Proceedings of the ASME 37th International Conference on Ocean, Offshore and Arctic Engineering
OMAE2018
June 17-22, 2018, Madrid, Spain

OMAE2018-77309

ADAPTIVE GRID REFINEMENT FOR TWO-PHASE OFFSHORE APPLICATIONS

Peter van der Plas
Arthur E.P. Veldman*
Henk Seubers
Institute for Mathematics and Computer Science
University of Groningen
P.O. Box 407, 9700 AK Groningen
The Netherlands
{a.e.p.veldman, p.van.der.plas, h.seubers}@rug.nl

Joop Helder
Ka-Wing Lam
MARIN
P.O. Box 28, 6700AA Wageningen
The Netherlands
{j.helder,k.w.lam}@marin.nl

ABSTRACT

In the past, the CFD simulation method ComFLOW has been successfully applied in a wide range of offshore applications, involving wave simulations and impact calculations. In many of these calculations the area of interest comprises a small part of the domain and remains fixed in time, which allows for efficient grid refinement by means of grid stretching or static local refinement. However, when trying to accurately resolve the surface dynamics and kinematics of irregular and breaking waves, the resolution requirements are strongly time-dependent and difficult to predict in advance. Efficient grids can only be obtained by means of time-adaptive refinement. A Cartesian block-based refinement approach is followed which allows for efficient grid adaptation, with moderate overhead. An array-based data structure is employed which exploits the semi-structured nature of the Cartesian block grid. Currently we are testing the method with the simulation of lifeboat drops in regular and irregular wave conditions. This poses several challenges such as accurately imposing the incoming waves and modifying the absorbing boundary conditions to support two-phase flow. To reduce the wall-clock time, the simulation method has been parallelized.

1 INTRODUCTION

Numerical modeling tools start to have increasing importance in engineering practice, as they can help to provide insight in flow situations without the need of expensive and often time-consuming model tests. Given the continuous increase of computational power, the possibilities of numerically simulating entire test setups, e.g. in a wave tank, become more and more feasible. By means of parallelization, clusters of computers can be employed as so-called *numerical* wave- or towing tanks. Finally, numerical modeling becomes crucial for problems that cannot be fitted in a test basin because scaling effects from full to model scale play a role.

A commonly encountered offshore engineering application involves the calculation of wave impacts on an offshore structure. To obtain details of flow around complicated structures, or in otherwise complicated environments, one has to resort to simulation of the full Navier–Stokes equations (the "CFD" or hydrodynamic modeling approach). The present work concentrates on this last type of applications and is performed in the context of the CFD solver ComFLOW [1–3].

In ComFLOW, the flow variables are discretized on a staggered Cartesian grid. A second-order accurate and energy-conservative finite-volume discretization of the flow equations is used to minimize the amount of artificial diffusion [4]. The boundaries are accounted for by means of a Volume-of-Fluid method [5] for the modeling of the free surface, and a cut-cell discretization for the geometry [6,7] in symmetry-preserving formulation [8,9].

The simplest solution for obtaining more accurate results is to apply mesh refinement across the entire computational do-

*Address all correspondence to this author.

main. For flow problems that are characterized by a large variation of localized physical phenomena this approach is not practical, because of its high computational cost. Given the limitations of current computer architectures, in many complex problems a sufficient resolution can only be obtained by implementing some form of local grid refinement, in which the grid resolution is only increased in areas where this is required either by physics or numerics. Local refinement provides an efficient tool to capture more details around boundaries where the solution has large gradients, but it also allows for low-cost expansion of the solution domain by coarsening the grid towards the boundaries. Grid refinement methods are therefore also grid coarsening methods.

In the paper, we will give a short overview of our grid refinement approach. In particular, the discretization around refinement interfaces receives attention. After a presentation of the underlying mathematical model in Sec. 2, Sec. 3 gives a sketch of the discrete treatment of the various terms in the flow equations. Finally, in Sec. 4 two examples are presented of applications from the offshore industry: a dambreak problem simulating green-water impact on deck, and a falling life boat.

2 FLOW MODEL

Incompressible, turbulent fluid flow can be modelled by means of the Navier–Stokes equations.

$$M\mathbf{u} = 0, \qquad \frac{\partial \mathbf{u}}{\partial t} + C(\mathbf{u})\mathbf{u} + Gp - V\mathbf{u} = \mathbf{f}. \tag{1}$$

Here M is the divergence operator, which describes conservation of mass. Conservation of momentum is based on the convection operator $C(\mathbf{u})\mathbf{v} \equiv \nabla(\mathbf{u} \otimes \mathbf{v})$, the pressure gradient operator $G = \nabla$, the viscous diffusion operator $V(\mathbf{u}) \equiv \nabla \cdot \nu \nabla \mathbf{u}$ and a forcing term $\mathbf{f}$. The kinematic viscosity is denoted by ν. Turbulence is modelled by means of large-eddy simulation (LES) using a low-dissipation model [10, 11].

The free-surface location is indicated by the Volume-of-Fluid method [5] and is reconstructed by means of Youngs' PLIC-VOF method [12, 13].

The Navier–Stokes equations (1) are discretized on a staggered Arakawa C-grid [14]. The second-order finite-volume discretization of the continuity equation at the 'new' time level $\cdot^{n+1}$ is given by

$$\mathsf{M}_0 \mathbf{u}^{n+1} = -\mathsf{M}_\Gamma \mathbf{u}_\Gamma^{n+1}, \tag{2}$$

where M_0 acts on the interior of the domain and M_Γ acts on the boundaries (with $\mathbf{u}_\Gamma$ denoting the velocity at the boundary). In the momentum equation, convection $C(\mathbf{u}_h)$ and diffusion V are discretized explicitly in time. The pressure gradient is discretized at the new time level. In this exposition, for simplicity reasons the first-order forward Euler time integration will be used. In the actual calculations, a second-order Adams–Bashforth method is being applied.

With the diagonal matrix Ω containing the geometric size of the control volumes, the discretized momentum equation becomes

$$\Omega \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} = -\mathsf{C}(\mathbf{u}^n)\,\mathbf{u}^n + \mathsf{V}\mathbf{u}^n - \mathsf{G}\mathsf{p}^{n+1} + \mathbf{f}. \tag{3}$$

For divergence-free velocity fields $\mathbf{u}$, the conservative discrete convection operator is skew-symmetric, such that convection does not contribute to the energy balance [4]. To make the discretization fully energy-preserving, the discrete gradient operator and the divergence operator are each other's negative transpose, i.e. $\mathsf{G} = -\mathsf{M}_0^T$, thus mimicking the analytic symmetry $\nabla = -(\nabla\cdot)^T$. In this way, also the work done by the pressure vanishes discretely.

