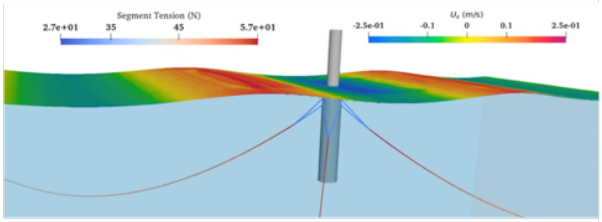
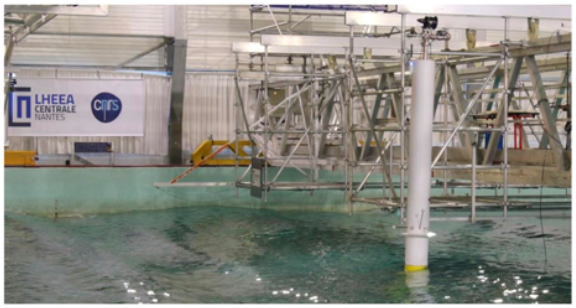
19e Journées de l'Hydrodynamique
26 – 28 novembre 2024, Nantes





SUR LA PRÉCISION DES CODES DE TENUE À LA MER LINÉAIRE PAR LA MÉTHODE DES ÉLÉMENTS AUX FRONTIÈRES

ON THE ACCURACY OF LINEAR BOUNDARY-ELEMENT-METHOD SEA-KEEPING CODES

Matthieu Ancellin
matthieu.ancellin@mews-labs.com
Eurobios Mews Labs, Cachan, France

Résumé

La tenue à la mer linéaire est le plus souvent calculée par des logiciels employant la méthode des éléments aux frontières. Des problèmes de précision sont parfois rencontrés par les utilisateurs de ces logiciels, en particulier les logiciels open-sources. Ce document est une vue d'ensemble des approximations numériques qui y sont faites et identifie les causes de certains de ces problèmes de précision. Les résultats présentés ont été obtenus avec le logiciel open-source Capytaine, ce document étant aussi une contribution à sa documentation théorique.

Summary

Linear sea-keeping analysis is usually computed with the help with software using the Boundary Element Method (BEM). Accuracy issues have been noticed by users of these software, especially users of open-source sofware. This paper is an overview of the different numerical approximation done in the implementation of these software and identifies the cause of some of the accuracy issues. The presented results have been computed with the open-source software Capytaine, this paper being also a contribution to its theory documentation.

I – Introduction

Linear potential flow models have been a critical part of the design of offshore floating devices for decades. Several commercial software, as well as a few free and open-source software, are available for the computation of hydrodynamical coefficients. The most widely used open-source software include Nemoh [6], HAMS [18] and Capytaine [3]. However, the accuracy and performance of these open-source software is often below their commercial counterparts. Since 2022, with the support of NREL and Sandia National Laboratories [17], the efficiency of the Boundary Element Method (BEM) solver Capytaine has been investigated and several improvements have been tested and implemented. Some results have been presented in [4] and [5].

The present paper gives an overview of the different approximations and transformations of the mathematical problem that are done to solve a first-order radiation-diffraction problem. A complete theory manual would contain at least three different "direct" methods and four different "higher-order" method, hence the need to clearly structure its content. This paper is thus structured in five sections corresponding to the five modelisation layers affecting the accuracy of the software. At each step, some alternative are reviewed and the effect on the final result is discussed. Although the examples focuses on the implementation in Capytaine, the same theory applies broadly to all BEM solver of this kind.

II – The five layers of BEM sea-keeping codes

Five steps are necessary to derive the typical algorithm of a BEM sea-keeping code:

1. The potential flow model is expressed as a Partial Differential Equation (PDE).
2. The PDE is reformulated as a Boundary Integral Equation (BIE) problem.
3. The BIE is discretized, typically using a colocation scheme.
4. The colocation scheme requires the integration of the Green function on each panel.
5. The numerical integration requires the evaluation of an efficient approximation of the Green function.

The five following sections detail each of these layers.

II – 1 Linear potential flow model

The first approximation, with which most users are already familiar, is the linear potential flow model for water waves. This model is documented in a lot of textbooks, including the reference works [20] and [21].

Airy wave model Assuming an incompressible inviscous and irrotational fluid, the velocity field u in the fluid can be represented as the gradient of an abstract scalar potential field ϕ, that is $u = \nabla\phi$. This potential is solution of a Laplace Partial Differential Equation (PDE) in the fluid, completed by boundary conditions, most notably on the free surface Γ_{FS}. Assuming small wave amplitude, the problem is treated as a small perturbation around the rest position. The fluid domain Ω and the free surface Γ_{FS} are assumed to the fixed in time and only the values of the fields in the fixed domain are evolving in

time. The small wave amplitude hypothesis also allows to linearize the wave boundary condition on the free surface:

$$g\frac{\partial\phi}{\partial x_3}+\frac{\partial^2\phi}{\partial t^2}=0,\quad \text{on } \Gamma_{FS} \tag{1}$$

where g is the gravity acceleration and x_3 is the vertical component of the position x.

