# A FIRST ASSESSMENT OF THE INTERDEPENDENCY OF MESH MOTION AND FREE SURFACE MODELS IN OPENFOAM REGARDING WAVE-STRUCTURE INTERACTION

**GRIET DECORTE, GEERT LOMBAERT AND JAAK MONBALIU**

KU Leuven, Department of Civil Engineering
Kasteelpark Arenberg 40, B-3001 Leuven, Belgium
e-mail: griet.decorte@kuleuven.be, web page: http://bwk.kuleuven.be/hydr

**Key words:** OpenFOAM, wave-structure interaction, mesh motion, free surface models

**Summary.** Mesh motion is of key importance in assuring adequate CFD modelling of wave-structure interaction problems, such as wave impact on floating offshore wind turbines and seakeeping of ships. Wave forcing often leads to large displacements of floating structures. As a consequence, the fluid domain boundaries need to move in order to accommodate for these wave-induced displacements. The mesh quality needs to be preserved at all times to guarantee accurate and stable results for the rigid body displacements as well as for the fluid variables.
Mesh deformation techniques, in particular algebraic mesh motion methods, have been widely used within the OpenFOAM framework during the last decade. Unfortunately, stability is easily jeopardized in case of large displacements. Large mesh deformation gives rise to computationally demanding and unstable results. Sliding meshes have been used to address this issue, but they are cumbersome for multi-degree of freedom motion. Therefore, overset methods have been implemented in recent versions of OpenFOAM. Especially, the newly implemented overset methods in the OpenFOAM branch `foam-extend`, have shown to give good results for an acceptable runtime.
Simultaneously, considerable progress has been made on the development of alternatives for algebraic volume-of-fluid methods for free surface modelling, which notoriously suffer from smearing effects. Although it seems reasonable to expect that the choice in free surface model combined with a certain mesh motion technique will have an influence on the overal result, the interdependency between mesh motion techniques and free surface modelling has not been studied yet. This paper aims at taking the first steps towards a better understanding of this mesh motion-free surface interdependency and, as such, facilitate an informed choice.

## 1 INTRODUCTION

Currently, the design of floating offshore structures strongly relies on the well-established experience from the oil and gas industry. In order to model the hydrodynamic behaviour of oil and gas rigs and related structures, the use of a potential flow approach combined with a

boundary element method is common practice in modelling flow and wave loads. Well-known examples of such hydrodynamics software packages are WAMIT and Ansys AQWA.
In the light of the world-wide energy crisis and the ocean's great renewable energy potential, the behaviour of innovative floating structures, such as floating offshore wind turbines and wave energy converters, needs to be thoroughly assessed. Due to their specific design, smaller submerged volumes and slender members, effects due to viscosity and higher-order wave loads need to be considered. Potential flow-based models are not able to include higher-order nonlinearities, generally not above second-order, and viscosity is only taken along empirically.
As computational power grows, Computational Fluid Dynamics (CFD) software, which solves the Navier-Stokes equations, presents a promising alternative for the previously mentioned potential flow-based packages. Because these codes account for both viscosity and higher-order nonlinear wave loads, they overcome the main shortcomings of the potential flow approach. One of the most promising of them is the open-source code OpenFOAM to which research groups worldwide contribute and of which different branches exist as shown in Figure 1 [1, 2]. In addition to the foam-extend branch, an extension package, `Naval Hydro Pack`, has been developed. `Naval Hydro Pack` is especially suited for simulating the viscous, two-phase, large-scale flows which naval structures frequently encounter, and the interactions involved [3].

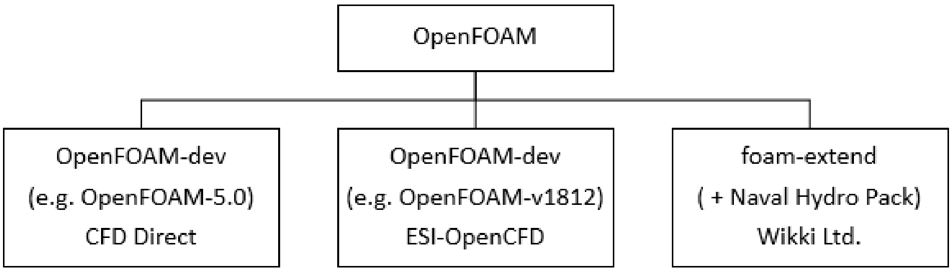


Figure 1: OpenFOAM branches.