The solution of the discrete Navier–Stokes equations is split into two steps. Firstly, an auxiliary variable $\tilde{\mathbf{u}}$ is defined through

$$\Omega \frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\delta t} = -\mathsf{C}(\mathbf{u}^n)\,\mathbf{u}^n + \mathsf{V}\mathbf{u}^n + \mathbf{f}, \tag{4}$$

with which the discrete momentum equation (3) can be reformulated as

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \delta t\, \Omega^{-1} \mathsf{G}\mathsf{p}^{n+1}. \tag{5}$$

Secondly, by imposing discrete mass conservation (2) at the new time level, substitution of (5) results in a discrete Poisson equation for the pressure:

$$\delta t\, \mathsf{M}_0 \Omega^{-1} \mathsf{G}\mathsf{p}^{n+1} = \mathsf{M}_0 \tilde{\mathbf{u}} + \mathsf{M}_\Gamma \mathbf{u}_\Gamma^{n+1}. \tag{6}$$

3 DISCRETIZATION NEAR REFINEMENT INTERFACES

At refinement interfaces it is no longer possible to apply the regular discretization as indicated above. Modifications have to be introduced, which will be described in some detail below. The treatment of the refinement interfaces is complicated by the fact that the equations are discretized on a staggered grid: different variables will have different offsets with respect to one another. Most of the grid refinement methods encountered in the literature are applied on collocated grids. Yet, methods for staggered grids do exist [15–17].

3.1 Refinement interfaces on staggered grids

On a staggered grid one has to distinguish between interfaces that coincide with the faces of continuity volumes, called *continuity interfaces* (Fig. 1(left)), and interfaces that coincide with the faces of momentum-conservation volumes, called *staggered interfaces* (Fig. 1(right)).

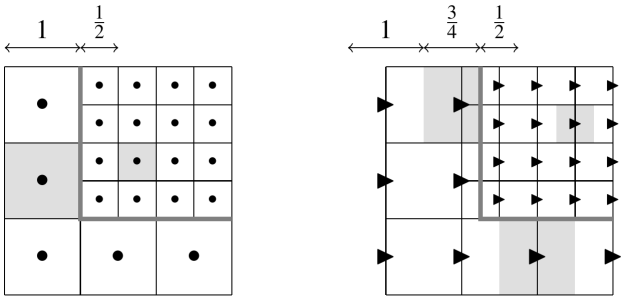


FIGURE 1. REFINEMENT INTERFACES ON THE STAGGERED GRID. SEVERAL CONSERVATION VOLUMES ARE SHADED TO ILLUSTRATE TYPICAL VOLUME SIZES IN THE VICINITY OF INTERFACES. LEFT: CONTINUITY EQN.; RIGHT: x-MOMENTUM EQN.

A choice has to be made concerning the discretization stencils around the refinement interface. One option is to extend the regular stencils in one subdomain into the adjacent domain. Virtual grid points are introduced at the other side of the interface which are found from interpolation between the local grid points. This keeps the structure of the discretization stencils regular, at the cost of interpolation activities. Such an approach fits in naturally with overlapping subgrids as in [16], and with multilevel grids [18, 19].

The virtual grid points can be considered as auxiliary variables built from combinations of the local points. We prefer to express the discretization stencil directly in these local variables, which makes the stencils irregular. This facilitates the analysis of the schemes, as the diffusive interpolation step is avoided. Therefore, we opt for a *non-overlapping* choice for the subdomains.

3.2 Accuracy

Symmetry properties Ideally, the interface treatment should result in a discretization that has the same order of accuracy and the same conservation properties as that of the existing discretization on regular grids; for a discussion see e.g. [17, Ch. 2]. The discretization in ComFLOW discretely conserves mass and momentum. It is also conserves energy when combined with an appropriate time integration method, like the midpoint rule [20]. It is highly desirable to maintain mass- and momentum-conservation across the refinement interfaces, with mass-conservation clearly being the very minimum requirement. For stability of the method it is also convenient if the discretization is energy-conservative across refinement boundaries, but not at all costs: accuracy and robustness as well as ease of implementation also play a role.

Reflections The discretization at the interface should reduce as much as possible the occurrence of spurious reflections. Within the ComFLOW project, research regarding this topic has already been performed in relation to the discretization at open boundaries [13, 21, 22]. Theoretical investigation of reflections on non-uniform grids mostly concerns the one-dimensional linear advection equation; research on this topic can be traced back to the early eighties, e.g. [23]. More recently, in [24] it is argued that second-order accurate interpolation reduces reflections in the discrete wave equation. In [25] it is shown for the discretized shallow-water equations that avoiding reflection of under-resolved wave components is only possible by introducing some form of dissipation, i.e. by sacrificing energy conservation.

Accuracy near refinement interfaces Traditionally the accuracy of a numerical scheme is investigated by determining the order of magnitude of the *local truncation error*. Generally, a second-order local truncation error is sufficient for obtaining a second-order accurate solution, but it may not always be necessary: the actual order of global accuracy may be higher than the order of the local truncation error, especially on non-uniform grids, e.g. [26]. Schemes that achieve higher-order accuracy than the order of their local truncation error have been called *supra-convergent* by Kreiss et al. [27].

Following [28], several local grid refinement schemes were introduced based on compact, symmetric discretization schemes, e.g. [29, 30]. A large number of basic discretization schemes are seemingly first (or zeroth) order in the vicinity of non-uniform grid spacing, but are actually second (or first) order accurate [26]. This suggests that a compact discretization scheme with first-order local truncation error could already be enough to obtain an effectively second-order accurate interface treatment.

3.3 Conservation properties

It is non-trivial to design a conservative and at the same time second or higher-order accurate discretization scheme around refinement interfaces. This holds for the discretization of the divergence operator (mass conservation) but also for the discretization of the pressure gradient, convection and diffusion (momentum conservation). Let us assume that on the regular part of the grid the spatial derivatives are approximated using central difference schemes. Consider, for example, the discretization of the first-order derivative $\partial/\partial y$ around the point $x_{-\frac{1}{4},+\frac{1}{2}}$ (cf. the divergence operator). The situation is depicted in Fig. 2. For simplicity a uniform grid is assumed, with grid spacings as indicated

in the same figure, and $v_{0,0}$ is located at $x = 0$.