Since the whole problem is linear in a fixed domain, it can be solved in the frequency domain. When using the same convention as in Capytaine and Nemoh, the complex-valued frequency-domain phasor $\hat{\phi}$ is defined as

$$\phi(x,t)=\mathrm{Re}\left(\hat{\phi}(x)e^{-i\omega t}\right) \tag{2}$$

where ω is the angular frequency of interest, leading to the following Laplace equation

$$\nabla^2\hat{\phi}=0,\quad \text{in } \Omega \tag{3}$$

and the following boundary condition on the fixed free surface at $z=0$:

$$g\frac{\partial\hat{\phi}}{\partial x_3}-\omega^2\hat{\phi}=0,\quad \text{on } \Gamma_{FS}. \tag{4}$$

Without a floating body, a solution to these equations is the Airy wave of amplitude a in direction β, given in infinite depth as

$$\hat{\phi}_{\text{Airy}}=-ai\frac{g}{\omega}\exp(kx_3)\exp(ik(x_1\cos\beta+x_2\sin\beta)), \tag{5}$$

where k is the wavenumber $k=\frac{g}{\omega^2}$.

Floating body hull For the wave-structre interaction problem, we denote by Γ the immersed hull of the floating body, which is supposed to be fixed like the rest of the domain. Integrating the pressure of the Airy wave field (5) on the hull gives the Froude-Krylov force. A more accurate account of the force on the body is obtained by solving (3) and (4) with a suitable Neumann boundary conditions on Γ.

The normal velocity of the fluid imposed as boundary condition on the hull depends on the kind of problem being solved. For radiation problems, the velocity is the velocity of the body for the degree of freedom causing the radiation. For diffraction problems, the velocity is the opposite of the velocity of the (5) in order to correct the Froude-Krylov force and get a total zero velocity on the hull. The problem is completed with a decay boundary condition at infinity.

Possible modifications of the recipe at this level:

- For the sake of brevity, **finite depth** is not discussed in this paper.
- The modelisation of **approximate forward speed** is done by modifying some of the equations at the level of description of the problem. Details can be found in [10], which was the base of the current implementation in Capytaine.
- The **mean drift force** and **QTFs** are the excitation force on the body computed with a second order approximation of the motion and free surface elevation. An open-source implementation is found in the latest version of Nemoh [15].

II – 2 Boundary integral equations

The partial differential equation defined above could be solved by a large variety of methods, including the Finite Difference Method as in [1]. The most common method in the literature is to rewrite the volumic problem as a Boundary Integral Equation (BIE) problem [11, 23].

For all points x on the ship hull Γ, the *direct BIE* (or "potential" formulation) reads

$$\frac{\hat{\phi}(x)}{2} + \iint_\Gamma \hat{\phi}(\xi)\ \nabla_2 G(x,\xi)\cdot n(\xi)\ d\xi = \iint_\Gamma \frac{\partial\hat{\phi}}{\partial n}(\xi)\ G(x,\xi)\ d\xi \tag{6}$$

where G is the fundamental solution (or *Green function*) satisfying

$$\nabla_1^2 G(x,\xi) = \delta(\xi - x).$$

with the boundary conditions on the free surface, on the sea bottom and at infinity. We denote by ∇_1 the derivative of the two-variable function G with respect to its first variable and ∇_2 the derivative with respect to its second variables (also denoted ∇_x and ∇_ξ in the literature). The direct BIE is applied notably in the reference commercial code WAMIT [16] and the open-source solver HAMS [18].

Nemoh [6] and Capytaine are using by default the *indirect BIE* (or "source" formulation), which reads

$$\hat{\phi}(x) = \iint_\Gamma \sigma(\xi) G(x,\xi)\ d\xi \qquad \frac{\partial\hat{\phi}}{\partial n}(x) = \frac{\sigma(x)}{2} + \iint_\Gamma \sigma(\xi)\ \nabla_1 G(x,\xi)\cdot n(x)\ d\xi \tag{7}$$

for all points x on the ship hull Γ, where G is the same *Green function* and σ is an intermediate variable without clear physical interpretation.

Although they are mathematically equivalent, these two BIE can exhibit very different behaviors in some cases. Sheng et al. [25] compared the results of WAMIT [16], HAMS [18] and Nemoh [6] on a few test cases. Nemoh was found to have inconsistent results on two test cases: the floating cylinder with thin damping plate and the floating cylinder with overlapping panels in mesh. By implementing both the direct and indirect boundary integral equation in Capytaine [5], the cause of the inconsistency has been identified: Nemoh is using the indirect boundary integral equation, while WAMIT and HAMS are using the direct boundary integral equation. Capytaine's results[1] for the overlapping mesh problem are displayed on Figure 1. The thin-plate case is also discussed more in depth in [5].

