Acceleration of a wave-structure interaction solver by the Parareal method

Yohan Poirier, Julien Salomon, Aurélien Babarit, Pierre Ferrant, Guillaume Ducrozet

To cite this version:

Yohan Poirier, Julien Salomon, Aurélien Babarit, Pierre Ferrant, Guillaume Ducrozet. Acceleration of a wave-structure interaction solver by the Parareal method. Engineering Analysis with Boundary Elements, 2024, 167, pp.105870. 10.1016/j.enganabound.2024.105870 . hal-04647112

HAL Id: hal-04647112
https://hal.science/hal-04647112

Engineering Analysis with Boundary Elements 167 (2024) 105870

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Acceleration of a wave-structure interaction solver by the Parareal method

Yohan Poirier [a,*], Julien Salomon [b,c], Aurélien Babarit [a], Pierre Ferrant [a], Guillaume Ducrozet [a]

[a] Nantes Université, École Centrale Nantes, CNRS, LHEEA, UMR 6598, F-44000 Nantes, France
[b] Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, France
[c] ANGE, INRIA Paris, Paris, France

ARTICLE INFO

Keywords:
Time parallelization
Wave-structure interactions
Weak-Scatterer

ABSTRACT

Potential flow theory-based solvers are commonly used in ocean engineering to investigate the interactions between ocean waves and floating bodies. Depending on assumptions, several methods have been proposed. Among them, the Weak-Scatterer method is an interesting trade-off in the sense that this approach is not limited in theory by the small wave amplitudes and small body motions assumptions of linear methods. Moreover, this approach is in practice more stable than the fully non-linear methods. An implementation of the Weak-Scatterer method is the WS-CN code (Letournel, 2015; Chauvigné, 2016; Wuillaume, 2019).

The computational time of the WS-CN code which is considered in the present study is relatively long for engineering purposes. In order to reduce it, the present paper presents an implementation of the Parareal method in the WS-CN code. The Parareal method is an algorithm for parallelizing a simulation in time that can accelerate the complete simulation (Lions, 2001) . This is a key difference in comparison to other acceleration techniques which have been studied in the literature (e.g. the Fast Multipole Method (FMM), the precorrected Fast Fourier Transform (pFFT) method, …). To the authors' knowledge, the present study is the first to couple the Parareal method to a potential flow theory-based wave-structure interaction solver. It is shown that the method can significantly reduce the computational time for small wave steepness, but that the performance decreases rapidly with increasing steepness.

1. Introduction

Many numerical methods have been developed for modeling wave-structure interactions. The most commonly used for engineering studies of a floating body (such as an oil and gas offshore platform, an offshore wind turbine or a wave energy converter) are the ones based on the boundary element method. A comprehensive review of boundary element methods for hydrodynamic modeling of wave energy systems is presented by Papillon et al. [1]. Among these methods, those based on the linearized potential flow theory are often preferred because they are very fast and because they have been found to be sufficiently accurate for many practical applications. Examples of popular numerical codes based on linear potential flow theory are WAMIT [2], ANSYS-AQWA [3] or NEMOH [4].

The linearized potential flow theory is based on the assumptions that the fluid is inviscid and incompressible, that the flow is irrotational, that the wave amplitude is small with respect to both the wavelength and the dimensions of the structure, and that the motion of the floating body is small in comparison to the dimensions of the structure. Therefore, this model is poorly suited in case of large amplitude motion of the body and/or in heavy seas (waves of large amplitude). In principle, Navier–Stokes solvers could be used to deal with such problems. However, Navier–Stokes solvers are associated with high computational time (it can exceed hours per wave period [5,6]) which hampers its use for daily engineering studies. Moreover, obtaining and maintaining a domain discretization of sufficient quality throughout the simulation can be challenging, particularly in the case of large-amplitude motions [7].

Alternative approaches for the study of a floating body with large amplitude motion and/or in heavy seas are the fully non-linear potential theory and the Weak-Scatterer method [8]. In the fully non-linear potential method, no assumptions are made about the amplitude of the incident wave or the amplitude of the motion of the body. Guerber et al. [9] developed a two-dimensional NWT (Numerical Wave Tank) using a High-Order Boundary Element Method based on a fully non-linear method to study the interactions between an incident wave and a fully immersed cylinder. This method appears to preserve fluid volume and energy at a high level of accuracy without using very fine spacial discretization. This model was then adapted by Dombre et al. [10] to study the case of a rigid body piercing the free surface. However, non-linear BEM methods only allow to study potential flows

---

* Corresponding author.
E-mail address: yohan.poirier@ac-poitiers.fr (Y. Poirier).

https://doi.org/10.1016/j.enganabound.2024.105870
Received 2 August 2023; Received in revised form 29 June 2024; Accepted 29 June 2024

and therefore do not allow the description of turbulent, rotational or viscous effects that can occur in the vicinity of submerged structures. By distinguishing between fluid regions close to interacting structures and regions that are sufficiently far away, the coupled use of a CFD solver and a potential solver makes it possible to benefit simultaneously from the advantages of both methods [11–13]

In contrast to the linearized potential flow theory, the Weak-Scatterer method [8] only assumes that the scattered waves due to the interaction of the body with the incident waves are small. This allows the free surface to be discretized at the position of the non-linear incident wave instead of the total free surface, which is expected to improve the robustness with respect to fully non-linear methods (avoiding for example to have to deal with wave breaking). This approach may also allow the use of coarser meshes, and therefore reduce the computational time. Zhang et al. [14] used a higher-order boundary element method based on the Weak-Scatterer method to study the performance of a point absorber wave energy converter. Comparisons of numerical simulation with experimental results showed good agreement. The method made it possible to study the effects of wave steepness or Power Take Off settings on the performance of the converter. Tong et al. [15] used a Harmonic Polynomial Method (HPC) based on a generalized Weak-Scatterer method, in which the boundary conditions takes into account the fact that the free surface moves tangentially to the surface of the piercing bodies. A good agreement was obtained with experimental results, fully non-linear simulations and theoretical solutions.

In previous work at École Centrale de Nantes, a numerical tool (WS-CN) has been developed based on the Weak-Scatter method [16–18]. The ambition is to achieve a fast and reliable engineering tool. With the current implementation, despite being smaller than Navier–Stokes solvers, computational times were found to be relatively long (between 10^2 and 10^3 s per wave period for a mesh with about 10^3 nodes). Therefore, for practical use, it has been deemed necessary to find ways of reducing the computation time.

In BEM methods, most of the computational cost is associated with the assembling and the solving of a dense linear system of equations. Naive implementations typically result in complexity of $O(N^3)$ with N the number of unknowns. Methods have been developed to reduce the computational cost. They include the fast multipole method [19], the pFFT method [20] and the use of $\mathcal{H}$-matrix [21]. The fast multipole method (FMM) allows theoretically to reach complexity of $O(N\log N)$ and even $O(N)$ in some cases. The principle of this method is to use a multipole expansion in order to group the source points close enough to each other and thus to be able to treat their influence on the distant points as that of a single source point. The precorrected Fast Fourier Transform (pFFT) method consists in building a uniform grid in which the long distance interactions are good approximations of those of the initial mesh. This uniform grid allows then to compute quickly the long distance interactions between the different points by performing a convolution product using a Fourier transform. It allows complexities of $O(N\log(N))$ to be reached. Teng and Song [22], for example, used this method to accelerate a BEM solver. They showed that it can save both computational time and memory space. Finally, the principle of the Hierarchical Matrix ($\mathcal{H}$-Matrix) method is to approximate the matrix associated with the linear system with a sparse matrix. This procedure is carried out by identifying sub-matrices of sufficiently low rank k in the original matrix. Doing so, the complexity reduces to $O(Nk^2\log(N))$.

Those methods allow to decrease the computational time associated with the building and the solving of the linear system. However, a complete simulation involves other computations (e.g. mesh management) whose computational time can be significant (they may even become dominant if the methods mentioned above, FMM, pFFT or $\mathcal{H}-matrices$, have been implemented). Conversely, the Parareal method allows the complete simulation to be accelerated. Indeed, this method consists in dividing the time interval of the simulation into sub-intervals in which the simulation is carried out in parallel by different processors [23]. The Parareal method has been implemented in various fields of physics such as quantum chemistry [24], hydrodynamics (simulation of system described by the Navier–Stokes equation [25,26] or free surface wave propagation [27,28]) and electromagnetism (simulation of turbulent plasmas [29]).

To our knowledge, there have been no attempt to implement the Parareal method in a wave-structure interaction solver based on the Weak-Scatterer method. Therefore, in this study, we describe an implementation of the parareal method in the WS-CN code and we investigate its effect on accuracy and computation time.

The remainder of the present paper is structured as follows: In Section 2 we describe the wave-structure interaction model and its present implementation in the WS-CN code. Section 3 is devoted to the description of the Parareal method and the implementation in the WS-CN code. Finally, we present in Section 4(i) a verification study and (ii) sensitivity studies of the speed-up factor as function of to the wave amplitude and the number of processors.

2. Wave-structure interaction model

The aim of this section is to describe how the WS-CN code works and to study its computation times before time parallelization.

2.1. Mathematical formulation

The wave-structure interaction problem is illustrated in Fig. 1. The wet body surface, i.e. the part of the body surface which is in the water, is denoted Γ_B. The part in air is denoted $\bar{\Gamma}_B$. The free surface is denoted Γ_{FS}. The seabed is Γ_{SB}. The lateral boundaries are denoted Γ_{LB}. The inner fluid domain Ω is such as $\partial\Omega = \Gamma_B \cup \Gamma_{FS} \cup \Gamma_{SB} \cup \Gamma_{LB}$. A point in the domain is identified by its Cartesian coordinates (x, y, z) where z is the altitude with respect to the mean free surface. In the absence of breaking waves, the altitude of a point on the free surface is given by a monovalent function $z = \eta(x, y, t)$ called elevation.

For sake of brevity, the derivation of the mathematical model is omitted in this paper. Only the key concept are presented. The details can be found in e.g. [30]. In the following, vectors will be indicated in bold ($\mathbf{v}$). A dot on a quantity indicates the derivative with respect to time of this quantity: $\dot{u} = \frac{\mathrm{d}u}{\mathrm{d}t}$.

The weak-scatterer method is based on potential flow theory, in which the fluid is assumed to be inviscid and incompressible, and the flow to be irrotational. With these assumptions, it can be shown that the flow velocity $\mathbf{V}$ derives from a velocity potential ϕ:

$$\mathbf{V} = \boldsymbol{\nabla}\phi \tag{1}$$

and that the velocity potential follows Laplace's equation:

$$\Delta\phi = 0. \tag{2}$$

The material boundaries (body and seabed) are assumed to be impermeable. The velocity of the fluid in a direction $\mathbf{n}$ perpendicular to these surfaces ($\mathbf{V}.\mathbf{n} = \frac{\partial\phi}{\partial n}$), therefore satisfies the conditions:

$$\frac{\partial\phi}{\partial n} = 0 \qquad \forall\mathbf{M} \in \Gamma_{SB} \tag{3}$$

$$\frac{\partial\phi}{\partial n} = \dot{\mathbf{x}}_\mathbf{b}.\mathbf{n} \qquad \forall\mathbf{M} \in \Gamma_B \tag{4}$$

where $\dot{\mathbf{x}}_\mathbf{b}(\mathbf{M}) = \dot{\mathbf{x}}_\mathbf{b}(\mathbf{G}) + \boldsymbol{\omega}_\mathbf{b} \times \mathbf{GM}$ is the body velocity, $\mathbf{n}$ is the unit vector normal to the body surface and pointing to the fluid and $\boldsymbol{\omega}_\mathbf{b}$ is the body angular velocity.

In this paper, only the case of a fixed body is considered ($\dot{\mathbf{x}}_\mathbf{b}(\mathbf{G}) = \mathbf{0}$ and $\boldsymbol{\omega}_\mathbf{b} = \mathbf{0}$) for sake of simplicity. However, note that the WS-CN code allows dealing with both prescribed motion and free motion of the body [16–18].