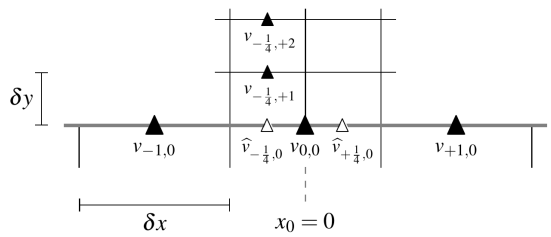


FIGURE 2. INTERPOLATION OF MISSING VELOCITIES AT A REFINEMENT INTERFACE WITH RATIOS $r = 2$. THE MISSING VELOCITIES (OR FLUXES) ARE INDICATED BY △

If the variable $v_{-\frac{1}{4},0}$ were directly available, the central scheme for the first-order derivative in y-direction would be second-order accurate

$$\frac{v_{-\frac{1}{4},+1} - v_{-\frac{1}{4},0}}{2\delta y} = \left.\frac{\partial v}{\partial y}\right|_{x=x_{-\frac{1}{4},+\frac{1}{2}}} + \mathscr{O}\left(\delta y^2\right). \tag{7}$$

In the presence of a refinement interface the velocity $v_{-\frac{1}{4},0}$ has to be approximated, leading to the virtual velocity $\widehat{v}_{-\frac{1}{4},0}$ which, in turn, is approximated by interpolation along the interface (here in x-direction). Let us denote the interpolation error by ε, in order to write $\widehat{v}_{-\frac{1}{4},0} = v_{-\frac{1}{4},0} + \varepsilon$. Let n denote the order of accuracy of the interpolation, in order to write $\varepsilon = \mathscr{O}(\delta^n)$. Then, for the central discretization of $\partial v/\partial y$ we get

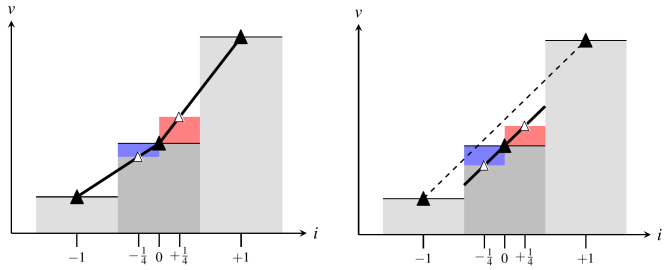
$$\frac{v_{-\frac{1}{4},+1} - \widehat{v}_{-\frac{1}{4},0}}{\delta y} = \frac{v_{-\frac{1}{4},+1} - v_{-\frac{1}{4},0} - \varepsilon}{\delta y} = \frac{\partial v}{\partial y} + \mathscr{O}\left(\delta y^2, \frac{\delta^n}{\delta y}\right).$$

Hence, for a consistent discretization of first-order derivatives (e.g. divergence, gradient, convection) the interpolation of missing variables should be at least second-order accurate.

When applying linear interpolation, the standard approach is to use a weighted average of the direct neighbours as shown in Fig. 3a:

$$\widehat{v}_{\pm\frac{1}{4},0} = (3v_{0,0} + v_{\pm1,0})/4.$$

This interpolation is *non-symmetric* because the linear slope is approximated differently in the left and right part of the underlying coarse cell. Because of this asymmetry, the two interpolated values do not add up to the value of the underlying coarse variable, more precisely $\frac{1}{2}\widehat{v}_{-\frac{1}{4},0} + \frac{1}{2}\widehat{v}_{\frac{1}{4},0} = v_{0,0} + \frac{1}{8}(v_{-1,0} - 2v_{0,0} + v_{+1,0}) \approx v_{0,0} + \frac{1}{8}\delta x^2(\partial^2 v/\partial x^2)$. This makes the interpolation *non-conservative*.



(a) Standard linear interpolation (b) Symmetric linear interpolation

FIGURE 3. ONE-DIMENSIONAL ILLUSTRATION OF STANDARD *NON-SYMMETRIC* LINEAR INTERPOLATION ON A TWO-POINT STENCIL AND *SYMMETRIC* INTERPOLATION ON A THREE-POINT STENCIL. THE GRAY COLUMNS ILLUSTRATE THE CELL AVERAGES IN THE COARSE CELLS; THE INTERPOLATED VALUES ARE INDICATED BY △; THE COLORED AREAS ILLUSTRATE THE DIFFERENCE OF THE INTERPOLATIONS FROM THE AVERAGE IN THE UNDERLYING COARSE CELL.

A better approach is to use *symmetric linear interpolation* in which a single linear slope is used for the interpolation on both sides of the point x_0, as illustrated in Fig. 3b. The same interpolant $\widehat{v}(x)$ is used to approximate both $\widehat{v}_{+\frac{1}{4},0}$ and $\widehat{v}_{-\frac{1}{4},0}$, resulting in

$$\widehat{v}(x) = v_{0,0} + \frac{1}{2}\frac{v_{+1,0} - v_{-1,0}}{\delta x}x, \qquad \varepsilon = \mathscr{O}\left(x^2\right). \tag{8}$$

This interpolation leads to an interface discretization scheme for $\partial v/\partial y$ that is first-order accurate ($n = 2, q = 1$). Moreover, it is directly seen that it is *c*onservative, i.e. the refined fluxes add up to the underlying coarse flux

$$\frac{\delta x}{2}\widehat{v}(+\delta x/4) + \frac{\delta x}{2}\widehat{v}(-\delta x/4) = \delta x\, v_{0,0}.$$

We can conclude that it is difficult to obtain a conservative flux difference scheme at interfaces that is both conservative *and* second or higher-order accurate, as was also observed in [15]. Only a limited number of authors tries to obtain a higher-order energy-conservative treatment for refinement interfaces, e.g. [17, 31–33], but these concern only one- or two-dimensional grids.

Thus, in the current work, for the divergence and pressure gradient a first-order accurate approach is followed based on linear interpolation, therewith ensuring mass conservation (see Section 3.4). For most industrial applications this is sufficient and poses no problem if the interfaces are located in relatively smooth regions of the flow.