Despite the higher accuracy of the direct method, the indirect method has been kept as default in Capytaine, because it allows some post-processing such as reconstruction of the free surface elevation or reconstruction of the velocity field. The latter is necessary for forward speed calculation, thus forward speed is only available in Capytaine with the indirect solver.

[1] These results have been computed with the development version of Capytaine 2.2.1, as earlier versions of the direct boundary integral equations had a bug when handling symmetric meshes.

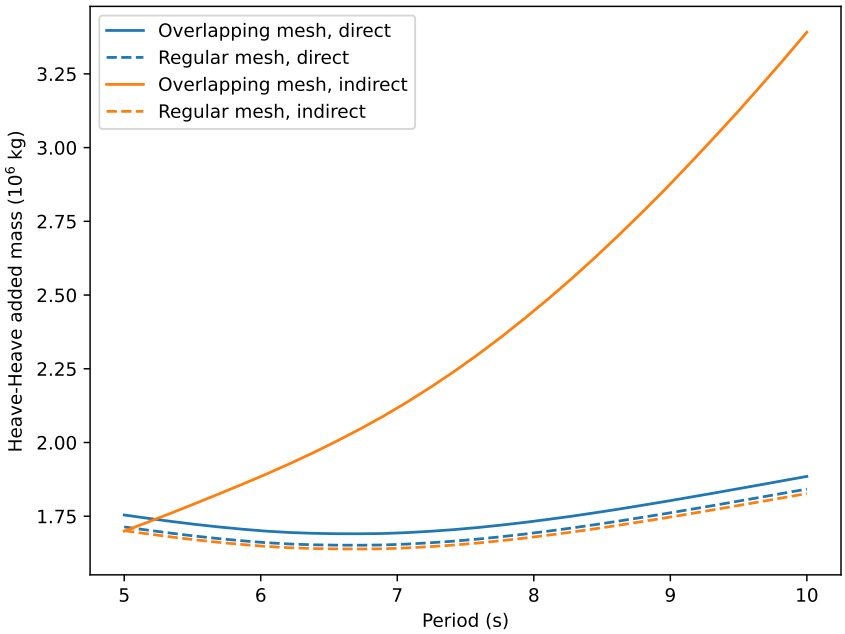


Figure 1: Heave-heave added mass of a floating vertical cylinder (radius 10 m and draft 10 m) for several wave periods, computed with the overlapping mesh of [25] (1536 panels) and a regular mesh of similar accuracy (1680 panels), using the two boundary integral equations implemented in Capytaine (direct and indirect).

Possible modifications of the recipe at this level:

- One of the consequences of the use of Boundary Integral Equation is the appearence of **irregular frequencies**, that is frequencies at which the BIE problem is ill-posed, perturbing the results in the neighborhood of these frequencies [14]. Several modifications of the BIE can be used to avoid this issue. The most common one (recently implemented in Capytaine) is the lid-based method involving an extension of the domain boundary on the free surface inside the hull [12]. Another method is the Burton-Miller method involving the second-kind boundary integral equations, which are yet other variants of the boundary integral equations presented above.

- The modelling of **thin plates** with dipoles elements is another possible modification of the BIE. The method is implemented notably in WAMIT [16] but no open-source implementation is available.

II – 3 Discretization

As already hinted in the previous section, the numerical resolution of the Boundary Integral Equations involves their discretization on a mesh of the immersed hull Γ.

The most common approach (used e.g. in WAMIT low-order solver, HAMS, Nemoh and Capytaine) is the discretization with a colocation scheme. The value of each fields (e.g. $\hat{\phi}(x)$) is assumed to be constant on each panel of the mesh, equal to the value at the center of the panel, and the solver manipulates a vector (still denoted e.g. $\hat{\phi}$) with as many components as the number of panels:

$$\hat{\phi}_i = \hat{\phi}(x_i), \qquad \left(\frac{\partial \hat{\phi}}{\partial n}\right)_i = \frac{\partial \hat{\phi}}{\partial n}(x_i), \qquad \dots$$

where x_i is the center of panell Γ_i.

With this discretization, the direct BIE is rewritten as a linear algebra problem

$$D\hat{\phi} = S\frac{\partial\hat{\phi}}{\partial n} \tag{8}$$

where

$$S_{ij} = \iint_{\Gamma_j} G(x_i, \xi)\ \mathrm{d}\xi, \qquad D_{ij} = \frac{\delta_{ij}}{2} + \iint_{\Gamma_j} \nabla_2 G(x_i, \xi)\cdot n(\xi)\ \mathrm{d}\xi \tag{9}$$

where Γ_j is the jth panel of the mesh and x_i is the center of the ith panel.