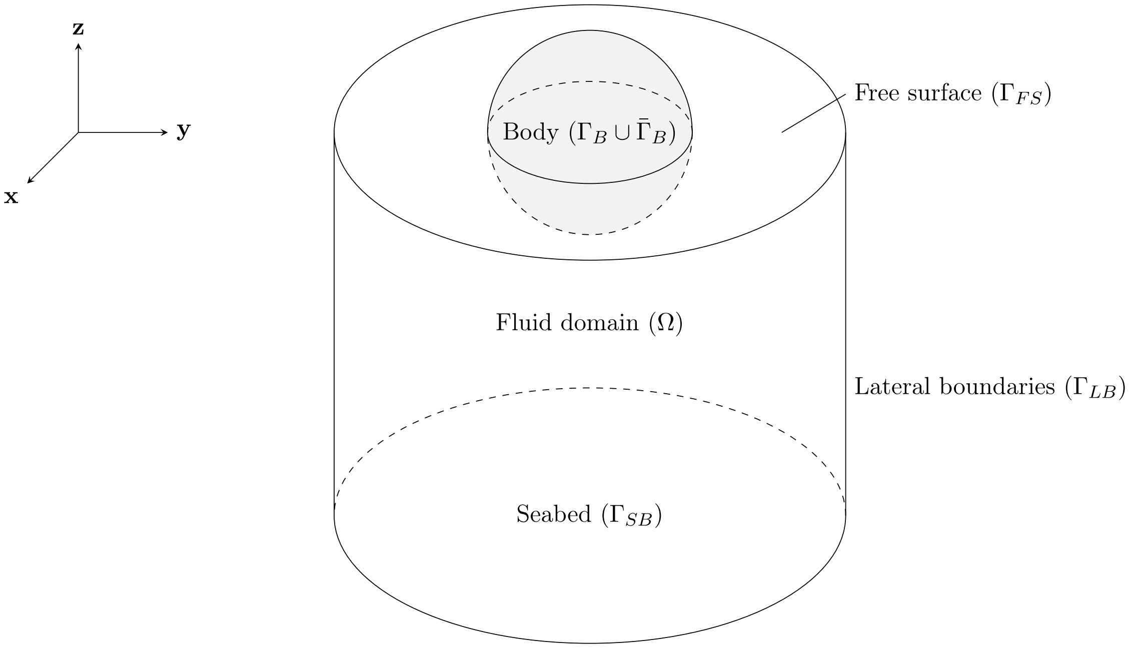

Fig. 1. Illustration of the wave-structure interaction problem.

A fluid particle located at the free surface shall remain there. The relationship derived from this constraint is called the kinematic free surface boundary condition (KFSBC):

$$\frac{\partial \eta}{\partial t} = \frac{\partial \phi}{\partial z} - \nabla\phi . \nabla\eta \text{ for } z = \eta(x, y, t). \tag{5}$$

The pressure discontinuity across the free surface due to surface tension effects is neglected. Thus, Bernoulli's theorem leads to the dynamic free surface boundary condition (DFSBC):

$$\frac{\partial \phi}{\partial t} = -g\eta + \frac{1}{2}\|\nabla\phi\|^2 + \frac{p_a}{\rho} \text{ for } z = \eta(x, y, t) \tag{6}$$

where ρ is the density of the fluid and p_a is the atmospheric pressure, which can be arbitrarily chosen to be zero.

2.1.1. Hydrodynamic force

The hydrodynamic force is obtained by integrating the pressure on the surface of the body:

$$\mathbf{F_H} = -\iint_{\mathbf{M}\in\Lambda_B} p(\mathbf{M})\mathbf{n}(\mathbf{M})\mathrm{d}s \tag{7}$$

where $\mathbf{n}(\mathbf{M})$ is the outgoing body normal and $p(\mathbf{M})$ is the pressure at a point $\mathbf{M}$.

The pressure at a point in the fluid is given by Bernoulli's theorem:

$$p = p_a - \rho\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}\|\nabla\phi\|^2 + gz\right) \tag{8}$$

2.1.2. Weak-scatterer hypothesis

It is assumed that ϕ, η, p and $\mathbf{V}$ can be split in two terms corresponding to the incident wave and a scattered wave:

$$\begin{aligned} \phi &= \phi_i + \phi_s \\ \eta &= \eta_i + \eta_s . \end{aligned} \tag{9}$$

The incident wave is the solution of the following boundary problem which corresponds to that of the propagation of waves without the body:

$$\begin{cases} \Delta\phi_i = 0 & \forall \mathbf{M} \in \Omega_i & \text{(a)} \\ \dfrac{\partial \phi_i}{\partial n} = 0 & \forall \mathbf{M} \in \Gamma_{SB} & \text{(b)} \\ \dfrac{\partial \eta_i}{\partial t} = \dfrac{\partial \phi_i}{\partial z} - \nabla\phi_i . \nabla\eta_i & \forall z = \eta_i(x, y, t) & \text{(c)} \\ \dfrac{\partial \phi_i}{\partial t} = -g\eta_i + \dfrac{1}{2}\|\nabla\phi_i\|^2 + \dfrac{p_a}{\rho} & \forall z = \eta_i(x, y, t) & \text{(d)} \end{cases} \tag{10}$$

where Ω_i is the inner fluid domain delimited by Γ_{LB}, Γ_{SB} and the surface $z = \eta_i(x, y, t)$.

The quantities associated with the incident wave field appear as forcing terms that are assumed to be known. They are consequently described with a dedicated nonlinear wave model. For regular waves, one may use a model based on the stream function theory [31,32]. For the irregular wave fields, the High-Order Spectral (HOS) method can be used [33].

The basis hypothesis of the weak-scatterer method is that the quantities related to the scattered flow (velocity potential ϕ_s, free surface elevation η_s, pressure p_s and velocity $\mathbf{V}_s$) are small compared to the incident ones:

$$\begin{aligned} \phi_s &\ll \phi_i \\ \eta_s &\ll \eta_i \end{aligned} \tag{11}$$

That hypothesis allows the free surface conditions (KFSBC and DFSBC) to be linearized on the position of the incident wave $z = \eta_i(x, y, t)$. Keeping only the first-order terms in η_s and ϕ_s, the free surface conditions can be written [16] on $z = \eta_i(x, y, t)$:

$$\frac{\partial \eta_s}{\partial t} = -\frac{\partial \phi_s}{\partial z} - \nabla\phi_s . \nabla\eta_i - \nabla\phi_i . \nabla\eta_s + \eta_s\left(\frac{\partial^2 \phi_i}{\partial z^2} + \frac{\partial \nabla\phi_i}{\partial z} . \nabla\eta_i\right) - \nu\eta_s \tag{12}$$

$$\frac{\partial \phi_s}{\partial t} = -g\eta_s - \nabla\phi_s . \nabla\phi_i - \eta_s\left(\frac{\partial^2 \phi_i}{\partial z \partial t} + \frac{\partial \nabla\phi_i}{\partial z} . \nabla\phi_i\right) - \nu\phi_s \tag{13}$$

The terms $-\nu\eta_s$ and $-\nu\phi_s$ are artificial damping terms [34] that take non-zero values in areas close to the domain edge. They are added to the free surface conditions in order to dissipate the waves reflected at the edge of the domain due to the boundary condition $\frac{\partial \phi_s}{\partial n} = 0$ on Γ_{LB}.

In the case of circular domains:

$$\begin{aligned} \nu(r) &= \alpha\omega\left(\tfrac{r - r_0}{\lambda}\right)^2 & r &\geq r_0 = R_e - \beta\lambda \\ \nu(r) &= 0 & r &< r_0 \end{aligned} \tag{14}$$

where r is the distance of the **M** point from the central axis of the domain, R_e is the radius of the domain. The parameters α and β are used to adjust the damping intensity and the width of the absorption range. Typical values are $\alpha = 0.7$ and $\beta \approx 1$ [16].

BVP for the scattered velocity potential

The boundary value problem (BVP) for the scattered velocity potential therefore finally reads:

$$\begin{cases} \Delta\phi_s = 0 & \forall \mathbf{M} \in \Omega \quad \text{(a)} \\ \dfrac{\partial\phi_s}{\partial n} = -\dfrac{\partial\phi_i}{\partial n} + \dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{n} & \forall \mathbf{M} \in \Gamma_B \quad \text{(b)} \\ \dfrac{\partial\phi_s}{\partial n} = 0 & \forall \mathbf{M} \in \Gamma_{SB} \cup \Gamma_{LB} \quad \text{(c)} \\ \dfrac{\partial\phi_s}{\partial t} = -g\eta_s - \nabla\phi_s.\nabla\phi_i & \\ \quad -\eta_s\left(\frac{\partial^2\phi_i}{\partial z\partial t} + \frac{\partial\nabla\phi_i}{\partial z}.\nabla\phi_i\right) - \nu\phi_s & \forall \mathbf{M} \in \Gamma_{FS} \quad \text{(d)} \\ \dfrac{\partial\eta_s}{\partial t} = -\dfrac{\partial\phi_s}{\partial z} - \nabla\phi_s.\nabla\eta_i - \nabla\phi_i.\nabla\eta_s & \\ \quad +\eta_s\left(\frac{\partial^2\phi_i}{\partial z^2} + \frac{\partial\nabla\phi_i}{\partial z}.\nabla\eta_i\right) - \nu\eta_s & \forall \mathbf{M} \in \Gamma_{FS} \quad \text{(e)} \end{cases} \tag{15}$$

BVP for scattered acceleration potential

Eq. (8) shows that the time derivative of the velocity potential (or acceleration potential) is required to compute the resulting pressure on the body. It can be shown that the acceleration potential is the solution of the boundary value problem:

$$\begin{cases} \Delta\dfrac{\partial\phi_s}{\partial t} = 0 & \forall \mathbf{M} \in \Omega \quad \text{(a)} \\ \dfrac{\partial^2\phi_s}{\partial n\partial t} = -\dfrac{\partial^2\phi_i}{\partial n\partial t} + \ddot{\mathbf{x}}_{\mathbf{b}}.\mathbf{n} + q & \forall \mathbf{M} \in \Gamma_B \quad \text{(b)} \\ \dfrac{\partial^2\phi_s}{\partial n\partial t} = 0 & \forall \mathbf{M} \in \Gamma_{LB} \cup \Gamma_{SB} \quad \text{(c)} \\ \dfrac{\partial\phi_s}{\partial t} = -g\eta_s - \nabla\phi_s.\nabla\phi_i & \\ \quad -\eta_s\left(\frac{\partial^2\phi_i}{\partial z\partial t} + \frac{\partial\nabla\phi_i}{\partial z}.\nabla\phi_i\right) - \nu\phi_s & \forall \mathbf{M} \in \Gamma_{FS} \quad \text{(d)} \\ \dfrac{\partial\eta_s}{\partial t} = -\dfrac{\partial\phi_s}{\partial z} - \nabla\phi_s.\nabla\eta_i - \nabla\phi_i.\nabla\eta_s & \\ \quad +\eta_s\left(\frac{\partial^2\phi_i}{\partial z^2} + \frac{\partial\nabla\phi_i}{\partial z}.\nabla\eta_i\right) - \nu\eta_s & \forall \mathbf{M} \in \Gamma_{FS} \quad \text{(e)} \end{cases} \tag{16}$$

where $\ddot{\mathbf{x}}_{\mathbf{b}}$ denotes the acceleration of the body at the point **M** and q is an advection term due to the motion of the body [35–37]:

$$\begin{aligned} q =& (\boldsymbol{\omega}_{\mathbf{b}}.\mathbf{s_1})\left(\frac{\partial\phi}{\partial s_2} - 2\dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{s_2}\right) - (\boldsymbol{\omega}_{\mathbf{b}}.\mathbf{s_2})\left(\frac{\partial\phi}{\partial s_1} - 2\dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{s_1}\right) \\ &+ \frac{\dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{s_1}}{R_1}\left(\frac{\partial\phi}{\partial s_1} - \dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{s_1}\right) + \frac{\dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{s_2}}{R_2}\left(\frac{\partial\phi}{\partial s_2} - \dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{s_2}\right) \\ &+ (\dot{\mathbf{x}}_{\mathbf{b}}.\mathbf{n})\left(\frac{\partial^2\phi}{\partial s_1^2} + \frac{\partial^2\phi}{\partial s_2^2} + \left(\frac{1}{R_1} + \frac{1}{R_2}\right)\frac{\partial\phi}{\partial n}\right) \end{aligned} \tag{17}$$

Here, $\mathbf{s_1}$ and $\mathbf{s_2}$ are the vectors of a frame tangent to the body surface and R_1 and R_2 are the radii of curvature in the $\mathbf{s_1}$ and $\mathbf{s_2}$ directions.

2.1.3. *Integral equations*

Green's second identity makes it possible to convert the volume problem into a surface problem. For any point **M** of the closed surface $\partial\Omega$ delimiting the fluid domain, we have:

$$0 = \iint_{\mathbf{P}\in\partial\Omega} \frac{\partial\phi_s(\mathbf{P})}{\partial n} G(\mathbf{P},\mathbf{M})\mathrm{d}s - \iint_{\mathbf{P}\in\partial\Omega} \left(\phi_s(\mathbf{P}) - \phi_s(\mathbf{M})\right) \frac{\partial G(\mathbf{P},\mathbf{M})}{\partial n}\mathrm{d}s \tag{18}$$

with:

$$G(\mathbf{P},\mathbf{M}) = \frac{1}{\|\mathbf{M}-\mathbf{P}\|} \tag{19}$$

$$\frac{\partial G(\mathbf{P},\mathbf{M})}{\partial n} = \frac{(\mathbf{M}-\mathbf{P}).\mathbf{n}}{\|\mathbf{M}-\mathbf{P}\|^3} \tag{20}$$

Likewise, for the acceleration potential:

$$0 = \iint_{\mathbf{P}\in\partial\Omega} \frac{\partial^2\phi_s(\mathbf{P})}{\partial n\partial t} G(\mathbf{P},\mathbf{M})\mathrm{d}s - \iint_{\mathbf{P}\in\partial\Omega} \left(\frac{\partial\phi_s(\mathbf{P})}{\partial t} - \frac{\partial\phi_s(\mathbf{M})}{\partial t}\right) \frac{\partial G(\mathbf{P},\mathbf{M})}{\partial n}\mathrm{d}s \tag{21}$$

2.2. *Implementation (WS-CN code)*

The boundaries $\partial\Omega$ of the domain are discretized into N_e triangular panels. The panels are denoted by $(\Sigma_k)_{1\le k\le N_e}$. The vertices are denoted by $(\mathbf{M}_i)_{1\le i\le N_p}$ where N_p is the total number of vertices of the mesh. Fig. 2 shows the workflow implemented in the WS-CN code where $u|_i^m$ corresponds to the value of the quantity u at node $\mathbf{M}_i$ and time t_m.