In order for OpenFOAM to model wave-structure interaction, two strongly grid dependent challenges have to be overcome. On the one hand, the desired wave conditions can only be obtained if the free surface is accurately described. On the other hand, the resulting floater motion needs to be adequately accommodated for by the mesh guaranteeing a stable and reliable outcome. Recent research has made considerable progress on both aspects compared to the original two-phase fluid solver `interFOAM` [1, 4, 5].
Concerning the free surface modelling, two main categories of volume-of-fluid (VOF) methods are traditionally used. Within the original `interFOAM` solver, an algebraic VOF solver, `MULES`, is implemented. The algebraic VOF solver describes the fluid interface by computing the water fraction in each cell. The free surface is therefore smeared over several cells. The `MULES` solver tries to limit this smearing by adopting a (non-physical) compression term. Due to this smearing effect, an accurate solution for the algebraic VOF solver can only be found through small spatial and temporal resolutions, rendering computations costly [1, 6]. In a bid to overcome the computational limits posed by the algebraic VOF method and to arrive at a sharper surface, the geometric VOF method, `isoAdvector`, has been proposed [5]. Notwithstanding the improved

water surface modelling, this method appears not to honour its original promise in allowing accurate results at higher Courant numbers (Co) and, hence, in reducing computational cost. This is attributed to the coupling of `isoAdvector` with the pressure-velocity coupling step (`PISO` loop) in the `interFOAM` solver [7].
As wave forcing often leads to large displacements of floating structures, the fluid domain boundaries need to accommodate for these wave-induced displacements. Moreover, the mesh quality needs to be preserved at all times to guarantee accurate and stable results for the solid body displacements as well as the fluid variables. The fluid domain mesh thus needs to adapt to its deforming boundaries. Because these displacements of the fluid-structure interfaces are, especially in the case of six-degrees-of-freedom (6-DOF) rigid body motion, a priori unknown and as such the result of the solution itself, this mesh motion is ideally done automatically.
Although several of the currently available automatic mesh motion techniques have shown to do to a proper job for a variety of naval applications [8, 9], the interdependency between these mesh motion techniques and the free surface description adopted has not been formally identified. This paper wants to take the first steps towards a better understanding of this interdependency of mesh motion technique and free surface description, and their combined effects on the overall model result, to facilitate an informed choice.
The paper is structured as follows. First, an overview of the mesh motion techniques implemented in OpenFOAM is given. Next, the research method is explained and, subsequently, the results are presented and discussed. Finally, conclusions concerning performance and further work are drawn.

## 2 OVERVIEW OF THE MESH MOTION TECHNIQUES IN OPENFOAM

In the literature, an exhaustive body of knowledge concerning (automatic) mesh handling techniques already exists. Because this paper attempts to merely assess the performance of only two of these, the algebraic mesh motion and the overset method, we will only give a brief overview of the methods currently implemented within the OpenFOAM framework. In OpenFOAM, the mesh motion techniques can be roughly divided into four main categories; algebraic mesh motion techniques, topological changes, immersed boundary methods and overset methods. An overview is shown in Figure 2.

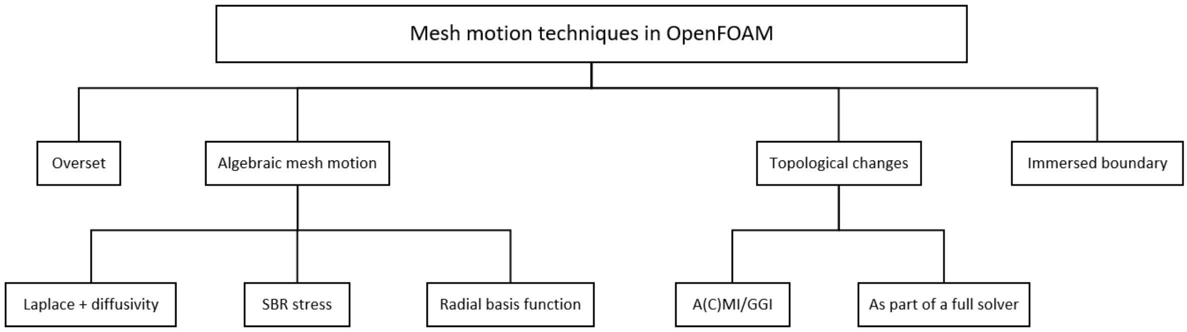


Figure 2: Overview of mesh motion techniques in OpenFOAM.

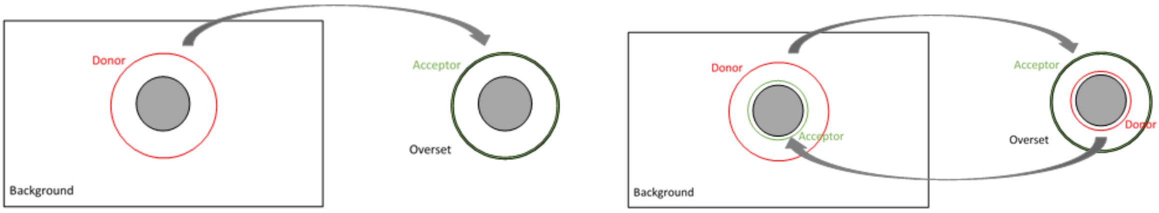
The algebraic mesh motion techniques adopt an Arbitrary Lagrangian-Eulerian (ALE) form of the conservation equations combined with a Space Conservation Law which prevents from creating an extra source term and thus from generating large errors [10, 11]. ALE necessitates that an automatic mesh displacement prescription algorithm be supplied to adapt the (fluid) grid velocities to the problem under consideration. Ideally, this is done through potential type equations or interpolations. Often, a Laplacian is solved with a certain diffusivity, allowing to distribute the mesh deformation over a certain distance from the moving boundary. Because of its high accuracy and its discretisation error being comparable to static meshes, this method is the preferred option for small displacements. Unfortunately, in case of large boundary deformation, the mesh deteriorates, giving rise to stability problems and loss of accuracy [12]. Two of the most promising ways to tackle this are the radial basis function technique (RBF) which allows for larger displacements but ultimately leads to distorted meshes as well [13], and topological changes.