The indirect BIE is discretized as

$$\hat{\phi} = S\sigma, \quad \frac{\partial\hat{\phi}}{\partial n} = K\sigma, \tag{10}$$

where

$$S_{ij} = \iint_{\Gamma_j} G(x_i, \xi)\ \mathrm{d}\xi, \qquad K_{ij} = \frac{\delta_{ij}}{2} + \iint_{\Gamma_j} \nabla_1 G(x_i, \xi)\cdot n(x)\ \mathrm{d}\xi. \tag{11}$$

The resolution of the problem involves the evaluation of the coefficient of the matrices S and D (resp. S and K for the indirect BIE) and the resolution of a linear system. This linear system is of size $n_{\text{panels}} \times n_{\text{panels}}$. This quadratic cost in CPU and RAM is the main contrainst in the performance of BEM solvers.

Possible modifications of the recipe at this level:

- **Higher order discretizations** of the BIE can be used. Application to linear potential flow are few (see e.g. [26]) and no open-source implementation is available to our knowledge.
- The **symmetries** of the problem are causing regularities in the shape of the matrices, which can be exploited to reduce the computation time [2]. Most notably, the symmetries with respect to vertical symmetry planes are implemented in most software.
- Since the size of the matrices is one of the main bottleneck for BEM solver, **approximation techniques** have been developped. The most famous is the Fast Multipole method (FMM), implemented for linear potential flow in [7]. In Capytaine, a prototype of H-matrices [4] has been tested.

II – 4 Integration over a panel

In this section and the next, the approximations are more technical and usually not encountered by most users. In Capytaine, they occur in the Fortran core of the software. In the previous section, the problem has been discretized using matrices containing the integrals of the Green function over faces of the mesh. Let us now discuss how to evaluate these integrals.

Firstly, let us notice that the (infinite depth) Green function is usually written by isolating the singular $\frac{1}{|x-\xi|}$-terms, usually refered to as Rankine terms. The singular Rankine

terms are integrated separately from the remaining terms. The integral of the Rankine terms is done with exact expressions of the integral of $1/|x-\xi|$ and its derivative on a quadrilateral [8]. The integral of the remaining terms is done by numerical quadratures [24], with often only a single point (e.g. in Capytaine and HAMS). Note that other singularities might appears in the derivatives of G, but are not discussed here for the sake of brevity.

Although most software agree on the above methodology, they differ in the way they extract the singularities (see also [28]):

- In WAMIT, HAMS and others, the Green function is written as:

$$-4\pi G(x,\xi) = \frac{1}{|x-\xi|} + \frac{1}{|x-s(\xi)|} + k\,\mathcal{G}^+(kr,kz) \tag{12}$$

  where s denotes the symmetry with respect to the free surface, that is $s(\xi) = (\xi_1, \xi_2, -\xi_3)$ assuming the free surface is at $x_3 = 0$, and r and z are defined as

$$r = \sqrt{(x_1-\xi_1)^2 + (x_2-\xi_2)^2}, \qquad z = x_3 + \xi_3.$$

- In Nemoh and former versions of Capytaine, the Green function is written as:

$$-4\pi G(x,\xi) = \frac{1}{|x-\xi|} - \frac{1}{|x-s(\xi)|} + k\,\mathcal{G}^-(kr,kz) \tag{13}$$

The former may be refered to as *low-frequency asymptotics* because its singularities corresponds to the Green function in the infinite frequency (or 0-wavenumber) limit:

$$k\,\mathcal{G}^+(kr,kz) \to_{k\to 0} 0, \tag{14a}$$

and conversely the latter may be refered to as *high-frequency asymptotics* because

$$k\,\mathcal{G}^-(kr,kz) \to_{k\to\infty} 0. \tag{14b}$$

Both variants are now implemented in Capytaine. Since version 2.2, the default has been changed from the *high-frequency* variant to the *low-frequency* variant. While both choices appear justified by the asymptotics (14), the *low-frequency* is more convenient in practice. It is especially more accurate near the free surface[2] where the accuracy of the Green function is the most critical. On Figure 2, the added mass of a sphere almost tangent to the free surface has been computed with both variants. Both variants converges to the same solution (also found with WAMIT), but the *low-frequency* variant converges much faster, due to its higher accuracy for horizontal panels near the free surface.

Possible modifications of the recipe at this level:

- Despite the extraction of the Rankine terms, the remaining term $\mathcal{G}^+$ is still singular on the free surface.

$$\mathcal{G}^+(kr,kz) \sim_{kr,kz\longrightarrow(0,0)} -2\log\left(\sqrt{k^2r^2+k^2z^2} - kz\right) + (\gamma - \log(2)) - 2\pi i \tag{15}$$

[2] A hand-wavy explanation is that low-frequencies corresponds to long wavelengths, and in long wavelengths everything is near the free surface with respect to the wavelength.

