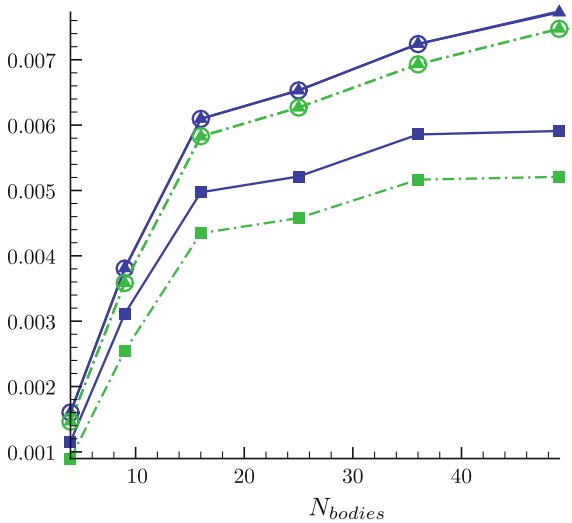
In this section we present the performance characteristics of the simplified algorithm. Arrays of square numbers (4–49) of floating cylinders are considered, with wave period equal to 7 s. Only the heaving problem is considered. Each cylinder is modeled by 130 or 260 panels. The cylinders are positioned along a regular square pattern of grid size 100 m. These computations are run on a 3.16 GHz core with 7.8 GB of RAM.

5.3.1 Accuracy

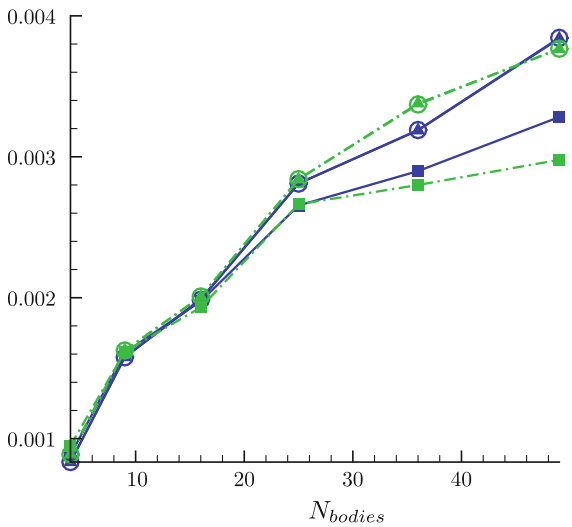
The relative difference between final results from GMRes alone and GMRes combined with FMAS were investigated, considering the radiation parameters (added mass and hydrodynamic damping) on the first two bodies of each array. They include the parameters of the body influencing itself by its motions and the body being influenced by the motions of the other bodies. Figure 6a, b shows good agreement between the two methods, the difference being always lower than 1 %. The $N_{\text{p}} = 10$ and $N_{\text{p}} = 15$ curves merge, proving the good convergence of the FMAS. The differences between methods increase when the number of bodies grows, as a greater proportion of interactions are computed by the FMAS (through M2Ls between bodies). This is probably due to the fact that the two methods use different mathematical operators (Bessel function, integral exponents), which are computed using numerical approximations.

5.3.2 Memory requirements

As seen in Sect. 4.1, various coefficients of the FMAS can be stored to save computational time. The memory requirements for the simplified algorithm follow these rules:

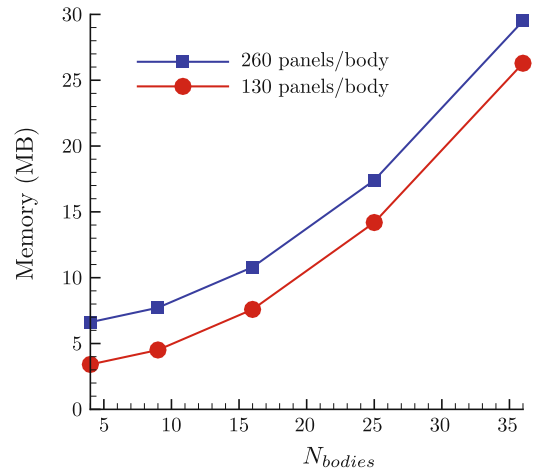


(a) Damping

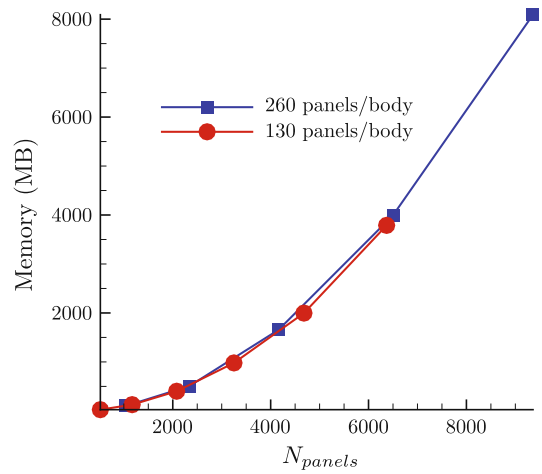


(b) Added mass

Fig. 6 Relative difference between results from GMRes+ FMAS and GMRes. *Squares* $N_{\mathrm{p}} = 5$, *triangles* $N_{\mathrm{p}} = 10$, *circles* $N_{\mathrm{p}} = 15$. *Solid line* 130 panels/body; *dashed line* 260 panels/bodies



(a) FMAS (direct interaction not considered)



(b) GMRes

Fig. 7 Memory required for storing the coefficient of the chosen solution method

– The M_n^m and Ms_n^m coefficient (plus Ms_n^m derivatives) are computed and stored only for the firs body. They are shared between bodies, as all the cylinders have the same shape.
– The $\mathrm{Coeff}_{\mathrm{M2L}(1)(2)}{}_n^m$ coefficient are stored for each M2L, leading to $N_{\mathrm{bodies}}(N_{\mathrm{bodies}} - 1)$ sets of coefficients
– The coefficient related to direct interactions (matrixes $\mathbf{K}_{\mathrm{near}}$ and $\mathbf{S}_{\mathrm{near}}$) are stored in RAM.

The storage of FMAS coefficient is statically implemented for $N_{p\,\max} = 20$. Figure 7a shows that the memory requirements have weak dependency on the number of panels per body (characterized by the "offset" of the curve), and a parabolic dependency on N_{bodies} as expected. The overall memory needs are low enough to store all the coefficient in RAM. For comparison, Fig. 7b shows the memory needs for the GMRes algorithm, in which the full matrixes **K** and **S** are stored. These need depend only on N_{panels}. The range of magnitude of memory needs is

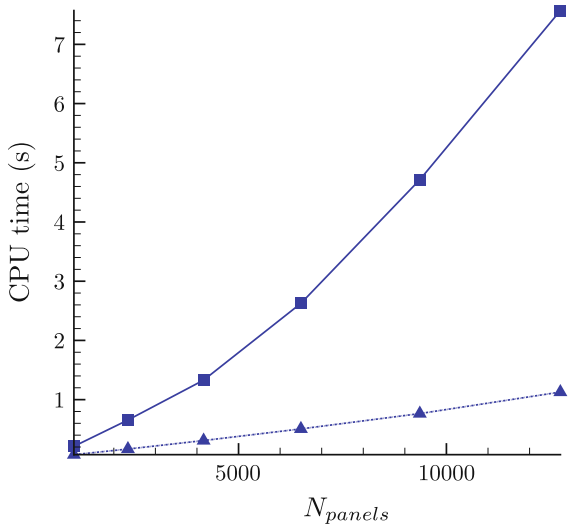


Fig. 8 CPU time for MVPs depending on the number of panels. *Squares* firs MVP, *triangles* average of all other MVPs

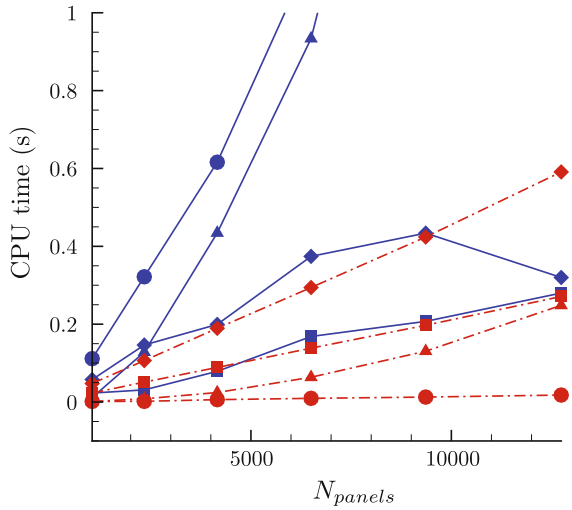


Fig. 9 CPU time for MVPs. *Solid line* firs MVP; *dashed line* average of all other MVPs. *Squares* multipole expansion; *triangles* M2L; *diamonds* distribution; *circles* direct

significantly smaller with the FMAS, making the software easier to use on a normal workstation. It is no longer necessary to write data on the hard drive, resulting in time savings in "wall clock" time (see Sect. 5.3.4).