Topological changes are used to change the mesh resolution and connectivity, using sliding interfaces, cell layering and similar techniques. They involve data mapping, which gives rise to distribution and conservation errors [12]. Ideally these operations are bundled. Examples of topological changes can be found in [9, 14]. Because applying topological changes for 6-DOF rigid body motion becomes quite tedious, the immersed boundary method and the overset methods might be a more viable candidates to accommodate the large displacements.

In the immersed boundary methods, a Dirac delta function is introduced as a body force in the conservation equations. This body force only differs from one at the fluid-structure interface. Because the immersed boundary method does not allow for mesh refinement close to the structure and, thus, does not allow for accurate boundary layer simulation, it is often not well-suited for the naval applications and will therefore be disregarded in the following [15]. However, it is worth to note that this method has been included in the `Naval Hydro Pack` [4].

Overset methods are by far the most versatile and computationally efficient techniques when confronted with large mesh motion. In this method, two meshes are created initially. A fixed mesh is used for the background, while the overset mesh which is allowed to move relatively to the background mesh, is connected to the moving object. Two fundamentally different implementations of overset are currently available within the OpenFOAM framework. One in the OpenFOAM-dev line, from version `v1706` on, and another one in the foam-extend branch, starting from version `4.1`. The main differences between both implementations are related to their position and role within the `PIMPLE` algorithm, which solves the momentum equations while accommodating for mass conservation in the pressure-velocity coupling step (`PISO`), and the amount of interpolations involved. Before reading on, the reader not familiar with the `interFoam` solver lay-out is referred to Figure 8 in Appendix A.

In `OpenFOAM-v1712`, at the beginning of each iteration, the motion of the structure is determined based on the pressure field computed on the overset mesh in the previous iteration. Subsequently, the overset mesh is moved accordingly and the hole cells, which are the background cells overlapped by the moving object which are excluded from the computations, the interpolated cells, which are the cells of the background overlapped by the overset mesh, and the calculated cells, which are the remaining background cells, are determined (see Figure 3a). Next, the water fraction and the velocity field are computed on both meshes and the velocity field is interpolated onto the border of the overset mesh. Finally, the velocity field is corrected for both meshes in the PISO loop and the resulting pressure field is computed, before starting a new round of fluid-structure interaction computations in the next iteration [1]. Although this implementation has certainly merits, the interpolation is not well parallelized and therefore highly time-consuming. Inadequate interpolation may introduce large mapping errors into the computations necessitating high spatial resolution and therefore an even larger computational cost.



(a) `OpenFOAM-v1712`   (b) `foam-extend-4.1`

,

Figure 3: Donor-acceptor assembly for `OpenFOAM-v1712` and `foam-extend-4.1`.

Contrary to the `OpenFOAM-v1712` implementation, `foam-extend-4.1` adopts a two-way coupling between the background and the overset mesh as shown in Figure 3b. At the beginning of each `PIMPLE` iteration, the motion of the structure is updated based on the pressure field computed during the previous iteration. Next, the fringe layers, i.e. donor-acceptor pairs, are created, either manually or through adaptive overlap, and the hole cells are determined. Subsequently, the water fraction is calculated and interpolated in the fringe layers, and a velocity estimate is computed. In the PISO loop, the interpolated velocity field is then used as "boundary condition" to correct the velocity fields and compute the pressure field on both meshes. The velocity field as well as the pressure field is interpolated and the next iteration starts [8]. This two-way coupling presents the main difference between the `OpenFOAM-v1712` and the `foam-extend-4.1` implementation. In addition, the overset method in `foam-extend-4.1` is well parallelized, which means substantial computational time can be saved.

## 3 METHODOLOGY

Because the choice of a mesh motion technique for a specific application depends to some extent on the chosen free surface model, the consequences of such combinations need to be known. Therefore, in the following, the two most commonly used mesh motion techniques, algebraic mesh motion and overset, will be both combined with the traditional algebraic VOF method, `MULES`, and the innovative geometric VOF method, `isoAdvector`. For each combination, visual assessment will be done through comparison with a reference case, the freely heaving cylinder case documented by Ito (1977). In these experimental tests, a rigid horizontal cylinder with a diameter of 0.1524m (0.5 ft.) is released from an initial height of 0.0254m (1 in.) above a calm water surface and left to freely decay its heave motion. The depth of the wave flume is 1.22m (4 ft.) and its original length is 27m (90 ft.)[16].
As algebraic mesh motion eliminates the influence of any inadequate coupling between meshes, it is used to assess grid convergence and is therefore treated first. A set-up with `MULES` and

isoAdvector is run for three mesh sizes each chosen relative to the cylinder diameter $D$, notably $D/10$, $D/20$ and $D/40$. As the interDyMFoam solver, an interFoam solver allowing for dynamic mesh motion, is quite similar in OpenFOAM-v1712 and foam-extend-4.1/Naval Hydro Pack and isoAdvector has already been added to the interDyMFoam solver in Naval Hydro Pack, navalFoam, a loosely coupled 6-DOF rigid body motion solver in navalFoam is adopted. Even though OpenFOAM-v1712 and foam-extend-4.1 both offer a lot of freedom through the availability of different solvers for solving the motion equations of the floater, only the extension package for foam-extend-4.1 offers anything besides loosely coupled fluid-structure interaction, for instance Aitken relaxation and multiple loops enabling strongly coupled fluid-structure computations [4]. In the following, only Aitken relaxation is used. As both free surface modelling approaches have shown to give best results for Co smaller than 0.1, an adaptive time step based on a maximum Co of 0.1 is used [6, 7].