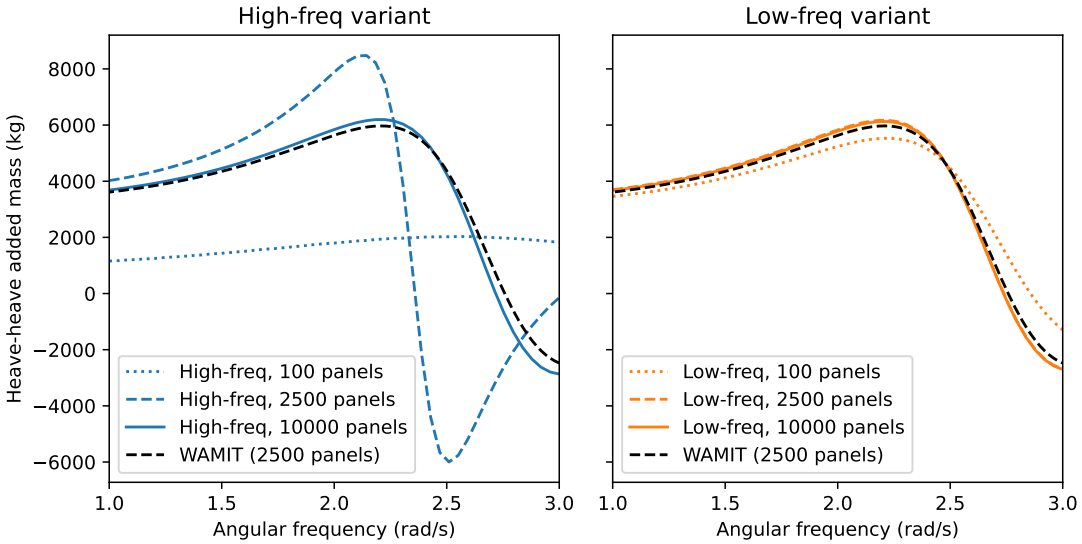


Figure 2: Added mass of floating sphere of radius 1 m, tangent to the free surface (that is its center is at $x_3 = 1.01\,\mathrm{m}$), depending on the angular frequency ω, computed with different mesh discretizations between 100 and 10000 panels and with different implementations of the Green function. "Low-freq" refers to the integration with the Rankine terms as in (12), while "high-freq" refers to (13).

Panels on the free surface require the integration of the **singular logarithmic term**. An analytical expression, similar to the one mentionned above for the Rankine term, is presented in [19]. In Capytaine, an approximate method has been implemented for horizontal panels on the free surface (where $x_3 = 0$): the integral of $\log(kr)$ on a quadrilateral panel is approximated as the integral of the same function on a disk of same center and area than the original quadrilateral.

$$\iint_{\Gamma_j} \log(kr) d\xi \simeq \int_0^{2\pi} \int_0^{r_j} \log(kr) r dr d\theta = \frac{\pi}{2} r_j^2 \left(\log(k^2 r_j^2) - 1\right) \tag{16}$$

where r_j is the radius of the disk of same area as Γ_j, that is $\pi r_j^2 = |\Gamma_j|$.

- An **higher-order quadrature** (four points instead of one) is used in the latest version of Nemoh [15]. From our experience, the increase in computation time is not worth the gain in accuracy. A good compromise could be to increase the quadrature order only on some faces, for instance near the free surface where the Green function varies the most.

II – 5 Evaluation of the Green function

In the previous section, we started to present the expression of the Green function, but it has not been fully expressed. Let us now discuss the computation of the remaining terms $\mathcal{G}^+$ or $\mathcal{G}^-$ for the infinite-depth Green function.

Recalling the following definition for the infinite-depth Green function

$$-4\pi G(x, \xi) = \frac{1}{|x - \xi|} - \frac{1}{|x - s(\xi)|} + k\,\mathcal{G}^-(\tilde{r}, \tilde{z})$$

with

$$\tilde{r} = k\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2}, \qquad \tilde{z} = k(x_3 + \xi_3),$$

then the wave term can be written in the following form (among others [28]):

$$\mathcal{G}^-(\tilde{r},\tilde{z}) = \frac{4}{\pi}\Re\left(\int_0^{\pi/2} e^{\zeta(\theta)}\left(E_1(\zeta(\theta)) + i\pi\right) - \frac{1}{\zeta(\theta)}\,\mathrm{d}\theta\right) + 4i\Re\left(\int_0^{\pi/2} e^{\zeta(\theta)}\,\mathrm{d}\theta\right) \tag{17}$$

where ζ is defined as

$$\zeta(\theta) = \tilde{z} + i\tilde{r}\cos\theta,$$

and E_1 is the exponential integral defined as

$$E_1(\zeta) = \int_\zeta^\infty \frac{e^{-t}}{t}\mathrm{d}t.$$

A similar expression can be derived for $\mathcal{G}^+$ and for its gradient (not discussed here, for the sake of brevity).