5.3.3 Computational time characteristics of the FMAS

The CPU time can be expressed as follows, depending on the number of MVPs:

$$T_{\text{FMA}} = T_{\text{FMA}}^{(1)} + (n_{\text{MVP}} - 1)T_{\text{FMA}}^{(\text{MVP})}. \tag{41}$$

The time for the firs MVP is different from for the other ones because the FMAS coefficient are computed at this step. For $N_{\text{p}} = 10$, Fig. 8 presents $T_{\text{FMA}}^{(1)}$ and $T_{\text{FMA}}^{(\text{MVP})}$ depending on the number of panels ($N_{\text{panels}} = 260 \times N_{\text{bodies}}$). The CPU time grows almost linearly with N_{panels}; this tendency is more pronounced for $T_{\text{FMA}}^{(\text{MVP})}$, when $\mathbf{K}_{\text{near}}$ has been built. At each MVP $i \geq 1$,

$$\begin{aligned} T_{\text{FMA}}^{(i)} = {} & N_{\text{panels/body}} \times N_{\text{bodies}} \times (T_{\text{Mexp}} + T_{\text{Distribution}}) \\ & + N_{\text{bodies}} \times (N_{\text{bodies}} - 1) \times T_{\text{M2L}} \\ & + N_{\text{panels/body}}^2 \times T_{\text{direct}} \times N_{\text{bodies}}. \end{aligned} \tag{42}$$

Figure 9 illustrates Eq. 42. The $O(N_{\text{panels}})$ tendency is more obvious after the firs MVP.

5.3.4 Acceleration of GMRes by FMAS

The CPU time required by the GMRes alone is expressed as follows:

$$T_{\text{GMRes}} = T_{\text{GMRes}}^{(1)} + (n_{\text{MVP}} - 1)T_{\text{GMRes}}^{\text{MVP}}, \tag{43}$$

with $T_{\text{GMRes}}^{(1)} \gg T_{\text{FMA}}^{(1)}$ (the complete linear system has to be built) and $T_{\text{GMRes}}^{(\text{MVP})} < T_{\text{FMA}}^{(\text{MVP})}$. As a consequence there will be a critical number of MVP at which the FMA will no longer accelerate the solution. Table 2 indicates the acceleration provided by FMAS for the firs MVP. Figure 10 gives the number of MVPs n_{MVPc} under which $T_{\text{FMA}} < T_{\text{GMRes}}$. Under the plotted curves, the FMAS provides an actual acceleration to the GMRes. For a constant

Table 2 Speed-up provided by the FMAS to the firs MVP depending on the expansion order

N_{bodies}	N_{panels}	$N_{\text{p}} = 5$	$N_{\text{p}} = 10$
4	1,040	11	12
9	2,340	19	19
16	4,160	48	42
25	6,500	116	92
36	9,360	142	110
49	12,740	150	128

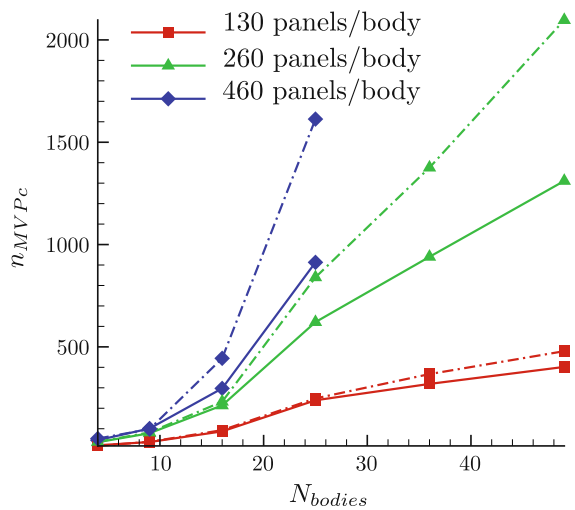


Fig. 10 Critical number of MVPs under which the FMAS speeds up the GMRes. *Solid line* considering CPU time; *dashed line* considering "wall clock" time

number of bodies, the acceleration is higher the more panels are used to represent each body. However, CPU-based results do not take into account access time to data in case GMRes is used alone. Figure 10 shows that considering wall clock time (in dashed lines) extends the zone where the FMAS is an improvement, by avoiding time spent reading or writing data. These results permit several configuration for which the FMAS speed-up will be significant over the whole computation to be underlined:

- If each body has a complex shape and needs to be modeled by a large number of panels. The FMAS thus permits computations on "realistic" point absorbers instead of simplifie shapes.
- If a limited number of radiation/diffraction problems are considered (reducing the overall number of MVPs). In cases where only a limited number of degrees of freedom are considered (for example, arrays of buoys with all motions restricted except heaving), the FMAS is perfectly suited. For more complex situations with six degrees of freedom per body, the FMAS will not accelerate the solution.
- If the convergence of the iterative solver is fast (reducing the overall number of MVPs).

6 Conclusions

This paper presents the implementation of a fast multipole algorithm into a BEM software package. The objective is to accelerate the solution of radiation/diffraction problems for floatin bodies in ocean waves. These problems are large and are characterized by sets of panels that are very irregularly distributed in space.

The firs step is to obtain a versatile expression for the multipole expansion of the free-surface Green's function. Results from [10] are extended, permitting the expansion center to be any depth. As a consequence, these formulations can be used in a FMA in three dimensions. They are then implemented into a distribution of the FMA

(DPMTA), in order to rapidly compute the matrix–vector products in the GMRes solution of diffraction/radiation problems. This implementation proved to be efficien when working on a single surface modeled by a large number of panels, assuming the FMA parameters are correctly chosen.

However, this tool is not suited for very sparse sets of particles (floatin bodies separated by more than fi e times their diameters). Its main limitation is that the oscillating kernel of the Green's function does not converge when the multipole expansion is translated along the hierarchical tree. A "simplified algorithm is proposed, based on the same formulations but not using a tree, which showed good performance in term of accuracy of the fina results (hydrodynamic coefficient of the floatin bodies). A significant speed-up can be achieved, provided that the number of matrix–vector products needed to solve the problems at each wave period is lower than a critical value. In situations where the acceleration is not significant, it is still of practical interest to use this implementation, which drastically reduces memory needs. The simplifie tool is tuned by only one parameter (expansion order) instead of three for the original FMA (expansion order, number of levels, multipole acceptance criterion), which makes it easier to use. The expansion order can be adapted to the wave period, the convergence for lower wave periods requiring more terms in the expansion.

Accelerating the convergence of the GMRes is critical to make the most of the fast multipole algorithm. A further perspective of this work is the implementation of a preconditioner dedicated to the FMA. Among other perspectives are the integration of finit depth formulations and the use of symmetries.

Appendix A: Calculation of G_4

The difference $(-1)^m \mathbf{H}_{-m}(\nu R_x) - Y_m(\nu R_x)$ contributes through G_4 to the oscillating part of the Green's function (Eq. 28). For better precision of the far field it has been found satisfactory to use an asymptotic expression. According to [20], for large arguments νR_x,

$$\mathbf{H}_m(\nu R_x) - Y_m(\nu R_x) = \frac{1}{\pi} \sum_{k=0}^{k_{\max}-1} \frac{\Gamma(k+\frac{1}{2})}{\Gamma(m+\frac{1}{2}-k)(\frac{\nu R_x}{2})^{2k-\nu-1}} \tag{44}$$

and

$$Y_{-m}(\nu R_x) = (-1)^m Y_m(\nu R_x). \tag{45}$$

Using

$$[(-1)^m \mathbf{H}_{-m}(\nu R_x) - Y_m(\nu R_x)] = (-1)^m [\mathbf{H}_{-m}(\nu R_x) - Y_{-m}(\nu R_x)] \tag{46}$$

we can use the asymptotic expansion Eq. 44 when $\nu R_x > 20$ with $k_{\max} = 20$.

The integral in G_4 is computed by the Simpson method. About 1,000 points are needed.