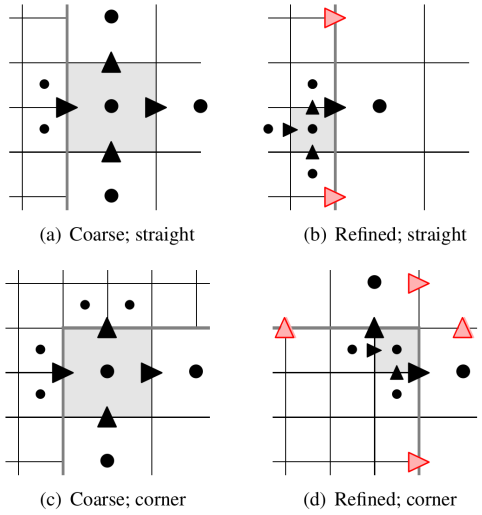
3.4 Interface schemes for divergence and gradient

The first decision that has to be made concerns the variable arrangement at the refinement interface. Due to the staggered grid layout, a refinement boundary contains a velocity (gradient) interface and a pressure (divergence) interface; cf. the discussion in Sect. 3.1. The refinement interface can be designed in two fashions:

- *Use an interface with refined velocities and coarse pressure variables.* At the interface the momentum equation is discretized at the refined level.
- *Use an interface with coarse velocities and refined pressure variables.* At the interface the momentum equation is discretized at the coarse level.

As motivated in [34] by testing a large number of flow situations, we have a (slight) preference for the second option.

Several methods found in the literature apply higher-order (typically quadratic) interpolation of missing variables in order to ensure a certain local truncation error of the divergence and/or gradient operator. In [35] the missing pressure is approximated by means of bi-quadratic interpolation; another example of higher-order bi-directional interpolation can be found in [15]. In all cases additional non-zero coefficients are introduced and the Poisson matrix is no longer symmetric.



(a) Coarse; straight (b) Refined; straight

(c) Coarse; corner (d) Refined; corner

FIGURE 4. VISUALIZATION OF THE PRESSURE POISSON STENCIL FOR DISCRETIZATION ON A 2-D GRID. THE VELOCITY VARIABLES ▶ AND ▲ ARE USED IN $\mathsf{M}u_h^{n+1}$. THE ACTUAL STENCIL FOR $\mathsf{MG}p_h^{n+1}$ CONSISTS OF THE PRESSURE VARIABLES ●. NOTE THAT THE VARIABLES ▶ AND ▲ ARE USED TO INTERPOLATE THE 'MISSING' MASS FLUXES IN THE FINE STENCIL.

Skipping many technical details, it is illustrative to show the stencils that result for the discrete pressure Poisson equation in Fig. 4. To solve it, the resulting Poisson matrix is preferrably symmetric positive definite to guarantee convergence of the solvers as currently implemented in ComFLOW: Krylov solvers with grid-point dependent preconditioning and parallelized with grid partitioning based on orthogonal recursive bisection (ORB) [36, 37]. It is not straightforward to find a theoretical proof of the (non)existence of this matrix property. The discretization scheme is diagonally dominant in coarse interface cells and away from interfaces, but this is not the case for fine interface cells. However, in all test cases considered so far, the presence of refinement interfaces did not cause the solvers to diverge.

4 OFFSHORE APPLICATIONS

4.1 Dambreak

Green water loading is characterized by strongly varying free-surface dynamics and therefore a good test for a local refinement method. Once the water starts to hit the ship, high grid resolution is required to capture all relevant flow details, whereas during the initial phase a coarse-grained approximation may already be sufficient.

For current purposes, we consider an idealized problem based on experiments that were performed at MARIN [1]. The flow domain consists of a closed box with dimensions $3.25 \times 1.0 \times 1.0$ m. A reservoir in the right part of the domain is filled with water up to a level of 0.55 m. The water is released by almost instantaneously opening a hatch. In the middle of the domain a small block ('box on deck') is placed that will be impacted by the released water mass. Eight pressure transducers were attached to the object and the relative water height probes were placed at four locations in the domain as shown in Fig.5. For experimental data see [38].

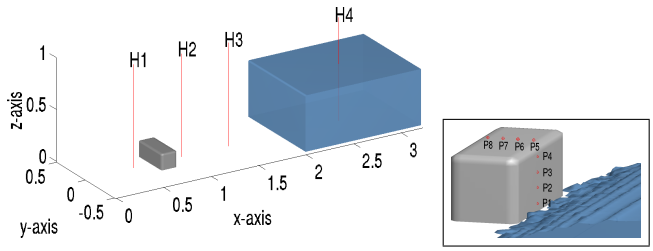
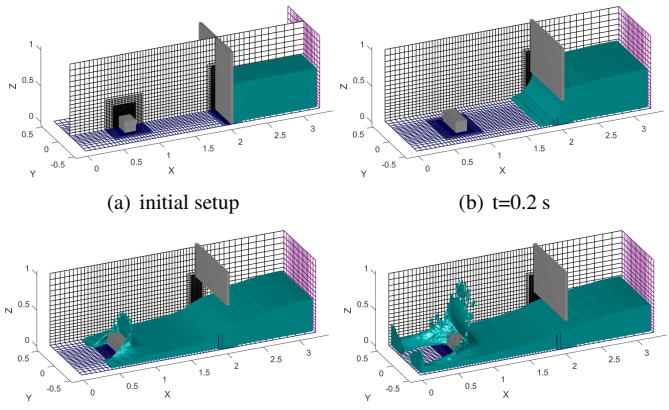


FIGURE 5. MEASUREMENT POSITIONS FOR WATER HEIGHTS (LEFT) AND PRESSURES (RIGHT) IN THE DAMBREAK EXPERIMENT

The setup of the experiment is fairly simple. Before hitting the object at around $t = 0.45$ s, the flow is essentially two-

dimensional (e.g. Fig. 6b). During the first second of the experiment a 2-D model may therefore provide reasonable results. After impact the flow becomes three-dimensional in nature and demonstrates a violent sloshing behaviour (e.g. Fig. 6d). The dambreak experiment has been considered in previous studies [1, 13] and good overall agreement was observed between experiment and CFD simulations.