The above expression is well-defined and can be easily computed numerically by discretizing the integral with respect to θ. An accurate evaluation of these integrals takes 10 µs to 100 µs. Given that the Green function typically needs to be evaluated millions of times for each BEM resolutions, a faster approximation needs to be found. Many surrogate models have been presented in the literature [28]. Nemoh and Capytaine are using the method proposed by G. Delhommeau [9]. Before the start of the resolution, a grid of points of $\tilde{r}$ and $\tilde{z}$ is constructed and the value of the Green function and its derivative is evalutated at each point. Then, during the resolution, a second-order Lagrange polynomial interpolation on a 3×3 stencil is used to compute the Green function (taking approximately 100 ns, that is 100 to 1000 times faster than the full evaluation).

The accuracy of the surrogate model is thus constrained by the density of the tabulation. Earlier version of Capytaine used the same tabulation as in Nemoh 2, composed of 328×46 points. This appeared to be insufficient and lead to high error in some cases (as in the case of Figure 4 below). In Nemoh 3 a denser tabulation has been used, while in Capytaine it is now possible for the user to specify a custom density for the tabulation (the default matches approximately the one of Nemoh 3). On Figure 3, the error introduced by the tabulation is plotted as a function of $\tilde{r}$ and $\tilde{z}$. The gain from version 1.5 to version 2.1 is due to the finer tabulation, while the difference between version 2.1 and 2.2 is due to the tabulation of the *low-frequency* variant of the Green function $\mathcal{G}^+$ instead of the *high-frequency* variant $\mathcal{G}^-$. Tabulation of $\mathcal{G}^+$ is worse for large $\tilde{r}$ and $\tilde{z}$ (that is high frequencies) but better for small $\tilde{r}$ and $\tilde{z}$ (that is low-frequency), as expected from (14).

Finally, Figure 4 presents the added mass of the DeepCWind floating offshore wind turbine reference design [22] computed in Capytaine with several variant discussed in this section and the previous one. The lower-resolution tabulation of previous version of Capytaine (green curved labelled *Legacy*) lead to very noisy results, which are greatly improved by the higher-resolution tabulation. Contrary to the test case of Figure 2, the *high-frequency* Green function integration variant appears slightly more accurate here than the *low-frequency* variant. However, this small high-frequency noise seems to us less significant than the discrepancy of Figure 2.

Possible modifications of the recipe at this level:

- In the latest version of Nemoh [15], a **higher-order polynomial interpolation** method is used (fifth order Lagrange interpolation). This has not been implemented in Capytaine, where we prefer to investigate refinements of the tabulation that would allow higher accuracy without the higher computational cost of the higher-order polynomial interpolation.

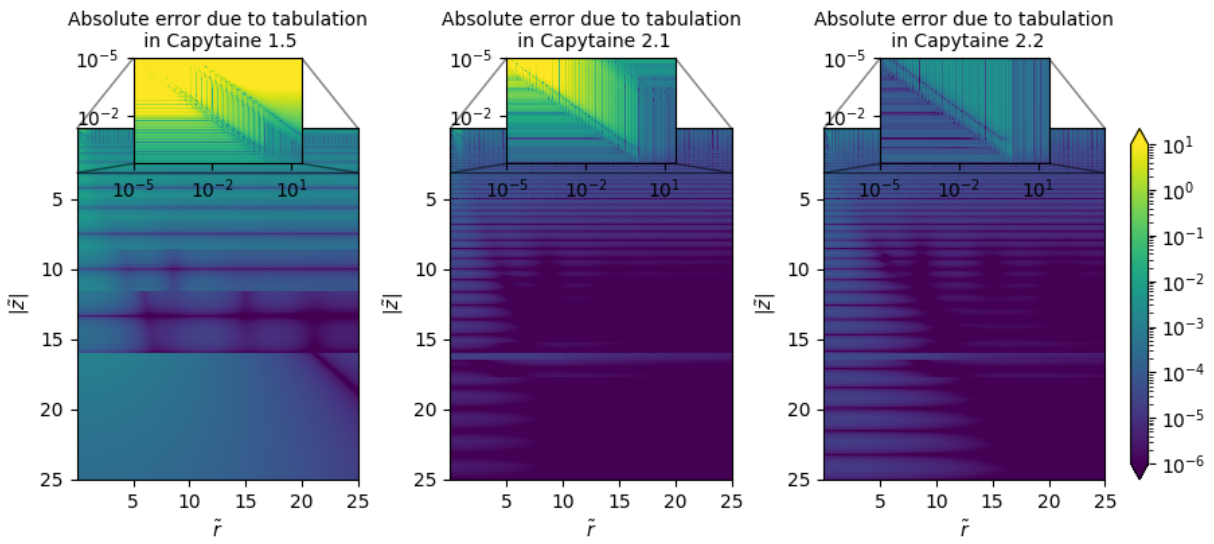


Figure 3: Error on the Green function introduced by different versions of the tabulation. (Actually, all computations have been done with the development version of Capytaine 2.2.1, with parameters mimicking the implementation of previous versions.)