2.2.1. *No-flow conditions (step 1)*

Eqs. (15b) and (15c) are used to compute the normal derivative of the velocity potential on the outer boundaries of the domain as well as on the body surface.

The normal vectors at the grid nodes are required to compute Eq. (15b). They are obtained by calculating the weighted average of the normal vectors of the panels to which each grid node belongs. The weight of the normal vector corresponding to panel k is taken equal to the angle of panel k at the vertex corresponding to the grid node.

To reduce the number of unknowns, it is possible not to mesh the surface corresponding to the seabed. To do this, the free surface Γ_{FS}, the outer boundary Γ_{LB} and the body surface Γ_B surface are mirrored with respect to the seabed (Fig. 3). The no-flow condition on the Γ_{SB} surface is ensured for this new problem because of symmetry. Moreover, for the same reason, the values of the potentials and normal derivatives of potentials at a mesh node are respectively equal and opposite to that of the symmetrical node, so that adding these fictitious surfaces does not increase the number of unknowns.

2.2.2. *1st boundary value problem (step 2)*

Using the discretization described in the previous section, the integral equation Eq. (18) is written at each point $(\mathbf{M}_i)_{1\le i\le N_p}$ of the mesh:

$$\begin{aligned} 0 =& \sum_{1\le k\le N_e} \iint_{\mathbf{P}\in\Sigma_k} \frac{\partial\phi_s(\mathbf{P})}{\partial n} G(\mathbf{P},\mathbf{M}_i)\mathrm{d}s - \sum_{1\le k\le N_e} \iint_{\mathbf{P}\in\Sigma_k} \left(\phi_s(\mathbf{P}) - \phi_s(\mathbf{M}_i)\right) \\ &\times \frac{\partial G(\mathbf{P},\mathbf{M}_i)}{\partial n}\mathrm{d}s \end{aligned} \tag{22}$$

Let $\mathbf{P_1^k}$, $\mathbf{P_2^k}$ and $\mathbf{P_3^k}$ be the three vertices of a mesh panel Σ_k. By adopting a linear parametrization, the position of a point **P** on the facet can be described by two variables $(u,v) \in [0,1]^2$ in the basis $(\mathbf{P_1^kP_2^k}, \mathbf{P_2^kP_3^k})$ (Fig. 4):

$$\mathbf{P} = \mathbf{P_1^k} + u.(\mathbf{P_2^k} - \mathbf{P_1^k}) + v.(\mathbf{P_3^k} - \mathbf{P_1^k}) \tag{23}$$

In the WS-CN code, an iso-parametric description is used for the spatial distribution of quantities (potential and normal derivative of the potential) and the coordinates over the surface of a panel:

$$\phi(\mathbf{P}) = \phi(\mathbf{P_1^k}) + u.(\phi(\mathbf{P_2^k}) - \phi(\mathbf{P_1^k})) + v.(\phi(\mathbf{P_3^k}) - \phi(\mathbf{P_1^k})) \tag{24}$$

$$\frac{\partial\phi(\mathbf{P})}{\partial n} = \frac{\partial\phi(\mathbf{P_1^k})}{\partial n} + u.\left(\frac{\partial\phi(\mathbf{P_2^k})}{\partial n} - \frac{\partial\phi(\mathbf{P_1^k})}{\partial n}\right) + v.\left(\frac{\partial\phi(\mathbf{P_3^k})}{\partial n} - \frac{\partial\phi(\mathbf{P_1^k})}{\partial n}\right) \tag{25}$$

Owing to the linearity of integrals, each term of Eq. (22) can be written as a linear combination of the values of the potentials or normal derivatives of the potential at the vertices of the panels. By introducing Eqs. (24) and (25) in the integrals, one can show:

$$\iint_{\mathbf{P}\in\Sigma_k} \frac{\partial\phi_s(\mathbf{P})}{\partial n} G(\mathbf{P},\mathbf{M}_i)\mathrm{d}s = \sum_{j/\mathbf{M}_j\in\Sigma_k} \alpha_{i,j,k} \frac{\partial\phi_s(\mathbf{M_j})}{\partial n} \tag{26}$$

$$\iint_{\mathbf{P}\in\Sigma_k} \phi_s(\mathbf{P}) \frac{\partial G(\mathbf{P},\mathbf{M}_i)}{\partial n}\mathrm{d}s = \sum_{j/\mathbf{M}_j\in\Sigma_k} \beta_{i,j,k}\phi_s(\mathbf{M_j}) \tag{27}$$

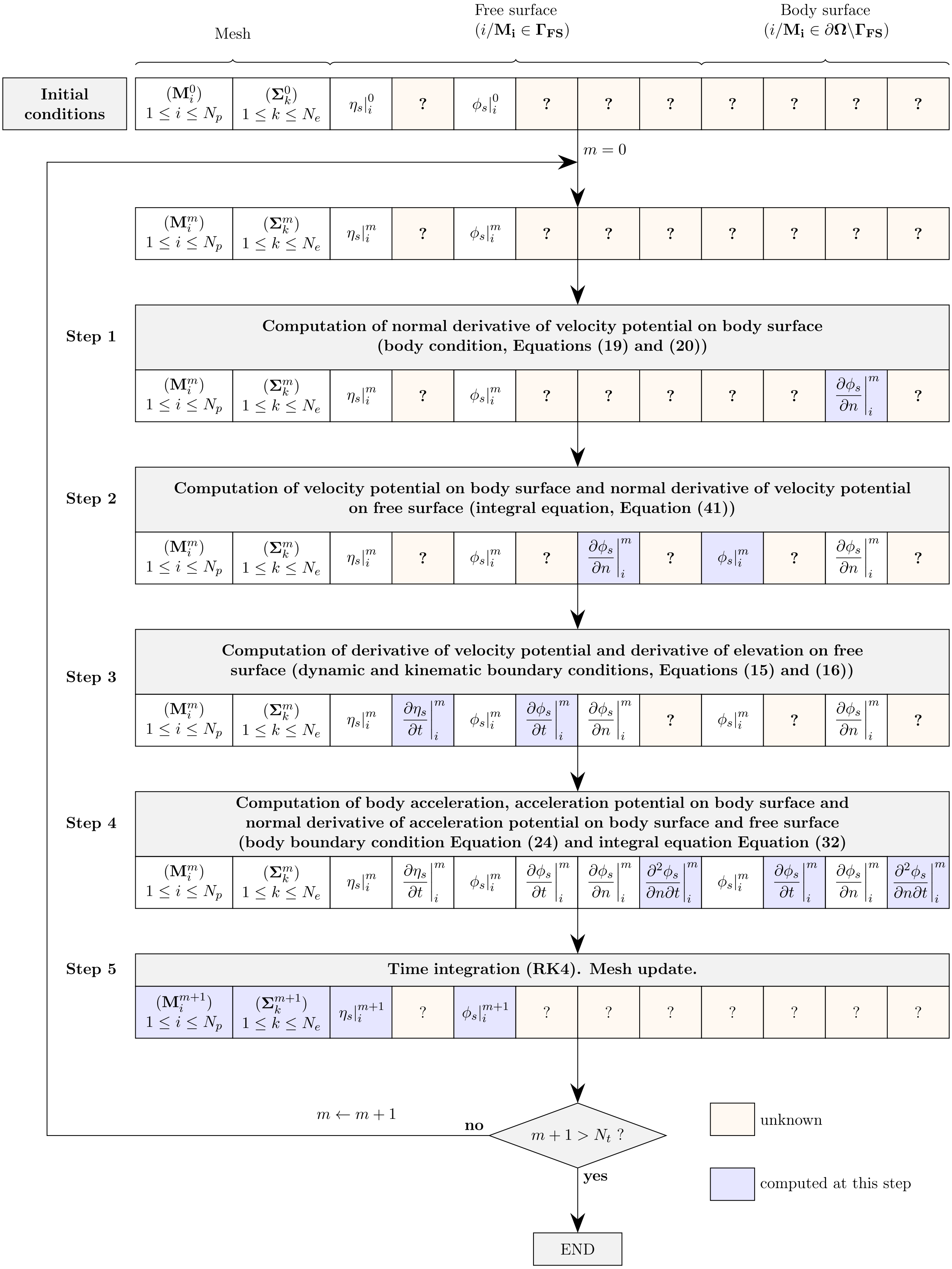

Fig. 2. Workflow.

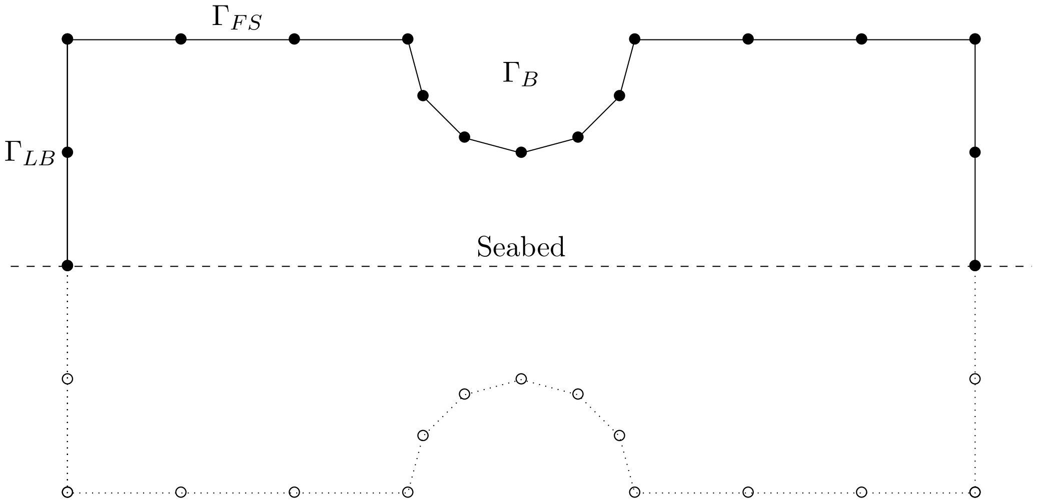

Fig. 3. Symmetrization of the discretization of the problem with respect to the seabed.

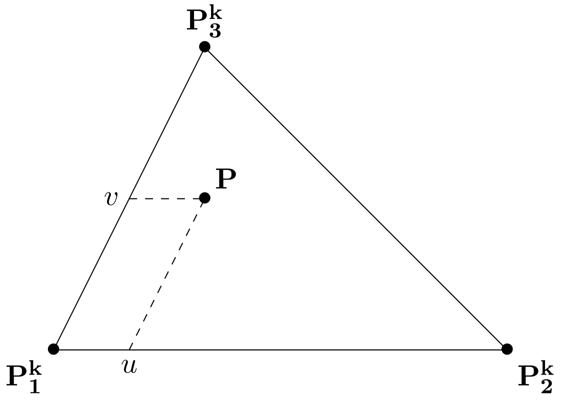

Fig. 4. Linear parametrization of a panel.

$$\iint_{\mathbf{P}\in\Sigma_k} \phi_s(\mathbf{M_i})\frac{\partial G(\mathbf{P},\mathbf{M}_i)}{\partial n}\mathrm{d}s = \gamma_{i,k}\phi_s(\mathbf{M_i}) \tag{28}$$

The expressions for $\alpha_{i,j,k}$, $\beta_{i,j,k}$ and $\gamma_{i,k}$ can be found in [38].

Introducing Eqs. (25), (26) and (27) in Eq. (21) leads to the linear system:

$$\sum_{1\leq j\leq N_p} CS(i,j)\frac{\partial\phi_s(\mathbf{M_j})}{\partial n} - \sum_{1\leq j\leq N_p} CD(i,j)\phi_s(\mathbf{M_j}) = 0 \tag{29}$$

where $CS(i,j)$ and $CD(i,j)$ are the coefficients of influence of a source point $\mathbf{M}_j$ on a control point $\mathbf{M}_i$ where $1 \leq i \leq N_p$. These coefficients are simple linear combinations of $\alpha_{i,j,k}$, $\beta_{i,j,k}$ and $\gamma_{i,k}$ coefficients.