In the present study we re-investigate the simulation setup and aim at higher grid resolutions by exploiting local grid refinement. First, a 2-D study is described of the first moments of the flow, to study the influence of including the moving door in the simulations. Because these simulations are relatively cheap we can thoroughly investigate grid convergence and assess the performance of the local grid refinement method. Thereafter, the simulations of the full experiment have been carried out with a 3-D model.



(a) initial setup (b) t=0.2 s

(c) t=0.6 s (d) t=0.8 s

FIGURE 6. ILLUSTRATION OF THE SIMULATION SETUP AND A NUMBER OF SNAPSHOTS. A UNIFORM RESOLUTION OF 5 [MM] WAS USED AROUND THE BOX AND THE RESERVOIR DOOR. FOR VISUALIZATION PURPOSES THE GRID IS DEPICTED AT A TWO TIMES COARSER RESOLUTION.

With and without door The video recordings show that only after about a quarter second the door is fully detached from the water. Yet, at that moment, a significant amount of water has already entered the left part of the domain. In previous studies a door was not included and the water was simply released instantaneously. The question arises whether this simplification is justified.

Thus, simulations were performed with and without inclusion of a moving door in order to study its influence. Only the first 0.75 s of the experiment were considered, and these simulations were performed on a 2-D grid to make high grid resolution feasible. The door positions recorded during the experiment are used for modeling the door and for time synchronization. Free-slip conditions are applied along all solid boundaries. The algorithm for inclusion of moving rigid bodies is described in [39]. The simulations were performed on uniform grids with resolutions ranging between 5 mm (100k grid cells) and 1.25 mm (1.6M cells).

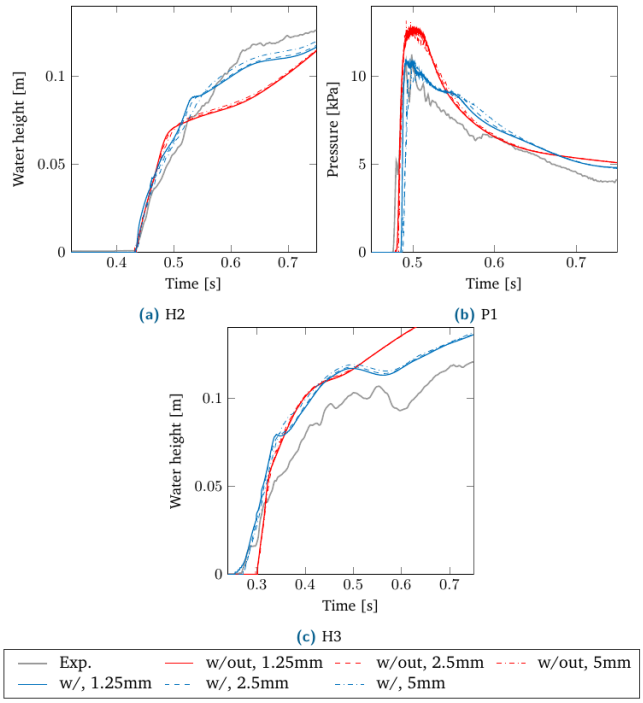


FIGURE 7. NUMERICAL RESULTS OF 2-D SIMULATIONS WITH AND WITHOUT RESERVOIR DOOR.

To assess the effect of modeling the moving door, we look at water height probe H2 and pressure sensor P1 (see Fig. 5). The numerical results, outlined in Fig. 7, show that the solution does not change much upon grid refinement from 5 mm to 1.25 mm. This justifies to use the grid with 5 mm resolution for the three-dimensional simulations discussed below.

Whether or not the door is modelled makes a much larger difference. The solution series without door under-predicts the water height in front of the object but over-predicts the pressure peak of the first impact (red graphs in Fig. 7). After inclusion of the reservoir door, the results improve considerably in comparison with the experiment (blue graphs in Fig. 7). In particular the pressure peak is predicted well and only a small time lag is observed. One has to keep in mind that there exist experimental uncertainties of the order of 1 cm in the actual shape and position of the door as well as in the shape and volume of the initial water column.

Local grid refinement / coarsening As a starting point a stretched reference grid is produced with uniform resolution $h = 1.25$ mm in the areas of interest. The reference grid is locally coarsened up to three times using the local grid refinement method (as illustrated in Fig. 6). From the results shown in Fig. 8, it can be seen that the numerical solution is hardly affected by the coarsening of the grid. In comparison with the experimental results, none of these grids can be called better that the others. Yet the difference in comptational effort is significant. The grid with the most levels of grid coarsening (fn-lgr3) has about 10 times less grid points inside the fluid, and is over 5 times as 'cheap' in wall-clock time (the code runs parallel on 24 cores), as can be inferred from Table 1a.

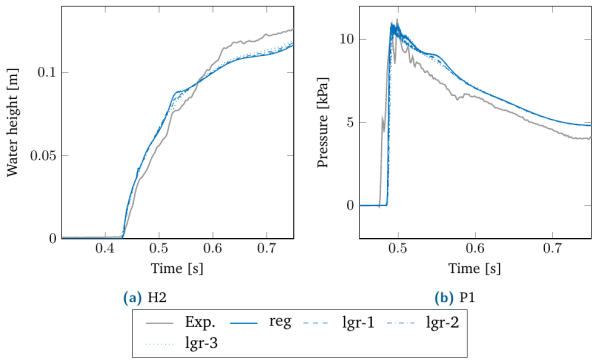


FIGURE 8. NUMERICAL RESULTS OF 2-D SIMULATIONS *WITH* MOVING RESERVOIR DOOR USING VARIOUS GRID CONFIGURATIONS.