Appendix B: Coefficient for the normal derivatives

$$\frac{\partial r_x}{\partial r_{\mathrm{R}}} = \frac{r_{\mathrm{R}} - z_{\mathrm{C}} \cos\theta_{\mathrm{R}}}{\sqrt{r_{\mathrm{R}}^2 + z_{\mathrm{C}}^2 - 2\, z_{\mathrm{C}}\, r_{\mathrm{R}} \cos\theta_{\mathrm{R}}}}, \tag{47}$$

$$\frac{\partial \cos\theta_x}{\partial r_{\mathrm{R}}} = \frac{z_{\mathrm{C}}}{r_{\mathrm{R}}^2} \frac{\sin\theta_x{}^3}{\sin\theta_{\mathrm{R}}}, \tag{48}$$

$$\frac{\partial \cos\theta_x}{\partial \theta_{\mathrm{R}}} = -\sin\theta_{\mathrm{R}} \frac{\sin\theta_x + \cos\theta_x \frac{\cos\theta_{\mathrm{R}}}{\sin\theta_{\mathrm{R}}}}{\sin\theta_{\mathrm{R}} + \frac{\cos\theta_x}{\sin\theta_x}(\cos\theta_{\mathrm{R}} - \frac{z_{\mathrm{C}}}{r_{\mathrm{R}}})}, \tag{49}$$

$$\frac{\partial r_x^n}{\partial \theta_{\mathrm{R}}} = n\, z_{\mathrm{C}}\, r_{\mathrm{R}} \sin\theta_{\mathrm{R}} (r_{\mathrm{R}}^2 + z_{\mathrm{C}}^2 - 2\, z_{\mathrm{C}}\, r_{\mathrm{R}} \cos\theta_R)^{\frac{n}{2}-1}. \tag{50}$$

References

1. Greengard L (1988) The rapid evaluation of potential field in particle systems. MIT Press, Cambridge
2. Kashiwagi M (2000) Hydrodynamic interactions among a great number of columns supporting a very large fl xible structure. J Fluids Struct 14:1013–1034
3. Kagemoto H, Yue D (1986) Interactions among multiple three-dimensional bodies in water waves: an exact algebraic method. Journal of Fluid Mechanics 166:189–209
4. Fochesato C, Grilli S, Dias F (2007) Numerical modeling of extreme rogue waves generated by directional energy focusing. Wave Motion 44:395–416
5. Newman JN (1985) Algorithms for free-surface Green's function. J Eng Math 19:57–67
6. Wehausen J, Laitone E (1960) Surface waves, chap 13. Springer, Berlin
7. Utsunomiya T, Watanabe E (2002) Accelerated higher order boundary element method for wave diffraction/radiation problems and its applications. In: Proceedings of the 12th ISOPE conference, Kitakyushu, Japan
8. Teng B, Gou Y (2006) Fast multipole expansion method and its application in BEM for wave diffraction and radiation. In: Proceedings of the 10th ISOPE conference, San Francisco, CA
9. Gou Y, Teng B (2008) Research on hydrodynamic interaction between multiple floatin bodies. Technical report, Proceedings of the 8th conference on hydrodynamics, Nantes, France
10. Utsunomiya T, Okafuji T (2007) Wave response of a VLFS by accelerated Green's function method in infinit water depth. Int J Offshore Polar Eng 17:30–38
11. Delhommeau G (1993) Seakeeping codes Aquadyn and Aquaplus. In: 19th WEGMENT School, numerical simulation of hydrodynamics: ship and offshore structures
12. Newman JN (1977) Marine hydrodynamics. MIT Press, Cambridge
13. WAMIT software website, http://wamit.com/. Accessed 9 May 2012
14. Liu YJ, Nishimura N (2006) The fast multipole boundary element method for potential problems: a tutorial. Eng Anal Bound Elem 30:371–381
15. Greengard L, Rokhlin V (1997) A fast algorithm for particle simulations. J Comput Phys 135:280–292
16. Carrier J, Greengard L, Rokhlin V (1988) A fast adaptative multipole algorithm for particles simulation. J Sci Comput 9:669–686
17. Board J, Hakura Z, Elliot W, Rankin W (1995) Scalable variants of multipole-accelerated algorithms for molecular dynamics applications. Technical report, Duke University, Department of Electrical Engineering
18. Thorne R (1953) Multipole expansions in the theory of surface waves. Math Proc Camb Philos Soc 49:707–716
19. Borgarino B, Babarit A, Ferrant P (2011) Extension of free-surface Green's function multipole expansion for infinit water depth case. Int J Offshore Polar Eng 21:161–168
20. Abramowitz M, Stegun I (1972) Handbook of mathematical functions with formulas, graphs, and mathematical tables, Tenth Printing. National Bureau of Standards Applied Mathematics Series 55. US Government Printing Office Washington