By separating the known terms (ϕ_s on the free surface and $\frac{\partial\phi_s}{\partial n}$ on the body) and the unknown terms at this stage ($\frac{\partial\phi_s}{\partial n}$ on the free surface and ϕ_s on the body), we obtain a system of the form:

$$A\mathbf{X} = \mathbf{B} \tag{30}$$

with:

$$A\mathbf{X}(i) = \sum_{j/\mathbf{M}_j\in\Gamma_{FS}} CS(i,j)\frac{\partial\phi_s(\mathbf{M_j})}{\partial n} - \sum_{j/\mathbf{M}_j\in\Gamma_{FS}\cup\Gamma_{LB}\cup\Gamma_{SB}} CD(i,j)\phi_s(\mathbf{M_j}) \tag{31}$$

$$\mathbf{B}(i) = -\sum_{j/\mathbf{M}_j\in\Gamma_{FS}\cup\Gamma_{LB}\cup\Gamma_{SB}} CS(i,j)\frac{\partial\phi_s(\mathbf{M_j})}{\partial n} + \sum_{j/\mathbf{M}_j\in\Gamma_{FS}} CD(i,j)\phi_s(\mathbf{M_j}) \tag{32}$$

The solution to this system can be obtained by LU decomposition or by using a GMRES (Generalized Minimal RESidual method) iterative solver. In the WS-CN code, the matrix A is assembled in both cases (even if it is not needed in case of the GMRES solver).

The matrix A depends on the mesh geometry and must therefore be determined at each time step. Assembling and solving System (30) account for the bulk of the computation time.

2.2.3. Dynamic and kinematic boundary conditions (step 3)

Eq. (13) is used to obtain the time derivative of the potential at each nodes of the free surface. This equation involves the gradient of the scattered potential, where only the normal component is known at this stage (Step 2).

$$\boldsymbol{\nabla}\phi_s = \boldsymbol{\nabla}_{\mathbf{s}}\phi_s + \frac{\partial\phi_s}{\partial n}\mathbf{n} \tag{33}$$

The surface gradients $\boldsymbol{\nabla}_s\phi_s$ and $\boldsymbol{\nabla}_s\eta_s$, i.e. the projection of the gradients in the horizontal plane, are obtained by approximating locally the spatial evolution of ϕ_s and η_s using B-splines. These B-splines are also used to compute the second derivatives involved in the advection term, Eq. (17) (Step 4).

2.2.4. 2nd boundary value problem (step 4)

Eq. (16b) is used to determine the value of $\frac{\partial^2\phi_s}{\partial n\partial t}$ at each point of the body. The resulting system is similar to Eq. (30), namely the 1st Boundary Value Problem.

$$A\mathbf{X} = \mathbf{B} \tag{34}$$

with:

$$A\mathbf{X}(i) = \sum_{j/\mathbf{M}_j\in\Gamma_{FS}} CS(i,j)\frac{\partial^2\phi_s(\mathbf{M_j})}{\partial n\partial t} - \sum_{j/\mathbf{M}_j\in\Gamma_{FS}\cup\Gamma_{LB}\cup\Gamma_{SB}} CD(i,j)\frac{\partial\phi_s(\mathbf{M_j})}{\partial t} \tag{35}$$

$$\mathbf{B}(i) = -\sum_{j/\mathbf{M}_j\in\Gamma_{FS}\cup\Gamma_{LB}\cup\Gamma_{SB}} CS(i,j)\frac{\partial^2\phi_s(\mathbf{M_j})}{\partial n\partial t} + \sum_{j/\mathbf{M}_j\in\Gamma_{FS}} CD(i,j)\frac{\partial\phi_s(\mathbf{M_j})}{\partial t} \tag{36}$$

where the terms $CS(i,j)$ and $CD(i,j)$ are the same values as for Eq. (30) and therefore do not need to be recomputed.

2.2.5. Time integration (step 5)

Finally, η_s and ϕ_s at the free surface are updated by time integration with a fourth-order Runge–Kutta scheme.

The points $(\mathbf{M}_i)$ are updated according to the new position of the body and of the incident wave at the next time step. If, after updating the mesh, some panels no longer comply with certain geometric constraints, a new mesh is generated [39].

The state of the system is thus fully determined by the knowledge at time t of the state vector U:

$$U = (\phi(\Gamma_{FS}), \eta(\Gamma_{FS}), \mathbf{x_b}, \dot{\mathbf{x}}_\mathbf{b}) \tag{37}$$

By using boundary conditions and assembling and solving the boundary problem, the state of the system at the next time step can be determined by integration (Fig. 2). The function used to estimate the value of the state vector at the next time step using this method is called the “fine propagator”, denoted by F:

$$U(t+\delta t) = F(t, t+\delta t, U(t)) \tag{38}$$

2.3. Computational cost and acceleration

Fig. 5 shows the computational time associated with each step of the fine propagator computation of a typical case study (simulation with two surface-piercing cylinders and a mesh of 8300 panels and 4550 nodes) [30]. One can see that steps associated with the computation of the influence coefficients (47.1%) and the build (8.4%) and solving of the linear systems (20%) account for most of the computational time (75.5%). Nevertheless, one can see that the time spent for remeshing is significant as it accounts for approximately 20% of the total computation time.

To reduce the computation time, Wuillaume et al. [30] implemented a parallelization of the computation of the influence coefficients required to assemble the matrices of the two boundary problems in the WS-CN code (step 2). They achieved a reduction of 70% in the computation time of step 2, thus reducing the total simulation time by 35%.

The acceleration methods presented in the introduction (p-FFT, FMM, $\mathcal{H}$-matrix) also make it possible to reduce the computational time of Steps 2 and 4 (computation of the influence coefficients, then building and solving of the linear systems), which represents approximately 75% of the total computational time. However, they have no influence on the other steps which represent non-negligible amounts of time. Parallelization of those other steps can be challenging. In particular, the remeshings of the free surface and the body are performed using an advancing front method. Techniques exist for parallelizing the meshing of surfaces [40,41], but using these methods would involve major modifications to the code.

Conversely, parallelization in time speeds up in principle the simulation as a whole, without the need to implement acceleration techniques depending on the step of the computation. This method is described in the next section.

3. Parareal method

The objective of this section is to present the time parallelization method and its implementation in the case of WS-CN code.

3.1. General description of the method

Parallelization in time consists in dividing the time interval over which the simulation is to be carried out and running the simulation in parallel on each of the sub-intervals. The concept is that each processor uses the fine propagator on one specific sub-interval. However, for a processor to be able to use the propagator, it needs to be provided with initial conditions that only the simulations on the previous sub-intervals, once completed, can provide. In the Parareal method, the concept is to use a coarse but fast propagator in order to obtain an approximation of the solution and thus provide estimates of the initial conditions for each sub-interval to the fine propagator [23].

An advantage of the Parareal method is that it can be implemented in existing codes with relatively few modifications. Indeed, in this method, both the coarse and the fine propagators can be considered as “black boxes” that just have to be capable of calculating the temporal evolution of a system given its initial state.

The method is illustrated in Fig. 6. Let N_p be the number of temporal subintervals, which also corresponds to the number of processors involved in the parallelization procedure. It is equal to the number of processors used in the simulation. The times delimiting the different subintervals are denoted by t_0, t_1, t_2, $\dots t_{N_p}$.

As mentioned previously, two propagators are required:

- $C(t_1, t_2, u_1)$ is the coarse propagator which provides a quick estimate of the response of the system at time t_2 given the starting time t_1 and the initial state u_1,
- $F(t_1, t_2, u_1)$ is the fine propagator which provides the system response at time t_2 given the starting time t_1 and the initial state u_1.

The first step of the Parareal algorithm consists in calculating first approximations of the solution at times t_k. They are denoted U_k^0. They are obtained using the coarse propagator sequentially.

The second step consists in using the fine propagator in parallel, starting from the initial conditions U_k^0 provided by the coarse propagator.

The rest of the algorithm consists in iteratively approximating the solution (Fig. 6). The initial conditions for each subinterval at time t_{k+1} and at iteration $n+1$ are given by the relationship:

$$U_{k+1}^{n+1} = C(t_k, t_{k+1}, U_k^{n+1}) + F(t_k, t_{k+1}, U_k^n) - C(t_k, t_{k+1}, U_k^n) \tag{39}$$

At iteration $n+1$, the $F(t_k, t_{k+1}, U_k^n)$ term uses states already computed at iteration n using the coarse propagator, so that they can be computed in parallel (one computation per sub-interval).

The $C(t_k, t_{k+1}, U_k^{n+1})$ term involves the result of the simulation over the previous interval. Therefore it must be obtained sequentially. This feature highlights the need for the coarse propagator to be as fast as possible. To achieve this requirement, one can use:

- a larger time step, as done in the original method [23],
- a coarser space discretization,
- a lower order time integration scheme,
- an other model [42,43].

In this work, we consider this last strategy.

The Parareal method being an iterative method, it requires a convergence test to stop. In order to quantify the convergence at iteration n, let us define the following quantity:

$$\alpha_n = \max\left(\alpha_n^k\right) = \max\left(\frac{\left\|U_k^n - U_k^{n-1}\right\|}{\left\|U_k^n\right\|}\right) \tag{40}$$

In this study, it is considered that convergence is achieved as soon as:

$$\alpha_n < \alpha_{min} \tag{41}$$

where $\alpha_{min} = 10^{-2}$.

As can be seen in Fig. 6, $\forall k \geq n, U_k^n = U_n^n$. Therefore, at each new iteration $k+1$ of the Parareal algorithm, the solution with the fine propagator is known up to time t_k. Despite this feature can save some computational time, it was not exploited in this work.

3.2. Implementation of the parareal method in the WS-CN code

In the present study:

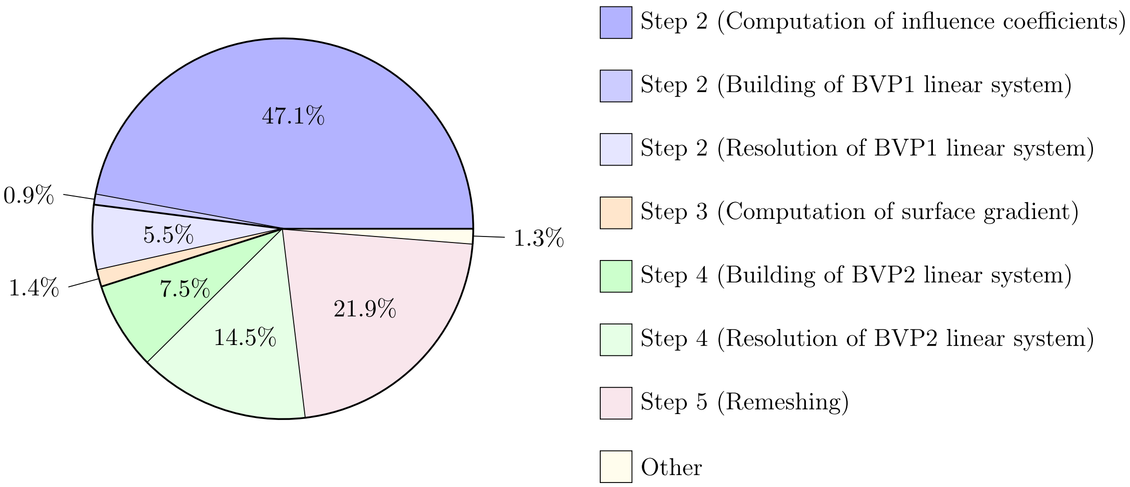

Fig. 5. Time distribution.

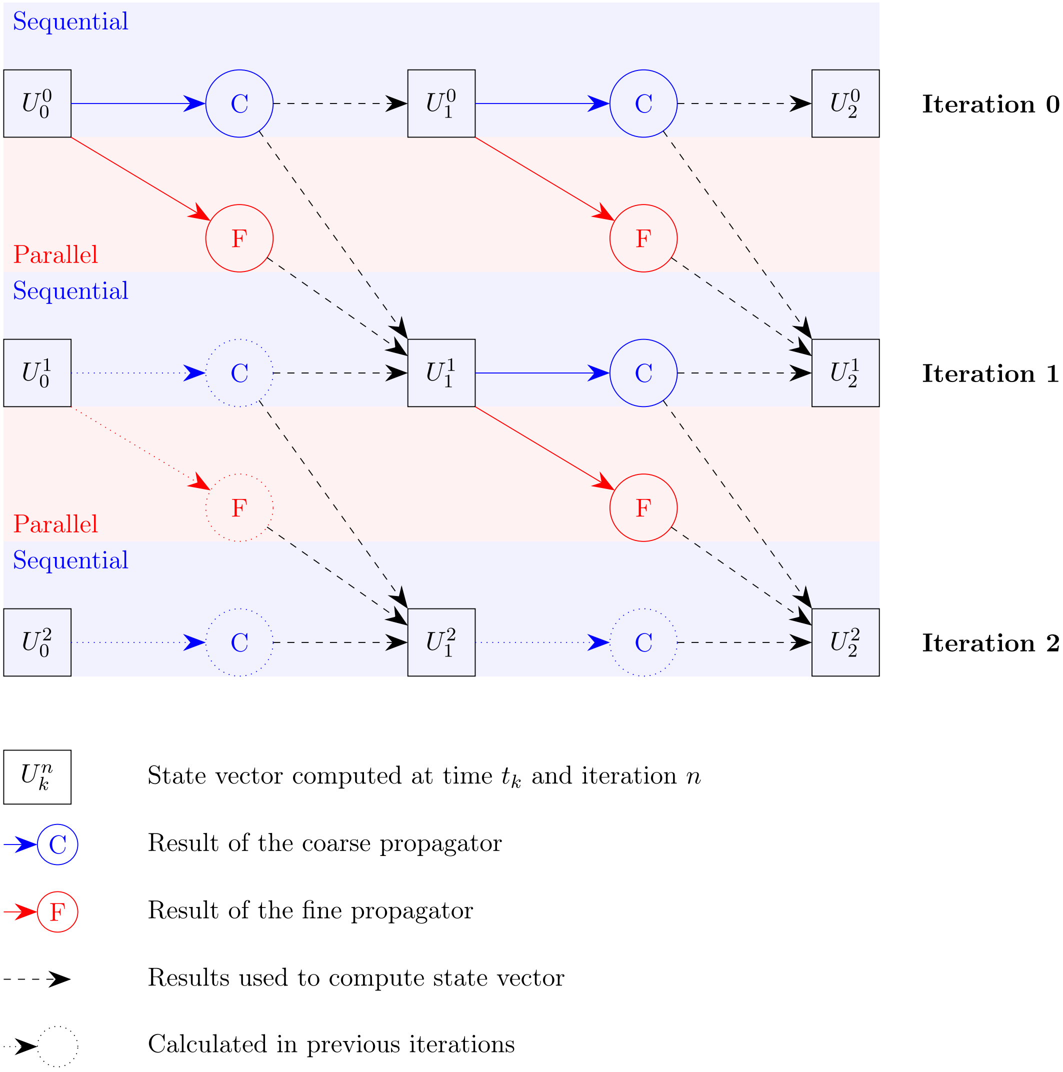

Fig. 6. Parareal algorithm.