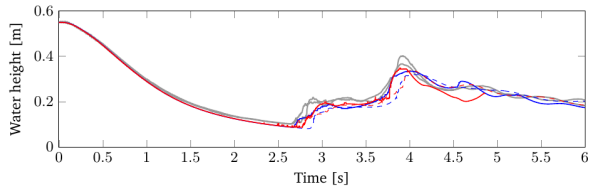
(a) 2-D simulations

run	Cells (total)	Fluid cells	d_0 [mm]	time steps	iter	wall clock [h]
fn–reg	5.0M	0.42M	1.25	5.6k	1.7M	6.8
fn–lgr1	1.8M	0.13M	1.25	4.9k	0.54M	2.1
fn–lgr2	1.1M	62k	1.25	4.8k	0.46M	1.3
fn–lgr3	0.91M	45k	1.25	4.7k	0.47M	1.2

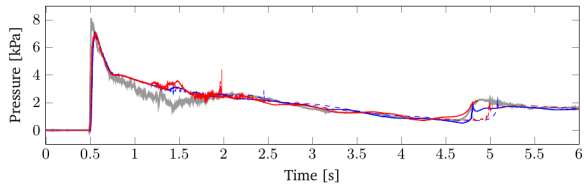
(b) 3-D simulations

run	Cells (total)	Fluid cells	d_0 [mm]	time steps	iter	wall clock [h]
fn–reg	6.5M	1.6M	5	21k	3.2M	296
fn–lgr1	1.5M	0.35M	5	13k	1.5M	29
med–reg	1.2M	0.24M	10	12k	1.0M	18
med–lgr1	0.29M	53k	10	4.5k	0.29M	2

TABLE 1. STATISTICS FOR THE DAMBREAK TEST CASE ON A REGULAR AND LOCALLY COARSENED GRIDS.



(a) Water height in reservoir (H4)



(b) Pressure in front of box (P2)

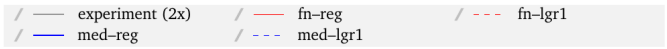


FIGURE 9.

For the three-dimensional case a similar setup procedure is followed. A lower resolution is chosen and to reduce computational time refinement zones are made smaller, which is equivalent to selecting larger coarsening zones. Only up to two levels of local coarsening were applied, since at that point the number of cells hardly reduces anymore. It was seen that above a grid resolution of $h = 5$ mm the results do not change much anymore. Note that this resolution is already 2 to 3 times higher than that of the finest grid used in the original study of 2005 [1]. As in the two-dimensional case, the results during the initial phase are only mildly affected by local coarsening. The gain in computational effort in three dimensions is much higher: 30 times less fluid cells and more than 100 times less wall-clock time; see Table 1b.

The present case study focused on accurately modeling the first impact, and the fine grid zones were chosen to comply with the interesting flow features during this initial phase. After approx. $t \approx 0.5$ s the water passes the obstacle. Then it enters coarse grid regions again, before it hits the opposite wall and the wave returns. Not surprisingly, the solutions start to deviate more, suggesting that finer grids are desirable in those regions. In fact, in almost all parts of the domain high grid resolution is desirable at some moment during the simulation. This asks for adaptive grid refinement that changes over time, a feature that will be demonstrated in the next example.

4.2 Falling life boat

As a second demonstration example, we consider the simulation of a free-fall life boat drop. Measurements were performed at MARIN, to be used for comparison. Adaptive grid

refinement is used to facilitate grid setup and reduce computational time. Currently we only consider life boat drops in still water, but ultimately it can be combined with incoming (irregular) waves (e.g. [40]). Snapshots of the simulations are shown in Fig. 10. The dynamics of the life boat is modelled by means of a 6-DOF mechanical model [41].

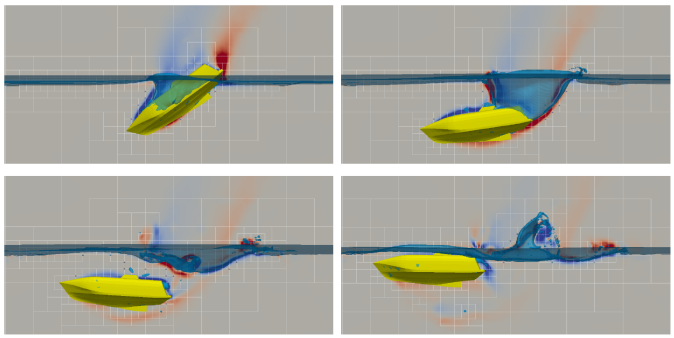
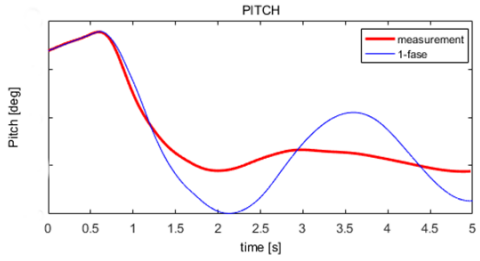


FIGURE 10. SIMULATION SNAPSHOTS OF A LIFE BOAT FALLING INTO FLAT WATER.SIMULATION OF A FREE-FALL LIFE BOAT (SNAPSHOTS ARE SORTED CLOCKWISE, STARTING FROM THE TOP-LEFT CORNER). THE GRID IS ADAPTIVELY REFINED BY MEANS OF A BASIC SURFACE AND GEOMETRY TRACKING CRITERION. EACH VISUALIZED BLOCK CORRESPONDS TO 6X6X6 GRID CELLS. COLORING REPRESENTS THE FLOW VORTICITY IN THE XZ-PLANE, CLIPPED TO THE INTERVAL [-12; 12].

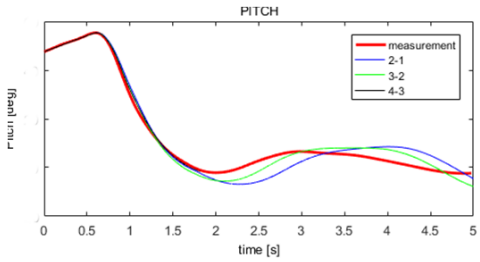
The fluid flow is modelled with the Navier–Stokes equations and solved on a series of grids, ranging from 35 k to 320 k active (i.e. within the fluid) grid points, including local grid refinement [34] around the life boat and the free surface (Fig. 10).

We present results obtained with the one- two-phase flow models and a second-order upwind convection. A comparison with experimental data reveals that the results with the one-phase model, which uses first-order upwind, are clearly inferior to those of the two-phase model; see Fig. 11. This is probably due to the entrained air bubble that forms behind the life boat upon water entry. It was concluded that two-phase flow effects play a crucial role that cannot be neglected. Furthermore, it was observed that second-order upwind discretization of convection significantly improves results. We hope to further investigate the role of turbulence modeling in the near future.