Next, the reference case presented above is computed for the overset approach. Due to the significantly different implementation of the overset methods in OpenFOAM-v1712 and foam-extend-4.1 /Naval Hydro Pack, the interdependency between the two free surface models and the overset approach are studied for both branches. As the 6-DOF rigid body motion is currently not implemented in the overset solver of the publicly released foam-extend-4.1, the navalOversetFoam solver in Naval Hydro Pack is used for the foam-extend-4.1 case and compared to its counterpart, overInterDyMFoam, in OpenFOAM-v1712. Both cases are run with inverseDistance interpolation for the donor-acceptor pairs. For navalOversetFoam, implicit overset is used to guarantee strong coupling between the meshes and prevent using many corrector loops to reach a sufficiently strong coupling as would be the case if explicit overset mesh had been chosen. For a thorough overview of the overset mesh coupling and interpolations strategies in foam-extend-4.1/Naval Hydro Pack, the reader is referred to the existing literature [4, 8]. In choosing time and spatial resolution, a trade-off needs to be made between accuracy and computational cost. This is done based on the algebraic mesh motion cases.

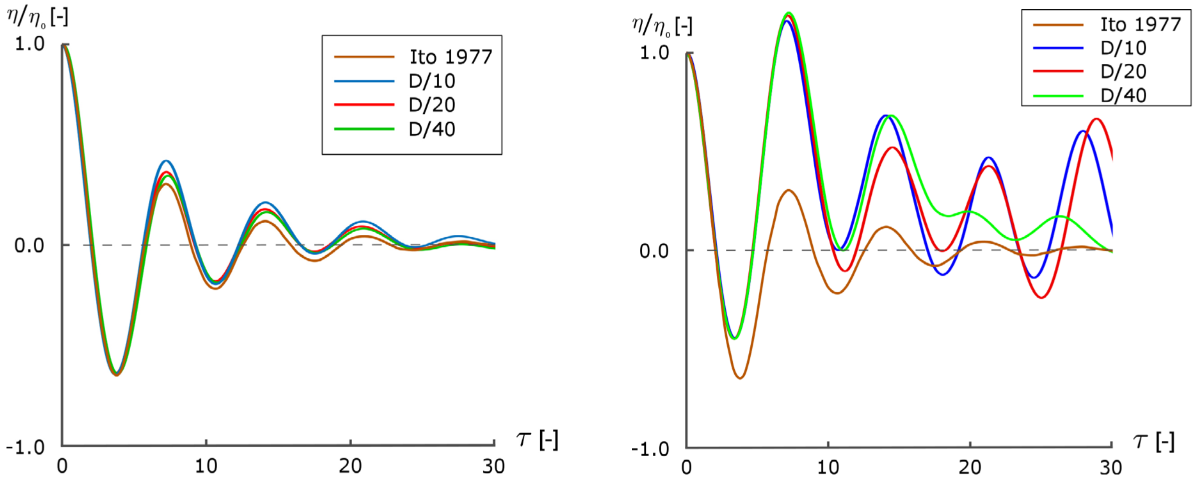
Although in the Naval Hydro Pack isoAdvector has been added to the navalFoam solver, which allows algebraic mesh motion to be applied in combination with isoAdvector, none of this has been done yet in OpenFOAM-v1712, nor for the navalOversetFoam solver in Naval Hydro Pack. Therefore, we have extended the overset solvers in OpenFOAM-v1712 and Naval Hydro Pack to allow for isoAdvector free surface modelling. However, it has to be noted that this coupling was only done as a first assessment. As will be shown in the next section, in most cases, enhancements should be made in order to arrive at a robust solver.

## 4 RESULTS AND DISCUSSION

To provide a first assessment of the interdependency between the two most commonly used mesh motion techniques, algebraic mesh motion and overset, and the free surface model, each mesh motion technique has been run with both the MULES approach and the isoAdvector method. In the following, first, the interdependency between the algebraic mesh motion and the free surface model, is discussed. Next, the same is done for the overset approach.

### 4.1 Algebraic mesh motion

As shown in Figure 4a, the MULES approach models the free decay well. The solution converges and, even for coarse meshes, the results collide well with the reference solution by Ito(1977). However, when isoAdvector is used to model the free surface, only the first time steps are modelled well. As can be seen in Figure 4b, the floating cylinder performs an oscillation with a frequency close to its eigenfrequency, but it does not execute a regular decay.