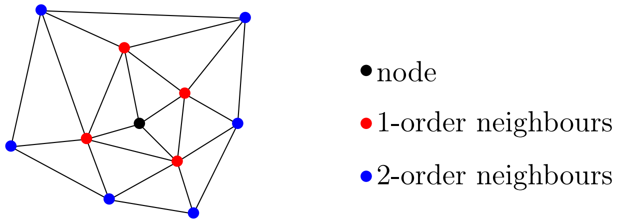

Fig. 7. 1 and 2-order neighbors.

- the fine propagator uses the Weak-Scatterer method, for which the linear system of the 1st Boundary Value Problem must be assembled and solved at each iteration.
- the coarse propagator uses a fully linearized approach where the domain is discretized on the mean position of the free surface and on the equilibrium position of the body. The mesh is then constant throughout the simulation. The matrix associated with Eq. (30), which only depends on the mesh position, can therefore be computed and inverted once at the start of the simulation, saving considerable time.

A significant difficulty in applying the Parareal method to the WS-CN code is that the two propagators do not *a priori* share the same meshes:

- the coarse propagator uses a mesh M_c which has been computed from the mean position at the free surface and at the equilibrium position of the body. This mesh does not change over time.
- the fine propagator uses a mesh M_f which follows the incident wave elevation and the instantaneous position of the body. It therefore changes over time.

In order to be able to relate the results of the two propagators, it is necessary for them to return the values of the velocity and elevation potentials at the same points. It is therefore decided to interpolate the state vector returned by the fine propagator to the mesh used by the coarse propagator. This interpolation is carried out using a B-spline approximation [44], as done in the computation of the surface gradients in Eq. (33).

Let (x, y) be the coordinates of a point on the free surface belonging to the mesh M_c. Let (x_0, y_0) be the coordinates of the point in the mesh M_f closest to this point and (x_i, y_i) where $1 \leq i \leq N_v$ are the coordinates of the N_v 1-order and 2-order neighbors in the mesh M_f (Fig. 7). Let us define a function σ such that $\sigma(x, y)$ gives a good approximation of the value of a quantity F (free surface elevation or velocity potential) at the point (x, y) of the mesh M_c from the values F_i of this quantity at the points (x_i, y_i) of the mesh M_f. For example, one may use order 3 pseudo-polynomial splines which are of the form:

$$\sigma(x, y) = \sum_{i=0}^{N_v} \alpha_i \left((x - x_i)^2 + (y - y_i)^2\right)^3 + P(x, y)$$

with

$$P(x, y) = \alpha_{n+1} + \alpha_{n+2} x + \alpha_{n+3} y + \alpha_{n+4} x^2 + \alpha_{n+5} xy + \alpha_{n+6} y^2 \tag{42}$$

where the values of α_i are determined using the following $N_v + 7$ constraints:

$$\sigma(x_i, y_i) = F_i, \forall i \in [\![0, N_v]\!] \tag{43}$$

$$\begin{aligned} &\sum_{i=0}^{N_v} \alpha_i = 0 && \sum_{i=0}^{N_v} \alpha_i x_i^2 = 0 \\ &\sum_{i=0}^{N_v} \alpha_i x_i = 0 && \sum_{i=0}^{N_v} \alpha_i y_i^2 = 0 \\ &\sum_{i=0}^{N_v} \alpha_i y_i = 0 && \sum_{i=0}^{N_v} \alpha_i x_i y_i = 0 \end{aligned} \tag{44}$$

Table 1
Error E_N as function of number of panels.

Number of panels N	1.9×10^2	2.4×10^2	4.0×10^2
Error E_N	2.2×10^{-2}	1.0×10^{-2}	0

4. Results

4.1. Case study

The case study corresponds to a vertical fixed cylinder of height $h = 0.9\,\text{m}$, radius $R = 0.2\,\text{m}$ and mass $m = 65\,\text{kg}$ (Fig. 8). The incident wave is a regular wave with period $T = 1\,\text{s}$, wavelength $\lambda = 1.6\,\text{m}$, amplitude A and steepness $\varepsilon = \frac{2A}{\lambda}$. The water depth is $H = 1\,\text{m}$ and the lateral boundaries of the domain are located at a distance $3.0\,\text{m}$ $(15R)$ from the center. For the various simulations, a time step $\delta t = 0.01\,\text{s}$ was used.

The aim is to simulate the interactions between the incident wave and the cylinder over time, in particular the scattered wave generated by the interactions of the cylinder with the incident wave and the forces applying to the cylinder. Simulation results for an example simulation with the WS-CN code without time parallelization are shown in Fig. 9. The simulation was carried out with 3 different meshes with varying discretization in order to assess mesh convergence. In Fig. 9, differences in the wave elevation and vertical force are hardly distinguishable between the three different meshes. Therefore, to quantify the accuracy, we define E_N a measure of the error on the vertical force :

$$E_N = \frac{\displaystyle\int_0^{N_t \delta t} \left| F_N(t) - F_{4000}(t) \right| \mathrm{d}t}{\displaystyle\int_0^{N_t \delta t} \left| F_{4000}(t) \right| \mathrm{d}t} \tag{45}$$

where $F_N(t)$ corresponds to the vertical force calculated with a mesh with N panels (thus $F_{4000}(t)$ is the solution with the finest mesh).

Table 1 shows the error E_N as function of the number of panels. One can see that the error is of order of 10^{-2}. In what follows, the mesh with $N = 1,9 \times 10^2$ panels was used.

Fig. 9 shows that after a short transient period, the free surface elevation and the force on the cylinder is quasi-periodic. The deviations from periodicity are due to wave reflections on the lateral boundaries of the simulation, which are not fully eliminated by the numerical absorbing beach.

For sake of comparison, Fig. 9 also includes simulation results obtained with the linear model (coarse propagator). Significant differences with the results obtained with the nonlinear model can be observed. This shows that nonlinear effects are non-negligible in this example, which justifies the use of a nonlinear method.

4.2. Convergence of the parareal method

Figs. 10 and 11 show the results of a simulation carried out using the WS-CN code with time parallelization for a wave of amplitude $A = 0.014\,\text{m}$. In each figure, the top graph shows the results obtained after the first iteration of the parareal method. The following curves correspond to the results after the second and the third iterations respectively. The vertical lines show the time intervals allocated to the different processors (20 processors were used).

On the top graph, one can observe that the simulation result obtained with the fine propagator after the first iteration are both very similar to that obtained with a simulation using only the coarse propagator (fully linear model) and very different from the reference simulation (original version of the WS-CN code). However, note the discontinuities at the times corresponding to the bounds of the intervals allocated to each processor (e.g at time $t_6 = 1.8\,\text{s}$ or $t_{18} = 5.4\,\text{s}$).

After the second and the third iterations, one can see that these discontinuities have been greatly reduced. Moreover, the scattered

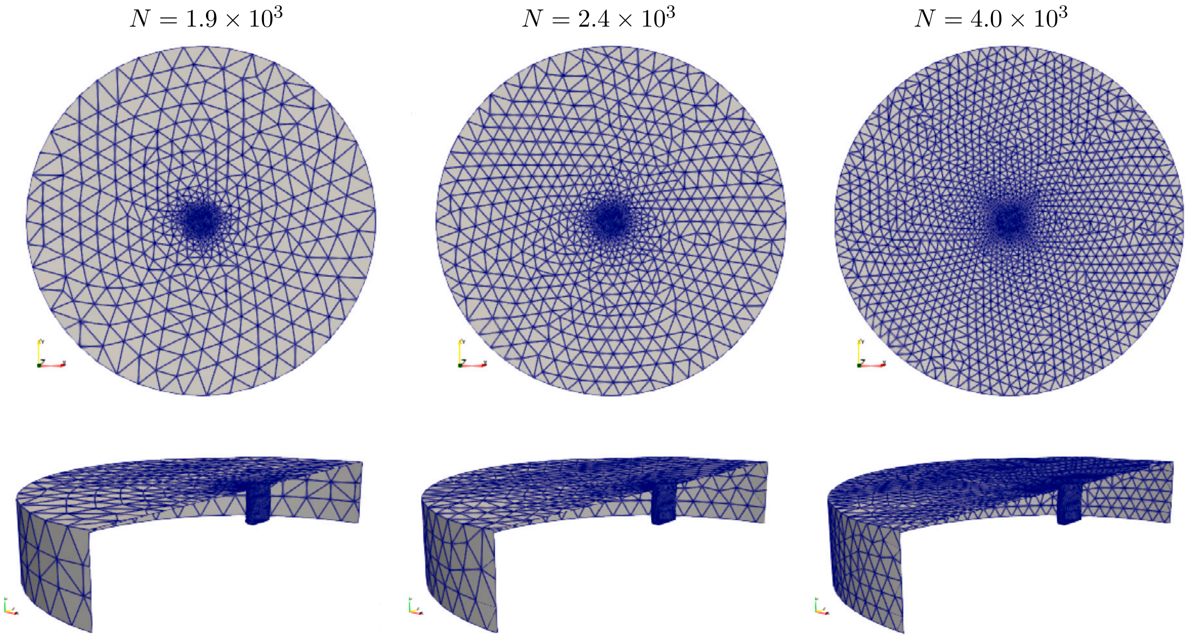

Fig. 8. Pictures of the meshes of the case study.

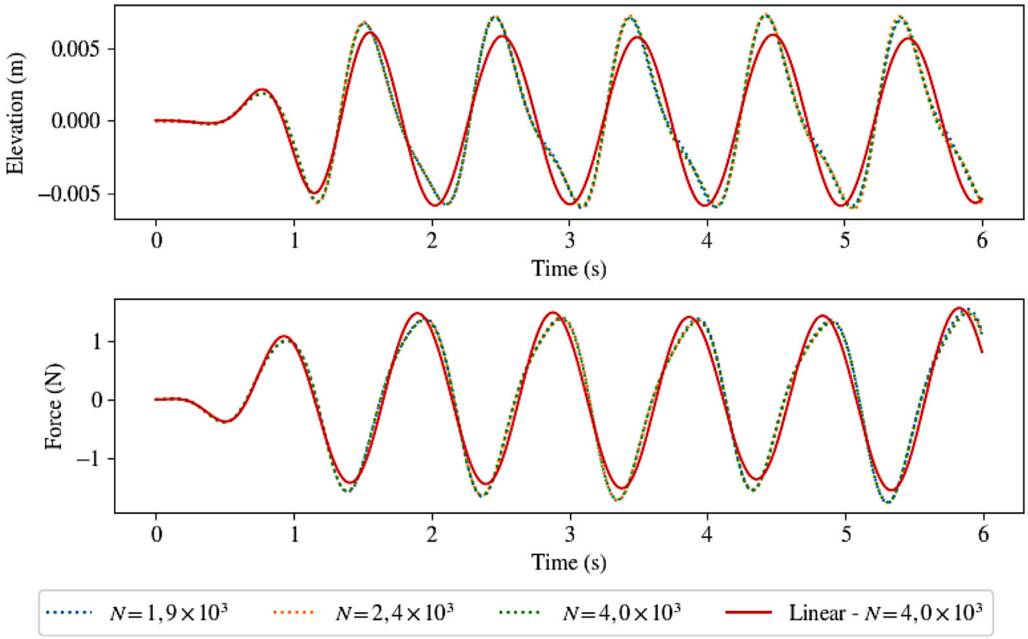

Fig. 9. Scattered wave elevations (top) as function of time for a point of the free surface located 0.2 m downstream of the cylinder and vertical forces (bottom) applying to the cylinder. They were obtained using the WS-CN without time parallelization ($A = 0.014\,\mathrm{m}$, $\varepsilon = 0.017$) with three different meshes and the linear model with a mesh with $N = 4,0{\times}10^2$ panels.

wave elevation and the vertical force applying to the body are very close to the reference results. The method therefore seems to quickly achieve convergence.