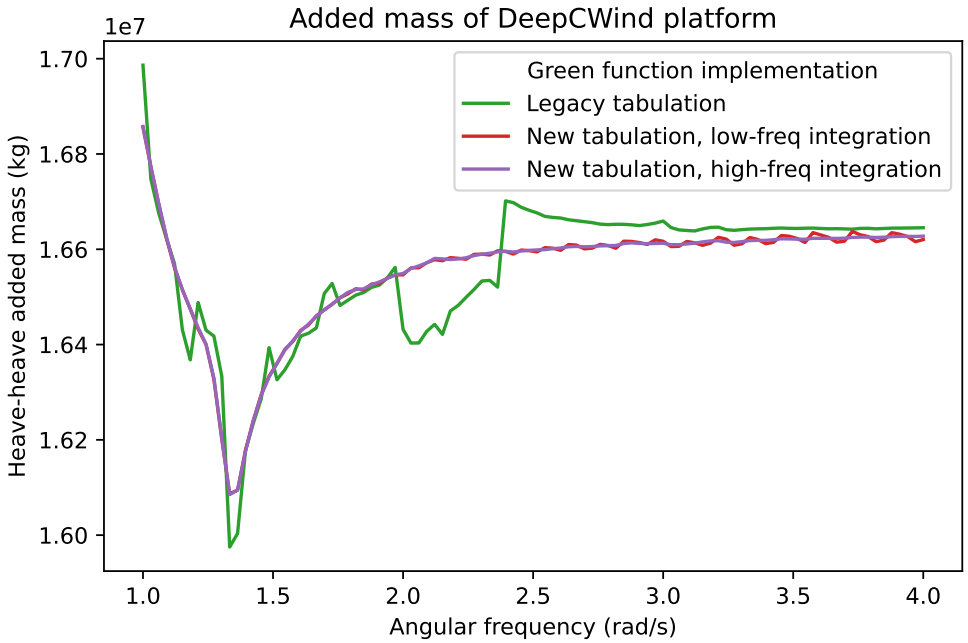


Figure 4: Added mass of the DeepCWind, depending on the angular frequency ω, computed with several variants of the implementation of the Green function. *Legacy tabulation* refers to Capytaine 1.5 (and Nemoh 2) implementation while the *new tabulation* refers to the tabulation of version 2.2, applied to the two variants of the Green function integration discussed at Section II – 4.

- A large number of **other surrogate models** are presented in [28]. While polynomial-type models are the most common, some research works propose original methods such as Differential-Equation-based models [27] or Machine-Learning-based models [13].
- The **finite-depth** version of the Green function deserves a long discussion, but is not tackled here for the sake of brevity.

III – Conclusion

This paper is a walk through the numerical techniques used for the discretization of linear potential flow problems with the Boundary Element Method. The end goal would be to tune of the precision of each layer in order to optimize the computation time of the solver for a given total precision. But, depending of the test case, the bottleneck of accuracy may come from slightly different places and unexpected side-effect might occur. By implementing several variants and allowing to test many combinations of solver parameters, Capytaine will hopefully allow to better understand and optimize linear BEM sea-keeping codes.

Acknowlegment

This work has been funded by the Alliance for Sustainable Energy, LLC, Managing and Operating Contractor for the National Renewable Energy Laboratory (NREL) for the U.S. Department of Energy.

References

[1] M. Amini-Afshar and H. B. Bingham. Solving the linearized forward-speed radiation problem using a high-order finite difference method on overlapping grids. *Applied Ocean Research*, 69:220–244, 2017.

[2] M. Ancellin and F. Dias. Using the floating body symmetries to speed up the numerical computation of hydrodynamics coefficients with Nemoh. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 51302, page V009T13A031. American Society of Mechanical Engineers, 2018.

[3] M. Ancellin and F. Dias. Capytaine: a Python-based linear potential flow solver. *Journal of Open Source Software*, 4(36):1341, 2019.

[4] M. Ancellin, P. Marchand, and F. Dias. Towards high-performance linear potential flow BEM solver with low-rank compressions. *Energies*, 17(2):372, 2024.

[5] M. Ancellin and N. Nguyen. Case studies of BEM solver accuracy with the open-source code Capytaine. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 87851, page V007T09A076. American Society of Mechanical Engineers, 2024.

[6] A. Babarit and G. Delhommeau. Theoretical and numerical aspects of the open source BEM solver NEMOH. In *11th European wave and tidal energy conference (EWTEC2015)*, 2015.

[7] B. Borgarino, A. Babarit, and P. Ferrant. An implementation of the fast multipole algorithm for wave interaction problems on sparse arrays of floating bodies. *Journal of Engineering Mathematics*, 77:51–68, 2012.