(a) MULES

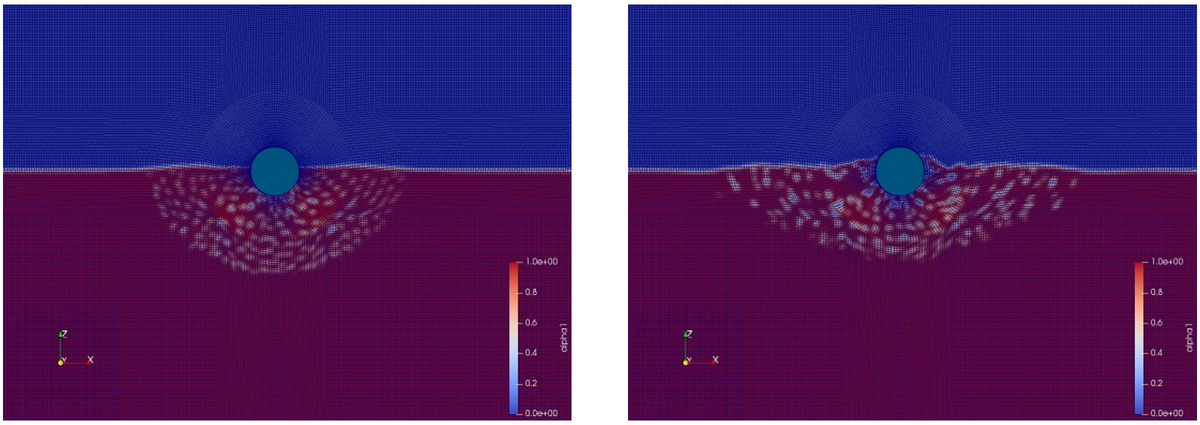
(b) isoAdvector

,

Figure 4: Non-dimensional free decay heave motion $\eta/\eta_0$, with $\eta$ the absolute heave and $\eta_0$ the initial displacement from rest, vs. non-dimensional time $\tau = t\sqrt{g/(D/2)}$, with $g$ the gravitational acceleration, for algebraic mesh motion combined with (a) MULES and (b) isoAdvector free surface modelling for three grid resolutions.

When analysing the water phase of the isoAdvector solution in Figure 5, it becomes clear that the free surface modelling may be at the origin of the unsatisfactory free decay simulation. Small volumes of low water fraction appear (Figure 5a), which subsequently lead to non-physical air bubbles in the water volume. Due to their lower density, these bubbles rise to disturb the free surface and as such the free decay motion (Figure 5b). The reason for this most probably lies in the way isoAdvector is currently implemented within the navalFoam solver.

Starting from the first time step, the mesh is deformed and subsequently the mass flux and the velocity fields are updated. However, although the volume and the shape of the cells have changed, the water fraction scalar field remains the one calculated in the previous PIMPLE iteration. In calculating the water fraction field in the isoAdvector, first, the change in volume over each cell-to-cell face is determined. To do so, the updated fluxes are multiplied by the water fraction field from the previous iteration. As the cell volumes have now changed due to the mesh deformation, the resulting change in volume over each cell-to-cell face is incorrect. For each fully immersed cell, the new water fraction is subsequently determined by subtracting from the water fraction, from the previous iteration, the total change in volume integrated over its



(a) $\tau = 4.5$

(b) $\tau = 11.3$

,

Figure 5: Water fractions for the algebraic mesh motion with `isoAdvector` at non-dimensionalized times $\tau = 4.5$ (left) and $\tau = 11.3$ (right).

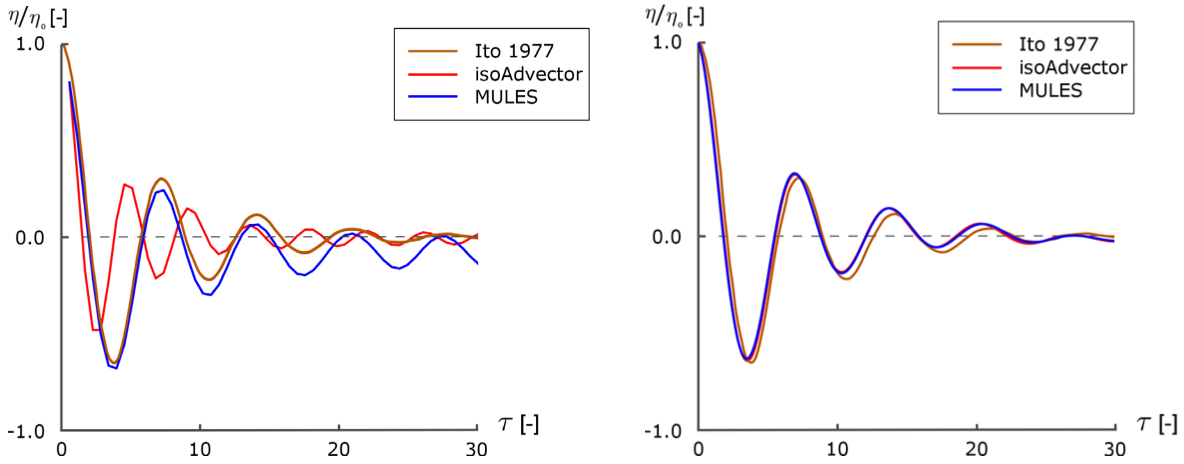
faces and divided by the new volume of that cell. This introduces an additional error into the resulting water fraction. When progressing further in time, this error will accumulate and, at some point, the erroneously computed water fractions of the fully immersed cells will reach the threshold for `isoAdvector` to consider them as partly filled with air. This leads then to the formation of non-physical air bubbles, which is clearly seen in Figure 5. For full details on the `isoAdvector` algorithm, the reader is kindly referred to the literature [5].
It has to be noted that `MULES` suffers slightly from this issue as well. In the water volume where the mesh deforms, smaller water fractions could also be observed. However, this was easily resolved by increasing the iterations over the `MULES` computation. The same reasoning was used in trying to eliminate the air bubbles for the `isoAdvector` algorithm. Although, their severity diminished, it did not seem possible to fully resolve the air bubbles by increasing the number of iterations, i.e. the correctors, in the `PIMPLE` loop.