To measure convergence, let us define a measure $\mathcal{E}_n$ of the consistency between the Parareal method, at iteration n, and the original version of WS-CN code:

$$\mathcal{E}_n = \frac{\displaystyle\int_0^{N_t\delta t} \left|F_n(t) - F_{WS}(t)\right| \mathrm{d}t}{\displaystyle\int_0^{N_t\delta t} \left|F_{WS}(t)\right| \mathrm{d}t} \tag{46}$$

where $F_n(t)$ corresponds to the vertical force obtained at time t after n Parareal iterations and $F_{WS}(t)$ corresponds to the vertical force obtained at time t with the original version of the WS-CN code.

The top panel in Fig. 12 shows $\mathcal{E}_n$ as function of the number of iterations for various wave steepness. For the sake of comparison, the errors obtained with the linear model (coarse propagator) are also shown in the figure. One can see that it takes only 2 iterations for the error on the vertical forces to be an order of magnitude less than that obtained by a fully linear simulation, irrespective of the wave steepness. Moreover, results show that increasing the number of iterations do not reduce further the error. This can be explained by the fact that the remeshing steps in the WS-CN code without time parallelization

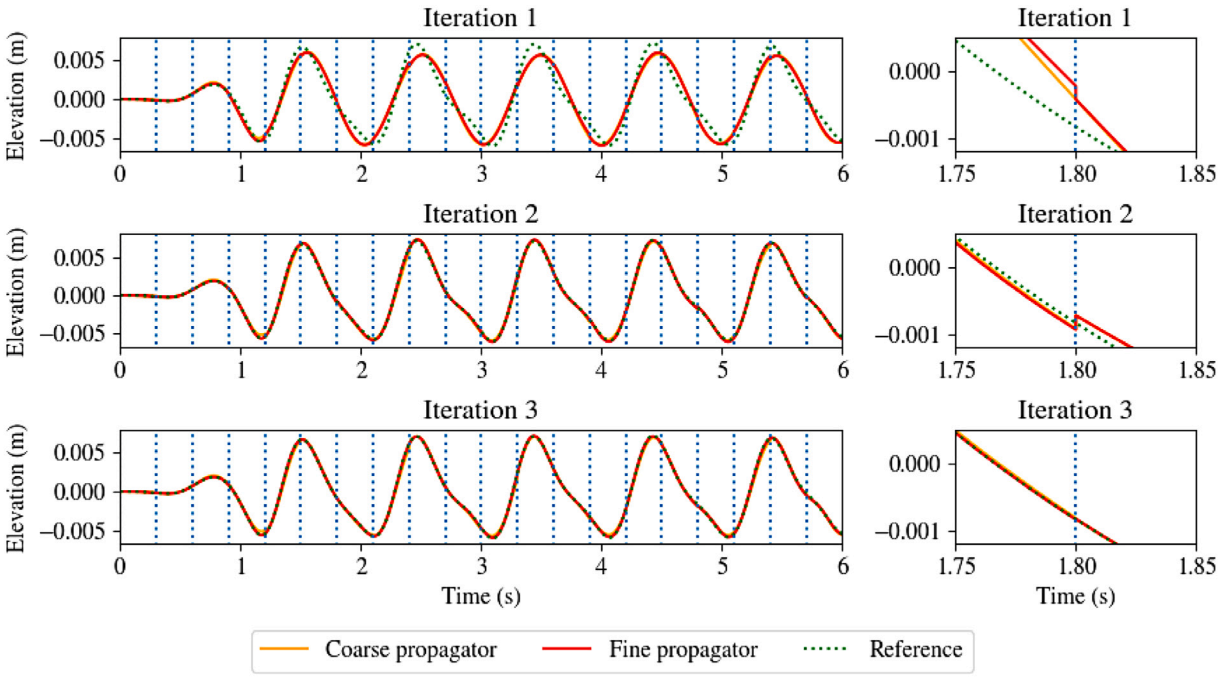

Fig. 10. Scattered wave elevation as function of time at a point located 0.2 m downstream of the cylinder (A = 0.014 m, ε = 0.017, T = 1 s, N_p = 20, N_t = 600). The vertical dotted lines separate the sub-intervals which are allocated to the different processors. On the right, we zoomed in on the time t_6 = 1.8 s corresponding to a transition between two processors.

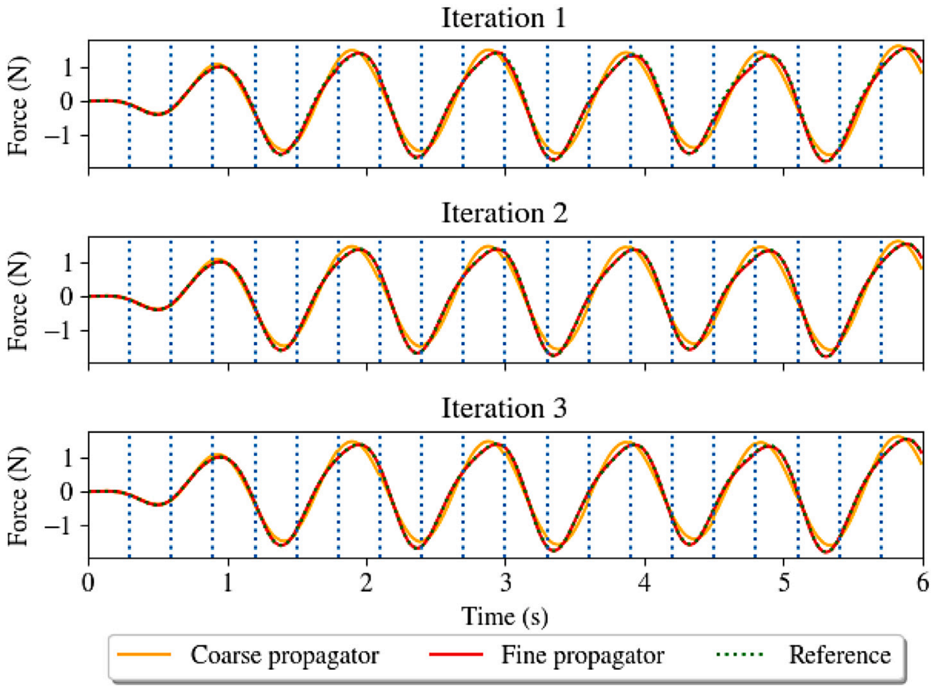

Fig. 11. Vertical forces experienced by the cylinder (A = 0.014 m, ε = 0.017, T = 1 s, N_p = 20, N_t = 600). The vertical dotted lines separate the sub-intervals which are allocated to the different processors.

and with time parallelization do not occur at the same time. Thus, the remaining error is a geometrical error, which is on the order of 10^{-2} according to Table 1. Thus, the fact that the error plateaus is not related to a convergence issue in the parareal algorithm. This is supported by the bottom panel in Fig. 12 which shows that the convergence criterion α_n (Eq. (40)) keeps decreasing with increasing number of iterations. Despite it was not tested by the authors, it is believed that increasing the mesh resolution could address this challenge.

Obviously, the speed-up achieved with the Parareal method depends on the number of iterations, which depends on the criterion $\alpha_n < \alpha_{min}$. According to Fig. 12, $\alpha_{min} = 10^{-1}$ appears to be enough to achieve convergence for the vertical force. Using this criterion allows significant speed-ups to be achieved for wave amplitudes of less than 0.02 m, as can be seen in Fig. 13 which shows results for simulations of $N_t = 600$ time steps carried out with 20 processors in parallel. For example, for an incident wave amplitude A = 0.008 m with wave period T =

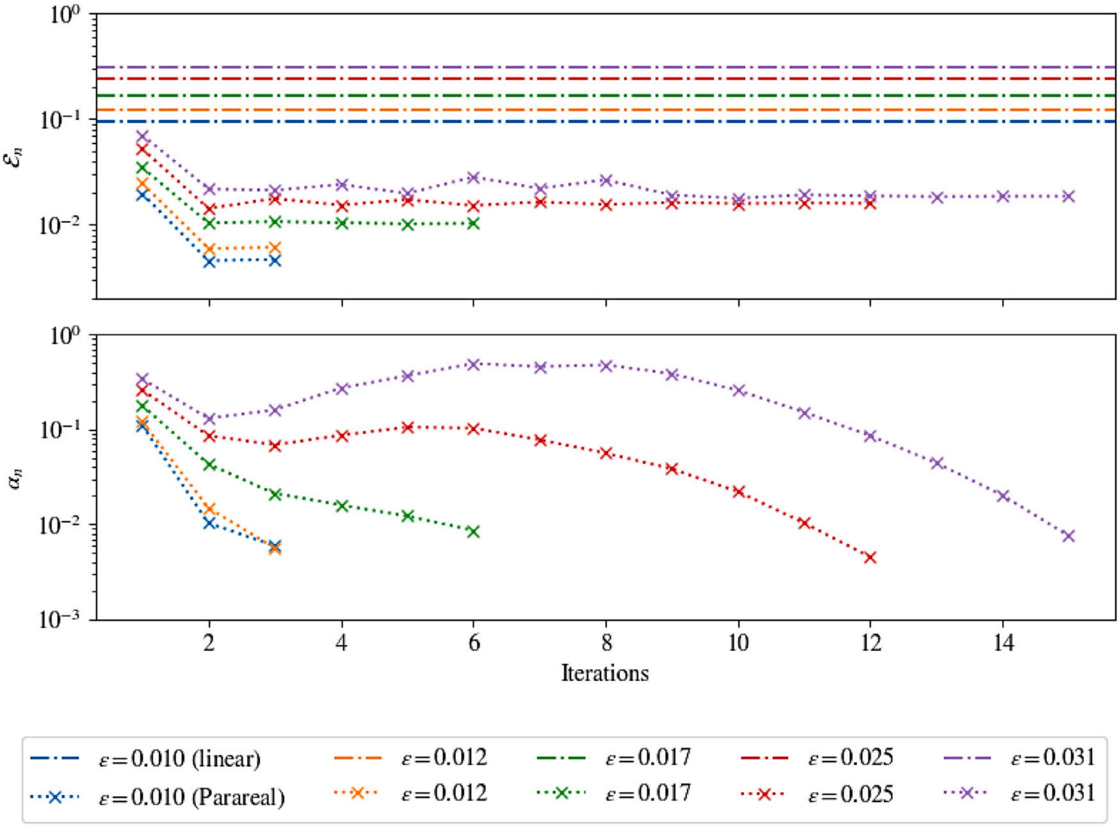

Fig. 12. Evolution of $\mathcal{E}_n$ and α_n (bottom) as function of number of iterations ($T = 1$ s, $N_p = 20$, $N_t = 600$).

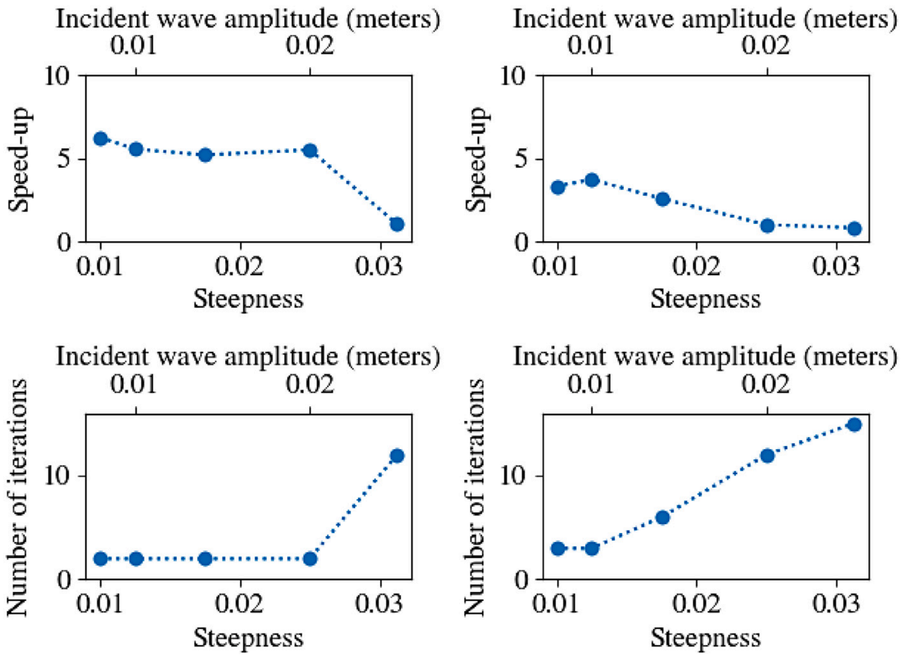

Fig. 13. Speed-up factor (top) and number of iterations required for the algorithm to converge (bottom) as a function of incident wave amplitude ($T = 1$ s, $N_p = 20$, $N_t = 600$) for a tolerance $\alpha_{min} = 10^{-1}$ (left) and $\alpha_{min} = 10^{-2}$ (right).