[8] G. Delhommeau. *Problèmes de diffraction-radiation et de résistance des vagues : étude théorique et résolution numérique par la méthode des singularités*. PhD thesis, 1987.

[9] G. Delhommeau. The seakeeping codes aquadyn and aquaplus. In *19th WEGEMT School, Numerical Simulation of Hydrodynamics: Ships and Offshore Structures*, 1993.

[10] L. Donatini, I. Herdayanditya, G. Verao Fernandez, A. B. K. Pribadi, E. Lataire, and G. Delefortrie. Implementation of forward speed effects on an open source seakeeping solver. In *6th MASHCON: International Conference on Ship Manoeuvring in Shallow and Confined Water*, pages 20–33. Knowledge Centre for Manoeuvring in Shallow and Confined Water, 2022.

[11] L. Gaul, M. Kögl, and M. Wagner. *Boundary element methods for engineers and scientists: an introductory course with advanced topics*. Springer Science & Business Media, 2013.

[12] I. Herdayanditya, B. Baert, F. Stempinski, P. Rauwoens, and E. Lataire. Application of irregular frequency removal feature in Capytaine for anode cage installation over a monopile wind turbine. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 87820, page V05AT06A063. American Society of Mechanical Engineers, 2024.

[13] S. Huang, R. Zhu, H. Chang, H. Wang, and Y. Yu. Machine learning to approximate free-surface Green's function and its application in wave-body interactions. *Engineering Analysis with Boundary Elements*, 134:35–48, 2022.

[14] T. Kelly, I. Zabala, Y. Pena-Sanchez, J. Ringwood, J. Henriques, and J. M. Blanco. A post-processing technique for removing 'irregular frequencies' and other issues in the results from BEM solvers. *International Journal of Marine Energy*, 5(1):123–131, 2022.

[15] R. Kurnia and G. Ducrozet. Nemoh: Open-source boundary element solver for computation of first- and second-order hydrodynamic loads in the frequency domain. *Computer Physics Communications*, 292:108885, 2023.

[16] C.-H. Lee and J. Newman. Computation of wave effects using the panel method. *WIT Transactions on State-of-the-art in Science and Engineering*, 18, 2005.

[17] J. Leon, J. Grasberger, D. Forbush, M. Sirigu, M. Ancellin, N. Tom, A. Keester, K. Ruehl, D. Ogden, and S. Husain. Advanced features and recent developments in the WEC-Sim open-source design tool. *National Renewable Energy Laboratory (NREL), Golden, CO (United States)*, 2024.

[18] Y. Liu. HAMS: A frequency-domain preprocessor for wave-structure interactions—theory, development, and application. *Journal of Marine Science and Engineering*, 7(3):81, 2019.

[19] Y. Liu and J. M. Falzarano. Suppression of irregular frequency effect in hydrodynamic problems and free-surface singularity treatment. *Journal of Offshore Mechanics and Arctic Engineering*, 139(5):051101, 2017.

[20] B. Molin. *Offshore structure hydrodynamics*. Cambridge University Press, 2023.

[21] J. N. Newman. *Marine hydrodynamics*. The MIT press, 2018.

[22] A. Robertson, J. Jonkman, M. Masciola, H. Song, A. Goupee, A. Coulling, and C. Luan. Definition of the semisubmersible floating system for phase II of OC4. 2014.

[23] S. A. Sauter, C. Schwab, S. A. Sauter, and C. Schwab. *Boundary element methods*. Springer, 2011.

[24] J. R. Saverin and L. Grüter. Quadrature schemes for the integration of the free-surface Green's function within the open-source boundary element method library BEMUse. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 85925, page V007T08A057. American Society of Mechanical Engineers, 2022.

[25] W. Sheng, E. Tapoglou, X. Ma, C. Taylor, R. Dorrell, D. Parsons, and G. Aggidis. Hydrodynamic studies of floating structures: Comparison of wave-structure interaction modelling. *Ocean Engineering*, 249:110878, 2022.

[26] P. Wuillaume, L. Letournel, F. Rongere, and C. Chauvigné. Développement d'un code de diffraction-radiation basé sur une discrétisation linéaire du potentiel de vitesse comparaison avec la méthode des panneaux constants. *Actes des 17èmes Journées de l'Hydrodynamique, Cherbourg*, 2020.

[27] C. Xie, X. Chen, A. H. Clément, and A. Babarit. A new ordinary differential equation for the evaluation of the frequency-domain Green function. *Applied Ocean Research*, 86:239–245, 2019.

[28] C. Xie, Y. Choi, F. Rongère, A. H. Clément, G. Delhommeau, and A. Babarit. Comparison of existing methods for the calculation of the infinite water depth free-surface Green function for the wave–structure interaction problem. *Applied Ocean Research*, 81:150–163, 2018.