## 4.2 Overset mesh motion

Based on the results for the algebraic mesh motion in Figure 4, a trade-off was made between accuracy and computational cost to determine time and spatial resolutions for the overset models. A grid resolution of $D/20$ and an adaptive time step according to a Co of 0.2 was chosen. In the following, first, the overset method in `OpenFOAM-v1712` is discussed and, next, its version in `foam-extend`.

As can be noticed in Figure 7, `MULES` aligns reasonably well with the experimental result. On the contrary, the `isoAdvector` solution decays, but at a frequency smaller than its eigenfrequency. In addition, when released from rest, the cylinder starts its decay, but does not reach the expected depth. This can most probably be attributed to one of the main shortcomings of the current overset implementation in `OpenFOAM-v1712` and seems as such not to be related to the overset mesh coupling.

`OpenFOAM-v1712` computes its water fraction (and pressure) for the hole cells, i.e. the cells



(a) `OpenFOAM-v1712` (b) `Naval Hydro Pack`

Figure 6: Non-dimensional free decay heave motion $\eta/\eta_0$, with $\eta$ the absolute heave and $\eta_0$ the initial displacement from rest, vs. non-dimensional time $\tau = t\sqrt{g/(D/2)}$, with $g$ the gravitational acceleration, for the overset method combined with MULES and isoAdvector for (a) `OpenFOAM-v1712` and (b) `Naval Hydro Pack` (`foam-extend-4.1`).

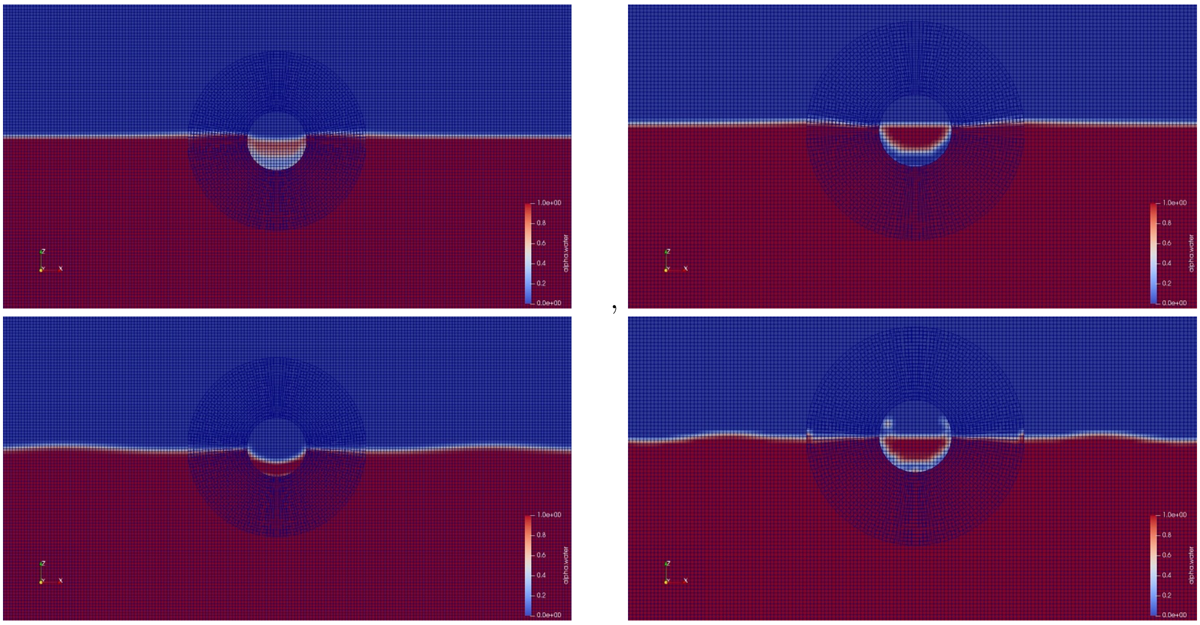
covered by the cylinder on the background grid, and assumes as such that there is water where there is in fact not. Due to the diffusive nature of `MULES`, no real air phase is formed in the hole as shown in Figure 7a, which lead to moderate negative pressures on the inside of the cylinder and can explain the relatively small error at first. However, in the case where `isoAdvector` is used in Figure 7b, a separate air phase and water phase arise inside the cylinder. This reduces the buoyancy and as such the eigenfrequency in heave. Additionally, as the air phase is enclosed under the water, suction seems to arise as the cylinder moves down. This results in a higher upward force on the cylinder, which might be the reason for the reduced depth.