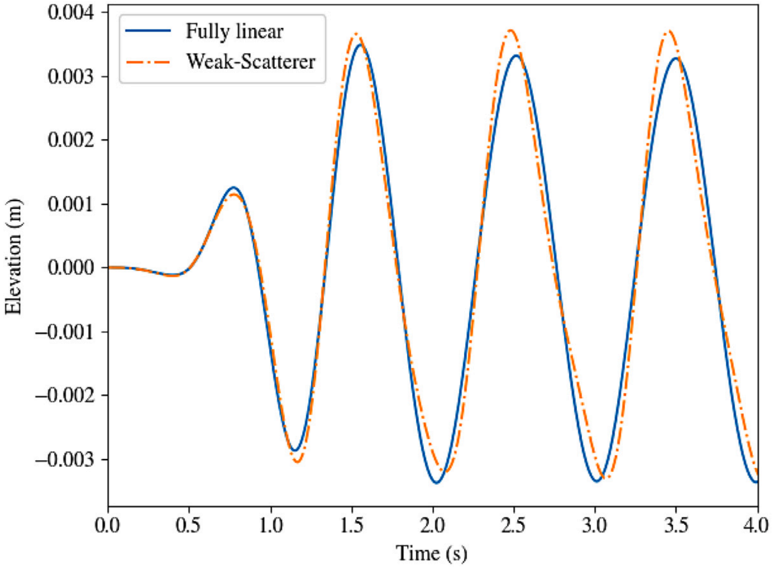

Fig. 14. Scattered elevation over time of a point on the free surface, located 0.2 m downstream of the cylinder obtained without time paralleling ($A = 0.008\,\mathrm{m}$, $\varepsilon = 0.01$, $N_t = 400$) obtained with fully linear or Weak-Scatterer simulation.

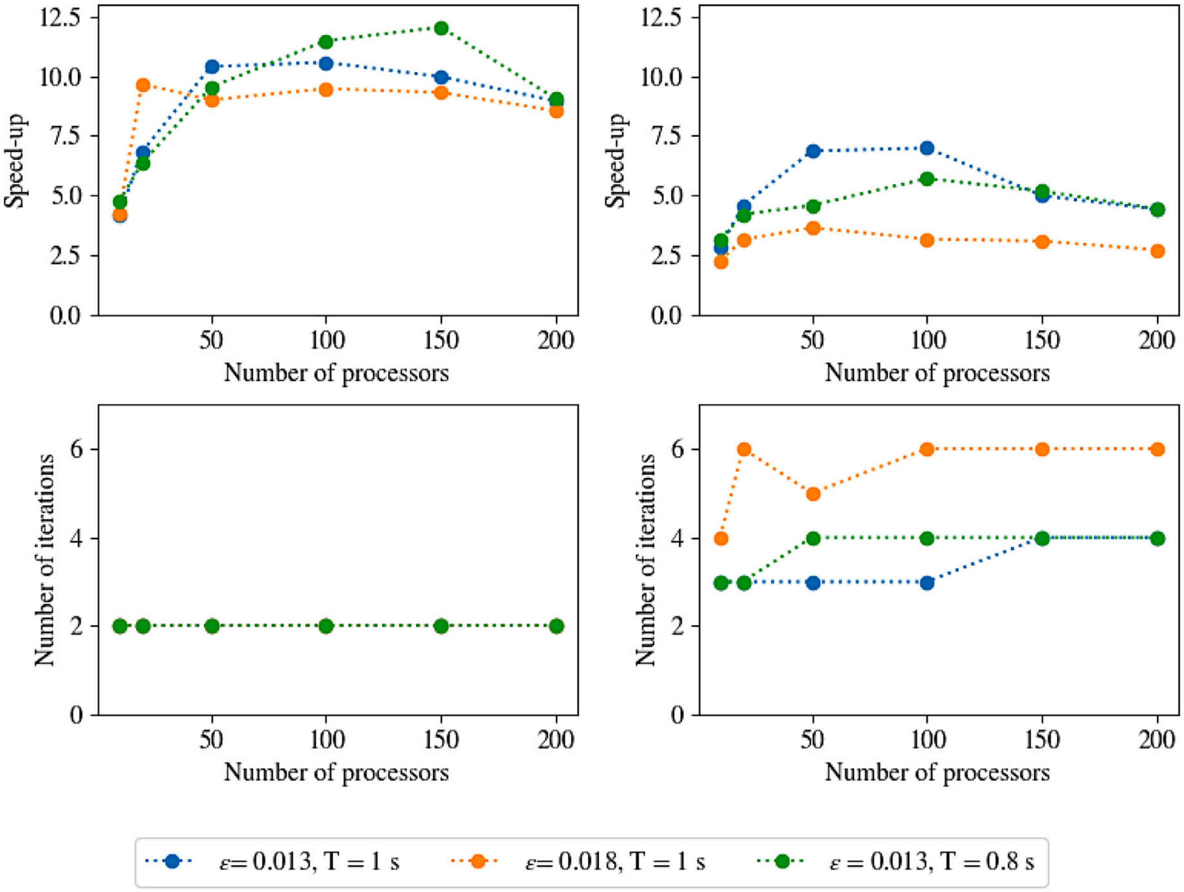

Fig. 15. Speed-up factor (top) and number of iterations required for the algorithm to converge (bottom) as a function of the number of processors ($N_t = 600$) for a tolerance $\alpha_{min} = 10^{-1}$ (left) and $\alpha_{min} = 10^{-2}$ (right).

1 s, the computational time is divided by 5. Note that the use of the weak-scatter method is worthwhile in this example albeit the small wave steepness $\varepsilon = 0.01$, as can be seen in Fig. 14 which shows that there are visible differences in the wave elevation when computed with the Weak-scatterer method and the fully linear method (coarse propagator).

For sake of comparison, Fig. 13 also shows the speed-up and number of iterations for a convergence criterion $\alpha_{min} = 10^{-2}$. It can be seen that, with increasing wave amplitude, the speed-up eventually decreases down to 1 (no acceleration). This is because, as the amplitude of the incident wave increases, the difference between the results provided by the fine and coarse propagators increases, which slows down the

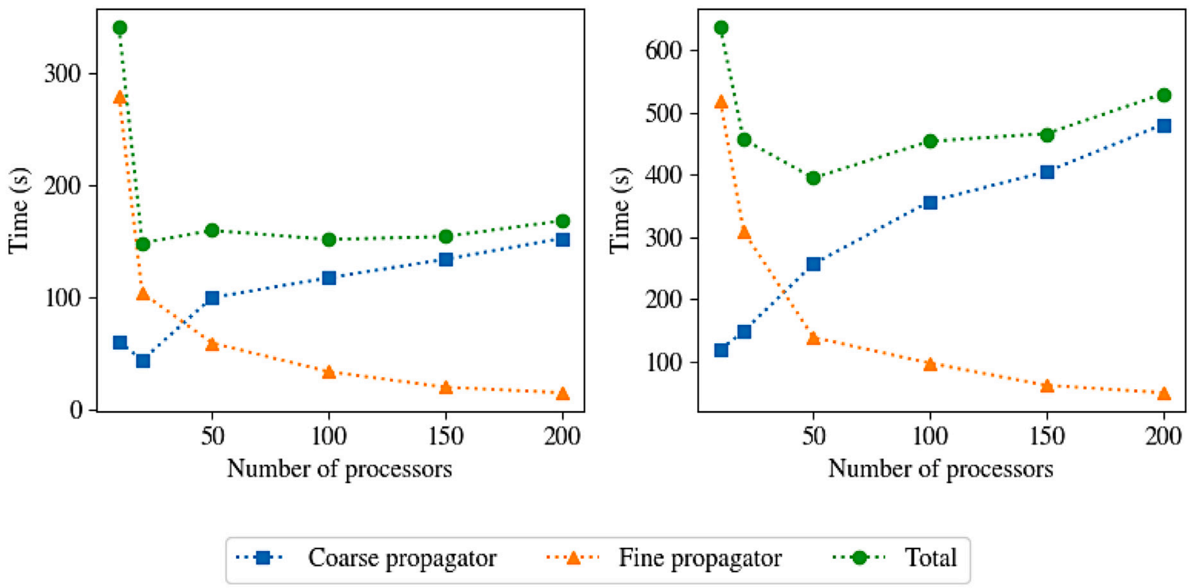

Fig. 16. Computational times associated with the coarse propagator alone, the fine propagator alone and the sum of the two as a function of the number of processors ($A = 0.01$ m, $\varepsilon = 0.013$, $T = 1$ s, $N_t = 600$) for a tolerance $\alpha_{min} = 10^{-1}$ (left) and $\alpha_{min} = 10^{-2}$ (right).

convergence of the algorithm. For example, the method converges in only 3 iterations for an incident wave of amplitude $A = 0.008$ m but only converges after 12 iterations for an incident wave of amplitude $A = 0.02$ m (Fig. 13).

4.3. *Influence of the number of processors*

Fig. 15 (top) shows the speed-up factor as function of the number of processors for three cases: a wave of wave period $T = 1$ s and wave steepness $\varepsilon = 0.013$, a wave with the same period but a different steepness $\varepsilon = 0.018$, and a wave of wave steepness $\varepsilon = 0.013$ but with shorter period $T = 0.8$ s. Overall, one can see that the computation is accelerated, but that the speed-up factor can be rather disappointing depending on the number of processors. Indeed, it appears that the speed-up factor increases with increasing number of processors up to a number of processors, after which it reduces. That limit number of processors seems to depend on the wave characteristics. However, overall, one can see that this limit is on the order of one hundred processors for the three sets of parameters that were considered. In addition, Fig. 15 (bottom) shows that the number of iterations required for the algorithm to converge increases slowly with the number of processors.

To explain why the speed-up factor reduces once a number of processors is exceeded, one should recall that the initial conditions given to each processor are computed from the results of the coarse and fine propagators (Eq. (39)). Because the fine propagator (based on the Weak-scatterer approach) and the coarse propagator (linear approach) use different meshes, it is necessary to interpolate the data to feed the updated initial conditions to the fine propagator. The time required for this operation being directly related to the number of processors, it becomes dominant when exceeding a number of processors. This is shown in Fig. 16. Therefore, for the parareal method to be efficient, one should not use a too large a number of processors.

5. Conclusions

In this study, we investigated whether the Parareal method can be used to reduce the computation time of a wave-structure interaction solver based on the weak-scatterer approach. We showed that after a sufficient number of iterations, the algorithm converges towards the solutions obtained by sequential simulation. For sufficiently low wave steepness, the method converges quickly enough to allow a significant reduction of the computational times. However, that benefit reduces with increasing wave steepness. We have also shown that there is an optimal number of processors which maximizes speed-up. This is due to the trade-off between the number of simulations which can be carried out in parallel and the cost of interpolation between the meshes used by the different propagators.

To improve the performance of the method, further work should focus on improving the speed of convergence for greater steepness. Moreover, the convergence criterion relates to the maximum value of the errors computed at the end of the simulations carried out by the different processors on each sub-interval. It can be seen that this error is greater the longer the simulation time associated with the processor is (Fig. 17). One strategy for increasing convergence speed is to use the Parareal algorithm over a shorter simulation time. Once the method has converged, the remaining time interval can then be treated in the same way (Fig. 18). Preliminary tests have shown encouraging results in this direction.

CRediT authorship contribution statement

Yohan Poirier: Writing – review & editing, Writing – original draft. **Julien Salomon:** Writing – review & editing, Writing – original draft. **Aurélien Babarit:** Writing – review & editing, Writing – original draft. **Pierre Ferrant:** Writing – review & editing, Writing – original draft. **Guillaume Ducrozet:** Writing – review & editing, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

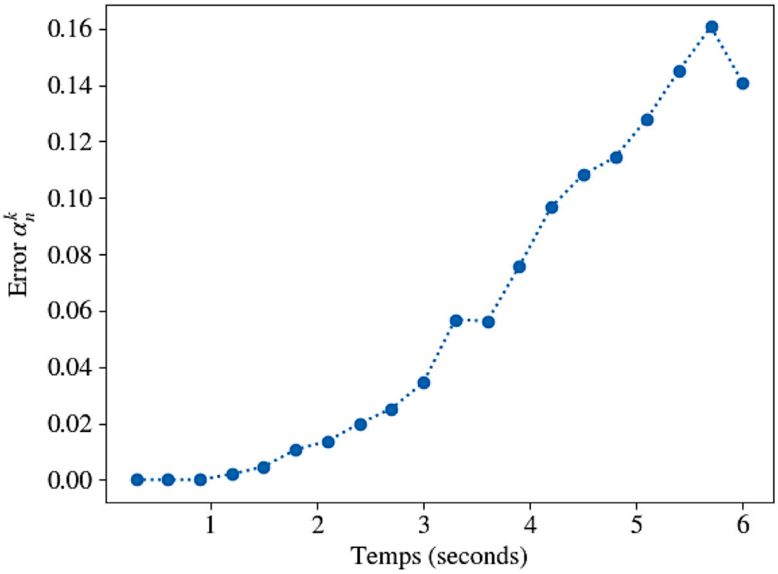

Fig. 17. Error associated with each processor for iteration $n = 3$ ($A = 0.025$ m, $\varepsilon = 0.03$, $T = 1$ s, $N_p = 20$, $\Delta t = 0.01$ s).