Of critical importance is the grid resolution around the object and the free surface. Various grid configurations were investigated with local resolutions ranging between 0.5 m and 0.06 m. The properties of three simulations are listed in Table 2. In the far-field, both in horizontal direction and towards the bottom of the domain, the grid resolution is coarsened down to a resolution



(a) One-phase simulation



(b) Two-phase simulation

FIGURE 11. ONE-PHASE (TOP) VERSUS TWO-PHASE (BOTTOM) RESULTS FOR THE PITCH ANGLE OF THE FALLING LIFE BOAT, AS COMPARED WITH EXPERIMENTAL DATA. THE ONE-PHASE RESULT CLEARLY MISSES THE AIR BUBBLE THAT IS ENTRAINMENT BY THE LIFE BOAT UPON WATER ENTRY.

run	resolution around surface [m]	resolution around geometry [cm]	# grid points	wall clock time [h]
2-1 / FF1000S-coarse	50.0	25.0	35k	3
3-2 / FF1000S-medium	25.0	12.5	90k	22
4-3 / FF1000S-fine	12.5	6.25	320k	199

TABLE 2. PROPERTIES OF THE NUMERICAL SIMULATIONS FOR THE FALLING LIFE BOAT. THE PHYSICAL SIMULATED TIME WAS 7 S.

of 2 m. The dependence of the results on the grid properties is investigated in a grid refinement study shown in Fig. 12.

The obtained numerical results are shown in Fig. 12, together with the measurements from experiment. Overall, a good agreement is observed between the numerical simulation and the measurements from experiment. Given the small differences between the solutions on the medium and fine grids it is expected that further grid refinement will only have minor effect on the solution. The remaining differences are most visible after 1.5 to 2 seconds, which is when the life boat is almost totally submerged and detaches from the air gap (see steps 2 and 3 in Fig. 10). It is thought that the results may improve further, by modeling more of the physical aspects at play. One could think of turbulence modeling (in particular behind the life boat), surface tension, as well as compressibility (in particular in the region of air inclu-

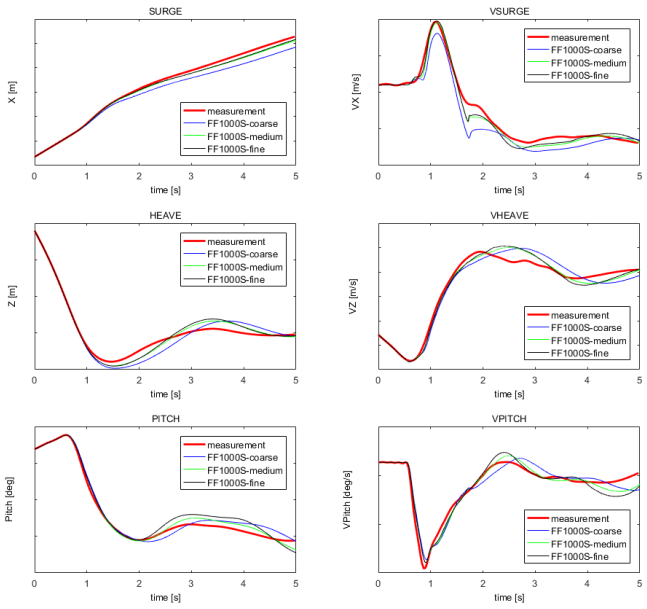


FIGURE 12. NUMERICAL RESULTS AT VARIOUS GRID RESOLUTIONS FOR THE FALLING LIFE BOAT COMPARED WITH MEASUREMENT DATA. THE LEFT-HAND COLUMN SHOWS THE POSITION OF THE C.O.G. IN THE (X,Z)-PLANE PLUS THE PITCH ANGLE. THE RIGHT-HAND COLUMN SHOWS THEIR TIME DERIVATIVES.

sion).

5 Discussion

The paper has presented a strategy for local grid refinement. Firstly, numerical discretization issues near refinement interfaces have been discussed, in order to prevent excessive reflections at those interfaces. Discrete energy conservation plays an essential criterium here. Thereafter, two examples of simulations have been presented, for which experimental validation data is available. Not surprisingly, the results show that grid coarsening in 'quit' regions in the flow does not significantly change the outcome of the simulations, wheres the computational effort is strongly reduced. Hence, local grid refinement provides an effective tool for free-surface impact problems. On the other hand, the dambreak case also shows that in more violent regions near the free surface fine grids are necessary. Thus, for this case the resolution equirements vary strongy in time, which suggests that the computational efficiency can benefit largely from adaptive refinement. We have demonstrated this in a further example of a falling life boat. Further applications of adaptive local grid refinement in free-surface problems can be found in the PhD thesis [34].

6 ACKNOWLEDGMENTS

This work is part of the research programme Maritime2013 with project number 13267 which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

[1] Kleefsman, K. M. T., Fekken, G., Veldman, A. E. P., Iwanowski, B., and Buchner, B., 2005. "A Volume-of-Fluid based simulation method for wave impact problems". *J. Comput. Phys.,* ***206***, pp. 363–393.

[2] Veldman, A. E. P., Gerrits, J., Luppes, R., Helder, J. A., and Vreeburg, J. P. B., 2007. "The numerical simulation of liquid sloshing on board spacecraft". *J. Comput. Phys.,* ***224***, pp. 82–99.

[3] Wemmenhove, R., Luppes, R., Veldman, A. E. P., and Bunnik, T., 2015. "Numerical simulation of hydrodynamic wave loading by a compressible two-phase flow method". *Computers & Fluids,* ***114***, pp. 218–231.

[4] Verstappen, R. W. C. P., and Veldman, A. E. P., 2003. "Symmetry-preserving discretization of turbulent flow". *J. Comput. Phys.,* ***187***, pp. 343–368.

[5] Hirt, C. W., and Nichols, B. D., 1981. "Volume of fluid (VOF) method for the dynamics of free boundaries". *J. Comput. Phys.,* ***39***, pp. 201–25.

[6] Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W., 1999. "An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries". *J. Comput. Phys.,* ***156***(2), pp. 209–240.

[7] Ingram, D. M., Causon, D. M., and Mingham, C. G., 2003. "Developments in Cartesian cut cell methods". *Math. Comput. Simul.,* ***61***(3-6), pp. 561–572.

[8] Dröge, M., and Verstappen, R., 2005. "A new symmetry-preserving Cartesian-grid method for computing flow past arbitrarily shaped objects". *Int. J. Num. Meth. Fluids,* ***47***, pp. 979–985.

[9] Cheny, Y., and Botella, O., 2010. "The LS-STAG method; a new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties". *J. Comput. Phys.,* ***229***, pp. 1043–1076.