On the opposite, as can be noticed in 6b, the `navalOverFoam` solver seems to perform equally well for both the `MULES` and the `isoAdvector` approach. When setting up a case in `foam-extend-4.1/Naval Hydro Pack`, the initial field values for the hole are set to zero and the hole cells are fully excluded from further computations. This effectively prevents the separate water volume from arising in the cylinder.

## 5 CONCLUSIONS

The aim of this paper was to take the first steps towards a better understanding of the interdependency of mesh motion methods and free surface models and, as such, facilitate an informed choice. Therefore, the two most commonly used mesh motion techniques, algebraic mesh motion and overset, were both combined with the traditional algebraic VOF method, `MULES`, and the innovative geometric VOF method, `isoAdvector`. For each combination, a visual assessment was done through comparison with a freely heaving cylinder reference case.

Concerning algebraic mesh motion, the solution for `MULES` converged towards the reference solution, while, in case of `isoAdvector`, the cylinder heaves in an unorganized way. The air bubbles



(a) MULES

(b) isoAdvector

,

Figure 7: Water fractions for the overset method in OpenFOAM-v1712 at $\tau = 3$ (up) and $\tau = 14$ (down). Left: MULES. Right: isoAdvector.

arising in the water volume and eventually disturbing the free surface are the most probable culprits. They arise due to the fact that isoAdvector uses the water fraction from the previous iteration, i.e. before the mesh was deformed. In order to overcome this issue, two ways forward seem practical. On the one hand, the water fraction could be corrected after mesh deformation. On the other hand, a conditional statement could be used to prevent these air bubbles from arising in the water volume.

Furthermore, the overset method in foam-extend-4.1/Naval Hydro Pack appeared to perform well for both the MULES and isoAdvector simulation. On the contrary, no similar observations could be made for the OpenFOAM-v1712 implementation. Although the overset in OpenFOAM-v1712 seemed to perform reasonably well for the MULES case, this appears not to be so for the isoAdvector simulation. This may have been caused by the lingering issue of the hole cells being included in most of the OpenFOAM-v1712 overset solver's fluid computations. Future releases of OpenFOAM need to tackle this issue in order to allow for robust overset mesh simulations.

Overall, all mesh motion solvers perform well with MULES. In addition, the overset methods show the most potential when combined with isoAdvector, while significant adaptions are needed to the algebraic mesh motion solvers in order for them to be combined with the isoAdvector free surface modelling.

## ACKNOWLEDGEMENTS

The first author is Ph.D. fellow of the Research Foundation Flanders (FWO), Belgium (Ph.D. fellowship 11A1217N). Furthermore, the authors gratefully acknowledge the collaboration with the research group of prof. Hrvoje Jasak at UniZagreb and the chance to use their extension package for `foam-extend-4.1`, `Naval Hydro Pack`, in their research.