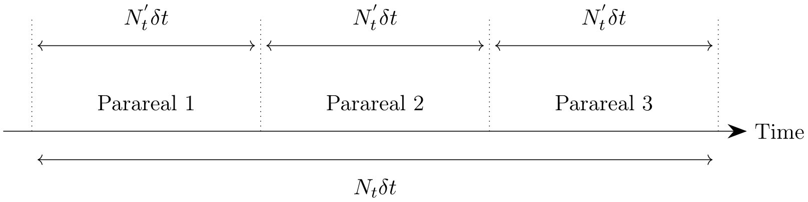

Fig. 18. Parareal slicing.

References

[1] Papillon Louis, Costello Ronan, Ringwood John. Boundary element and integral methods in potential flow theory: A review with a focus on wave energy applications. J Ocean Eng Mar Energy 2020;6:303–37. http://dx.doi.org/10.1007/s40722-020-00175-7.

[2] Lee C-H. WAMIT theory manual. Massachussets Institute of Technology, Department of Ocean Engineering; 1995.

[3] AQWA theory manual. ANSYS; 2013.

[4] Babarit Aurélien, Delhommeau Gérard. Theoretical and numerical aspects of the open source BEM solver NEMOH. In: 11th European wave and tidal energy conference. EWTEC2015, 2015.

[5] Luquet Romain. Simulation numérique de l'écoulement visqueux autour d'un navire soumis à une houle quelconque (Ph.D. thesis), École Centrale de Nantes; 2007.

[6] Bhinder Majid, Mingham Clive, Causon DM, Rahmati Mohammad, Aggidis George, Chaplin Robert. Numerical and experimental study of a surging point absorber wave energy converter. In: Proceedings of the 8th European wave and tidal energy conference. 2009, http://dx.doi.org/10.1115/OMAE2009-79392.

[7] Windt Christian, Davidson Josh, Ringwood John V. High-fidelity numerical modelling of ocean wave energy systems: A review of computational fluid dynamics-based numerical wave tanks. Renew Sustain Energy Rev 2018;93:610–30. http://dx.doi.org/10.1016/j.rser.2018.05.020, URL https://www.sciencedirect.com/science/article/pii/S1364032118303629.

[8] Pawlowski J S. A non-linear theory of ship motion in waves. In: Proceedings of the 19th symposium on naval hydrodynamics. 1992.

[9] Guerber Etienne, Benoit Michel, Grilli Stephan T, Buvat Clément. A fully nonlinear implicit model for wave interactions with submerged structures in forced or free motion. Eng Anal Bound Elem 2012;36(7):1151–63. http://dx.doi.org/10.1016/j.enganabound.2012.02.005, URL https://www.sciencedirect.com/science/article/pii/S0955799712000288.

[10] Dombre E, Benoit Michel, Violeau Damien, Peyrard C, Grilli S. Simulation of floating structure dynamics in waves by implicit coupling of a fully non-linear potential flow model and a rigid body motion approach. J Ocean Eng Mar Energy 2014;1:55–76. http://dx.doi.org/10.1007/s40722-014-0006-y.

[11] Colicchio Giuseppina, Greco Marilena, Lugni Claudio, Faltinsen Odd Magnus. Towards a fully 3D domain-decomposition strategy for water-on-deck phenomena. J Hydrodyn Ser B 2010;22(5, Supplement 1):462–7. http://dx.doi.org/10.1016/S1001-6058(09)60237-7, URL https://www.sciencedirect.com/science/article/pii/S1001605809602377.

[12] Kristiansen Trygve, Faltinsen Odd M. Gap resonance analyzed by a new domain-decomposition method combining potential and viscous flow DRAFT. Appl Ocean Res 2012;34:198–208. http://dx.doi.org/10.1016/j.apor.2011.07.001, URL https://www.sciencedirect.com/science/article/pii/S0141118711000538.

[13] Choi Youngmyung. Two-way coupling between potential and viscous flows for a marine application (Ph.D. thesis), Ecole Centrale de Nantes; 2019.

[14] Zhang Yi, Teng Bin, Gou Ying. Nonlinear modelling of a point-absorber wave energy converter based on the weak-scatterer approximation. Ocean Eng 2021;239:109924. http://dx.doi.org/10.1016/j.oceaneng.2021.109924, URL https://www.sciencedirect.com/science/article/pii/S0029801821012695.

[15] Tong Chao, Shao Yanlin, Bingham Harry B, Hanssen Finn-Christian W. A generalized weak-scatterer approximation for nonlinear wave–structure interaction in marine hydrodynamics. Mar Struct 2022;86:103292. http://dx.doi.org/10.1016/j.marstruc.2022.103292, URL https://www.sciencedirect.com/science/article/pii/S0951833922001289.

[16] Letournel Lucas, Chauvigne Camille, Gelly Baptiste, Babarit Aurelien, Ducrozet Guillaume, Ferrant Pierre. Weakly nonlinear modeling of submerged wave energy converters. Appl Ocean Res 2018;75:201–22.

[17] Chauvigne Camille, Letournel Lucas, Babarit Aurelien, Ducrozet Guillaume, Bozonnet Pauline, Gilloteaux Jean-Christophe, Ferrant Pierre. Progresses in the development of a weakly-nonlinear wave body interaction model based on the weak-scatterer approximation. In: of Mechanical Engineers American Society, editor. International conference on offshore mechanics and arctic engineering. Vol. 56574, 2015.

[18] Wuillaume Pierre-Yves, Babarit Aurelien, Rongere François, Lynch Mattias, Ferrant Pierre. Numerical simulation of lowering operations from the coupling between the composite-rigid-body algorithm and the weak-scatterer approach. Ocean Eng 2021;241(109997).

[19] LF Greengard, Rokhlin V. A fast algorithm for particle simulation. J Comput Phys 2001;73:325–48. http://dx.doi.org/10.1016/0021-9991(87)90140-9.

[20] Phillips Joel R, White Jacob K. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. IEEE Trans Comput Aided Des Integr Circuits Syst 1997;16:1059–72.

[21] Börm Steffen, Grasedyck Lars, Hackbusch Wolfgang. Introduction to hierarchical matrices with applications. Eng Anal Bound Elem 2003;27(5):405–22. http://dx.doi.org/10.1016/S0955-7997(02)00152-2, URL https://www.sciencedirect.com/science/article/pii/S0955799702001522. Large scale problems using BEM.

[22] Teng Bin, Song Zhi-jie. Extension of the frequency-domain pFFT method for wave structure interaction in finite depth. China Ocean Eng 2017;31:322–9. http://dx.doi.org/10.1007/s13344-017-0038-x.

[23] Lions Jacques-Louis, Maday Yvon, Turini Gabriel. Résolution d'edp par un schéma en temps <pararéel >. In: Série i, mathématique, C R L'acad Sci In: Série i, mathématique.2001.

[24] Maday Yvon, Turinici Gabriel. Parallel in time algorithms for quantum control: Parareal time discretization scheme. Int J Quantum Chem 2003;93:223–8. http://dx.doi.org/10.1002/qua.10554.

[25] Trindade J, Pereira Jose. Parallel-in-time simulation of the unsteady Navier–Stokes equations for incompressible flow. Internat J Numer Methods Fluids 2004;45:1123–36. http://dx.doi.org/10.1002/fld.732.

[26] Fischer Paul F, Hecht Frédéric, Maday Yvon. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In: Barth Timothy J, Griebel Michael, Keyes David E, Nieminen Risto M, Roose Dirk, Schlick Tamar, Kornhuber Ralf, Hoppe Ronald, Périaux Jacques, Pironneau Olivier, Widlund Olof, Xu Jinchao, editors. Domain decomposition methods in science and engineering. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005, p. 433–40.

[27] Glimberg Stefan Lemvig. Designing scientific software for heterogeneous computing: with application to large-scale water wave simulations (Ph.D. thesis), DTU compute PHD-2013, Technical University of Denmark; 2013.

[28] Nielsen Allan S. Feasibility study of the parareal algorithm. 2012, URL https://api.semanticscholar.org/CorpusID:123881089.

[29] Samaddar D, Newman DE, Sanchez Raul. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. J Comput Phys 2010;229:6558–73. http://dx.doi.org/10.1016/j.jcp.2010.05.012.

[30] Wuillaume Pierre-Yves. Numerical simulation of installation operations for offshore wind farms (Ph.D. thesis), École centrale de Nantes; 2019, URL https://theses.hal.science/tel-02390021.

[31] Fenton JD, Rienecker MM. A Fourier method for solving nonlinear water-wave problems: Application to solitary-wave interactions. J Fluid Mech 1982;118:411–43. http://dx.doi.org/10.1017/S0022112082001141.

[32] Ducrozet Guillaume, Bouscasse Benjamin, Gouin Maïté, Ferrant Pierre, Bonnefoy Félicien. CN-Stream: Open-source library for nonlinear regular waves using stream function theory. 2019, arXiv:1901.10577.

[33] Ducrozet Guillaume, Bonnefoy Félicien, Le Touzé David, Ferrant Pierre. HOS-ocean: Open-source solver for nonlinear waves in open ocean based on high-order spectral method. Comput Phys Comm 2016;203:245–54. http://dx.doi.org/10.1016/j.cpc.2016.02.017, URL https://www.sciencedirect.com/science/article/pii/S0010465516300327.

[34] Bai W, Eatock Taylor R. Higher-order boundary element simulation of fully nonlinear wave radiation by oscillating vertical cylinders. Appl Ocean Res 2006;28(4):247–65. http://dx.doi.org/10.1016/j.apor.2006.12.001, URL https://www.sciencedirect.com/science/article/pii/S0141118706000848.

[35] Cointe Raymond. Quelques aspects de la simulation numérique d'un canal à houle (Ph.D. thesis), Marne-la-vallée, ENPC; 1989.

[36] Tanizawa Katsuji. A nonlinear simulation method of 3-D body motions in waves (1st report). J Soc Naval Archit Japan 1995;1995(178):179–91. http://dx.doi.org/10.2534/jjasnaoe1968.1995.178_179.

[37] Letournel Lucas, Ducrozet Guillaume, Babarit Aurélien, Ferrant Pierre. Proof of the equivalence of Tanizawa–Berkvens' and Cointe–van Daalen's formulations for the time derivative of the velocity potential for non-linear potential flow solvers. Appl Ocean Res 2017;63:184–99. http://dx.doi.org/10.1016/j.apor.2017.01.010, URL https://www.sciencedirect.com/science/article/pii/S0141118716302693.

[38] Letournel Lucas. Développement d'un outil de simulation numérique basé sur l'approche "Weak-Scatterer" pour l'étude des systèmes houlomoteurs en grands mouvements (Ph.D. thesis), École Centrale de Nantes; 2015.

[39] Chauvigné Camille. Tenue à la mer d'un flotteur animé de grands mouvements pour les Energies Marines Renouvelables (Ph.D. thesis), École Centrale de Nantes; 2016.

[40] Chrisochoides Nikos, Nave Démian. Parallel delaunay mesh generation kernel. Internat J Numer Methods Engrg 2003;58(2):161–76. http://dx.doi.org/10.1002/nme.765, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.765. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.765.

[41] Ito Yasushi, Shih Alan M, Erukala Anil K, Soni Bharat K, Chernikov Andrey, Chrisochoides Nikos P, Nakahashi Kazuhiro. Parallel unstructured mesh generation by an advancing front method. Math Comput Simulation 2007;75(5):200–9. http://dx.doi.org/10.1016/j.matcom.2006.12.008, URL https://www.sciencedirect.com/science/article/pii/S0378475406003351. Applied Scientific Computing: Advanced Grid Generation, Approximation and Simulation.

[42] Grigori Laura, Hirstoaga Sever Adrian, Nguyen Van-Thanh, Salomon Julien. Reduced model-based parareal simulations of oscillatory singularly perturbed ordinary differential equations. J Comput Phys 2021;436:110282, URL https://hal.science/hal-03104042.

[43] Grigori Laura, Hirstoaga Sever A, Salomon Julien. A parareal algorithm for a highly oscillating Vlasov-Poisson system with reduced models for the coarse solving. Comput Math Appl 2023;130:137–48. http://dx.doi.org/10.1016/j.camwa.2022.12.004, URL https://www.sciencedirect.com/science/article/pii/S0898122122005065.

[44] Pelletier Karine. Simulations au second ordre des interactions houle-structure en régime instationnaire (Ph.D. thesis), Université de Nantes; 2002.