[10] Rozema, W., Bae, H. J., Moin, P., and Verstappen, R., 2015. "Minimum-dissipation models for large-eddy simulation". *Phys. Fluids,* ***27***, p. 085107.

[11] Abkar, M., Bae, H. J., and Moin, P., 2016. "Minimum-dissipation scalar transport model for large-eddy simulation of turbulent flows". *Physical Review Fluids,* ***1***(4), p. 041701.

[12] Youngs, D. L., 1987. An interface tracking method for a 3d Eulerian hydrodynamics code. Technical Report AWRE/44/92/35, Atomic Weapons Research Establishment.

[13] Düz, B., 2015. "Wave generation, propagation and absorption in CFD simulations of free surface flows". PhD thesis, Technical University Delft.

[14] Arakawa, A., 1966. "Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part I". *J. Comput. Phys.,* ***1***, pp. 119–143.

[15] Minion, M., 1996. "A Projection Method for Locally Refined Grids". *J. Comput. Phys.,* ***127***(1), pp. 158–178.

[16] Liu, Q., 2011. "A stable and accurate projection method on a locally refined staggered mesh". *Int. J. Num. Meth. Fluids,* ***67***(1), pp. 74–92.

[17] Kort, A. J. A., 2016. "Discrete energy conservation in numerical simulations with local grid refinement". PhD thesis, University of Groningen.

[18] Vincent, S., and Caltagirone, J., 2000. "A one-cell local multigrid method for solving unsteady incompressible multiphase flows". *J. Comput. Phys.,* ***163***(1), pp. 172–215.

[19] Delage-Santacreu, S., Vincent, S., and Caltagirone, J., 2009. "Tracking fronts in one and two-phase incompressible flows using an adaptive mesh refinement approach". *J. Sci. Comp.,* ***41***(2), pp. 221–237.

[20] Subbareddy, P. K. and Candler, G. V., 2009. "A fully discrete, kinetic energy consistent finite-volume scheme for compressible flows". *J. Comput. Phys.,* ***228***, pp. 1347–1364.

[21] Wellens, P., 2012. "Wave simulation in truncated domains for offshore applications". PhD thesis, Delft University of Technology.

[22] Düz, B., Borsboom, M. J. A., Wellens, P. R., Veldman, A. E. P., and Huijsmans, R. H. M., 2017. "An absorbing boundary condition for free-surface water waves". *Comput. Fluids,* ***156***, pp. 562–578.

[23] Vichnevetsky, R., 1981. "Propagation through numerical mesh refinement for hyperbolic equations". *Math. Comp. Sim.,* ***23***(4), pp. 344–353.

[24] Choi, D., Brown, D., Imbiriba, B., Centrella, J., and MacNeice, P., 2004. "Interface conditions for wave propagation through mesh refinement boundaries". *J. Comput. Phys.,* ***193***(2), pp. 398–425.

[25] Long, D., and Thuburn, J., 2011. "Numerical wave propagation on non-uniform one-dimensional staggered grids". *J. Comput. Phys.,* ***230***(7), pp. 2643–2659.

[26] Manteuffel, T., and White, A., 1986. "The numerical solution of second-order boundary value problems on nonuniform meshes". *Math. Comp.,* ***47***(176), pp. 511–535.

[27] Kreiss, H.-O., Manteuffel, T., Swartz, B., Wendroff, B., and White, A., 1986. "Supra-convergent schemes on irregular grids". *Math. Comp.,* ***47***(176), pp. 537–554.

[28] Gibou, F., Fedkiw, R., Cheng, L., and Kang, M., 2002. "A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains". *J. Comput. Phys.,* ***176***(1), pp. 205–227.

[29] Min, C., Gibou, F., and Ceniceros, H., 2006. "A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids". *J. Comput. Phys.,* ***218***(1), pp. 123–140.

[30] Olshanskii, M., Terekhov, K., and Vassilevski, Y., 2013. "An octree-based solver for the incompressible Navier–Stokes equations with enhanced stability and low dissipation". *Comp. & Fluids,* ***84***, pp. 231–246.

[31] Nordström, J., Gong, J., Van der Weide, E., and Svärd, M., 2009. "A stable and conservative high order multi-block method for the compressible Navier–Stokes equations". *J. Comput. Phys.,* ***228***(24), pp. 9020–9035.

[32] Kramer, R., Pantano, C., and Pullin, D., 2009. "Nondissipative and energy-stable high-order finite-difference interface schemes for 2-D patch-refined grids". *J. Comput. Phys.,* ***228***(14), pp. 5280–5297.

[33] Hicken, J., Ham, F., Militzer, J., and Koksal, M., 2005. "A shift transformation for fully conservative methods: turbulence simulation on complex, unstructured grids". *J. Comput. Phys.,* ***208***(2), pp. 704–734.

[34] van der Plas, P., 2017. "Local grid refinement for free-surface flow simulations". PhD thesis, University of Groningen.

[35] Popinet, S., 2003. "Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries". *J. Comput. Phys.,* ***190***(2), pp. 572–600.

[36] Williams, R. D., 1991. "Performance of dynamic load balancing algorithms for unstructured mesh calculations". *Concurrency and Computation: Practice and Experience,* ***3***(5), pp. 457–481.

[37] Pilkington, J. R., and Baden, S. B., 1996. "Dynamic partitioning of non-uniform structured workloads with spacefilling curves". *IEEE Transactions on Parallel and Distributed Systems,* ***7***(3), pp. 288–300.

[38] ComFLOW website. URL `www.math.rug.nl/~veldman/comflow/comflow.html`.

[39] Fekken, G., 2004. "Numerical simulation of free-surface flow with moving objects". PhD Thesis, University of Groningen, The Netherlands.

[40] Veldman, A. E. P., Seubers, H., van der Plas, P., and Helder, J., 2017. "Free-surface flow simulations with interactively moving bodies". In Proc. 36th Int. Conf. Ocean, Offshore Arctic Eng. OMAE2017. paper OMAE2017-61175.

[41] Veldman, A. E. P., Seubers, H., van der Plas, P., Hosseini Zahraei, S. M., and Wellens, P. R., 2018. "Preventing the added mass instabiity in fluid-solid interaction for offshore applications". In Proc. 37th Int. Conf. Ocean, Offshore Arctic Eng. OMAE2018. paper OMAE2018-77308.