## A General overview of the `interDyMFoam` and `overInterDyMFoam` solvers in OpenFOAM.

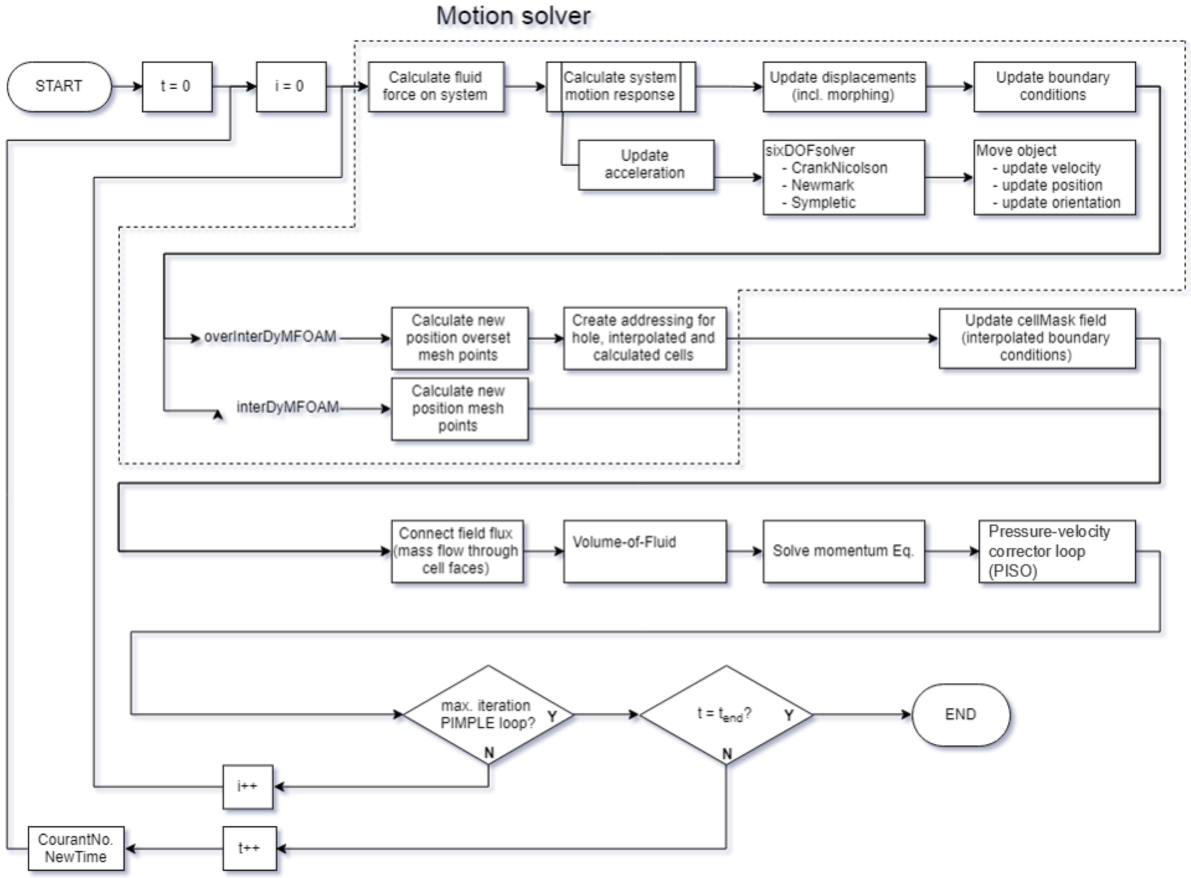

Motion solver
START
t = 0
i = 0
Calculate fluid force on system
Calculate system motion response
Update displacements (incl. morphing)
Update boundary conditions
Update acceleration
sixDOFsolver
- CrankNicolson
- Newmark
- Sympletic
Move object
- update velocity
- update position
- update orientation
overInterDyMFOAM
Calculate new position overset mesh points
Create addressing for hole, interpolated and calculated cells
Update cellMask field (interpolated boundary conditions)
interDyMFOAM
Calculate new position mesh points
Connect field flux (mass flow through cell faces)
Volume-of-Fluid
Solve momentum Eq.
Pressure-velocity corrector loop (PISO)
max. iteration PIMPLE loop?
Y
N
t = t_end?
Y
N
END
i++
t++
CourantNo. NewTime

Figure 8: The program structure of the two-phase fluid-structure motion solvers, `interDyMFoam` (includes algebraic mesh motion and topological changes) and `overInterDyMFoam` (includes overset mesh motion).

## REFERENCES

[1] OpenCFD. *OpenFOAM user guide.* (2008) http://www.openfoam.org.

[2] Weller, H., Tabor, G., Jasak, H. and Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics.* (1998) **12**:620-631.

[3] Jasak, H. OpenFOAM: open source CFD in research and industry. *International Journal of Naval Architecture and Ocean Engineering.* (2009) **1**(2):89-94.

[4] Vukčević, V., Gatin, I. and Jasak, H. The Naval Hydro Pack: current status and challenges. *In: 13th OpenFOAM Workshop.* (2018)

[5] Røenby, J., Bredmose, H. and Jasak, H. A computational method for sharp interface advection. *Open Science.* (2016) **3**(11).

[6] Larsen, B.E., Fuhrman, D.R., and Røenby, J. Performance of interFoam on the simulation of progressive waves. Preprint, ArXiv:1804.01158. (2018).

[7] Røenby, J., Larsen, B.E., Bredmose, H. and Jasak, H. A new Volume-Of-Fluid Method in OpenFOAM. *In: MARINE 2017 Computational Methods in Marine Engineering VII.* (2017) pp. 266-278.

[8] Vukčević, V. and Jasak, H. Overset Mesh Library in Foam-Extend [Powerpoint slides]. (2018) Retrieved from https://foam-extend.fsb.hr/wp-content/uploads/2018/07/OversetTrainingSlides.pdf.

[9] Devolder, B. and Stratigaki, V. and Troch, P. and Rauwoens, P. CFD Simulations of Floating Point Absorber Wave Energy Converter Arrays Subjected to Regular Waves. *Energies.* (2018) **11**(3):641.

[10] Demirdžić, I. and Perić, M. Space conservation law in finite volume calculations of fluid flow. *International Journal of Numerical Methods in Fluids.* (1988) **8**(9):1037-1050.

[11] Demirdžić, I. and Perić, M. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal of Numerical Methods in Fluids.* (1990) **10**(7):771-790.

[12] Jasak, H. Dynamic mesh handling in openfoam. *In: 47th AIAA Aerospace Sciences.* (2009) pp. 341-350.

[13] Bos, F.M. *Numerical simulations of flapping foil and wing aerodynamics.* PhD Thesis, TU Delft (2009).

[14] Liu, Y., Xiao, Q., Incecik, A., Peyrard, C. and Wan, D. Establishing a fully coupled CFD analysis tool for floating offshore wind turbines. *Renewable Energy.* (2017) **112**: 280-301.

[15] Dooms, D. *Fluid-structure interaction applied to flexible silo constructions.* PhD Thesis, KU Leuven (2009).

[16] Ito, S. *Study of the transient heave oscillation of a floating cylinder.* MSc Thesis, Massachusetts Institute of Technology (1977).