Partitioned solution to fluid-structure interaction problems in application to free-surface flows

Christophe Kassiotis, Adnan Ibrahimbegovic, Hermann G. Matthies

Partitioned solution to fluid-structure interaction problem in application to free-surface flows

C. Kassiotis[3], A. Ibrahimbegovic[1], H. Matthies[2]

July 12, 2010

[1] LMT-Cachan (ENS Cachan/CNRS/UPMC/PRES UniverSud Paris)
61 avenue du Président Wilson, F-94230 Cachan, France

[2] Institut für Wissenschaftliches Rechnen (TU-Braunschweig)
D-38092 Braunschweig

[3] Saint-Venant Laboratory for Hydraulics, Université Paris-Est (Joint Research Unit EDF R&D, CETMEF, École des Ponts ParisTech)
6 quai Watier, BP 49, 78401 Chatou, France
christophe.kassiotis@enpc.fr

Abstract

In this work we discuss a way to compute the impact of free-surface flow on nonlinear structures. The approach chosen rely on a partitioned strategy that allows to solve strongly coupled fluid-structure interaction problem. It is then possible to re-use existing and validated strategy for each sub-problem. The structure is formulated in a Lagrangian way and solved by the finite element method. The free-surface flow approach considers a Volume-Of-Fluid (VOF) strategy formulated in an Arbitrary Lagrangian-Eulerian (ALE) framework, and the finite volume are used to discrete and solve this problem. The software coupling is ensured in an efficient way using the Communication Template Library (CTL). Numerical examples presented herein concern 2D validations case but also 3D problems with a large number of equations to be solved.

1 Introduction

In this work we focus on fluid-structure interaction problems, with free-surface flows. The kind of application motivating this development stems from the coastal engineering problems, chief among them dam-breaking problem and tsunami (*e.g.* see [29]). The main difficulty for this class of problems is not only the usual one for fluids pertaining to the free-surface flow (*e.g.* see [22, 42, 33, 5, 72, 70, 71]) but also the fluid-structure interface. Providing reliable description at the interface for a fluid-structure interaction problem is quite a challenge, since the descriptions generally used for each of the sub-problems are different: for the structure part, it is natural to follow material point motion in a Lagrangian formulation, while an Eulerian formulation is often preferred

for the fluid part. Naturally, there are bridges between these two formulations, which have to be derived carefully for fully nonlinear setting.

Moreover, the optimal discretization technics for each sub-problem are in general not the same. For the solid part, the method widely used since the sixties is the Finite Element Method [81, 41], even if it is possible to solve the Partial Differential Equations that describe the continuum problem either by other discretization techniques such as Finite Volume [69]. For the fluid part, two different methods are mainly used: either the Finite Element Method [80, 9], or the Finite Volume Method [28]. The first one has the advantages of a complete mathematical framework and well established convergence properties, but requires special care to stabilize the solution when incompressible flow are aimed at [32]. For the second one, the mathematical framework is slightly more difficult to complete but the more physical foundation allows to naturally get good conservative properties. There are bridges that combine advantages of both formulations [62]. In this work, we choose the Finite Element Method for structure discretization and the Finite Volume Method for fluid discretization. This choice provides the largest selection of existing software products to be used for the software coupling strategy we suggest herein.

For the fluid-structure interaction framework, it is required to consider flow problem in a moving domain, as the fluid domain shares at least one of its boundaries with the structure that is undergoing deformation. It is possible to solve this kind of problem with for instance the fictitious domain approach [78] or with an Arbitrary Lagrangian-Eulerian approach [17, 37]. The latter requires to solve the deformations of the fluid underlying grid, that is often not able to support very large deformation without re-meshing; nevertheless it is often the one chosen in existing tools to solve fluid problems in moving domain, and was thus also used herein. The impossibility to represent too large deformations without re-meshing makes the choice of another method necessary when complex fluid domain deformations such as the ones observed for the sloshing of a wave are represented. Here a two-phase approach is chosen where both air and water are represented, and a characteristic function is used to distinguish the two phases. To solve this problem, a Volume-Of-Fluid strategy is considered herein.

For coupled fluid-structure interaction problems, the monolithic [38, 65, 20] and partitioned strategy [24, 53, 59, 11, 27, 58, 18, 31, 26, 77] can be used. The monolithic approach is abandoned in favor of the partitioned approach. The latter is preferred for its modularity and the possibility of re-using existing software. The partitioned approach used here is based on the Direct Force-Motion Transfer. Both explicit and implicit coupling algorithms for multi-physics problems are detailed. An explicit strategy leads to the so-called "Added Mass effect", and for that justifies the use of more costly implicit solvers.

Implicit solvers can use either a fixed-point strategy that is known to be rather slow to converge, or the Newton strategy that requires building up and evaluating the costly Jacobian. In this work, the fixed-point strategy based on the Block Gauss-Seidel algorithm (DFMT-BGS) with an adaptive relaxation parameter [51] shows sufficient performances for the example proposed. The properties as well as stability of the implicit coupling DFMT-BGS algorithms used herein are presented in detail in [48].

In this work, a general fluid-structure interaction framework based on existing software was used. This framework was built using the middleware Communication Template Library (CTL) [61] which offers good performances, and can

therefore be used for scientific computing of large systems. An important feature is the possibility to couple types of software product that were initially not programmed to be coupled (here `FEAP` for the structure and `OpenFOAM` for the fluid), even if they are based on different discretization techniques (respectively FV for the fluid and FE for the structure) and were programmed in different languages, `C++` and `Fortran`. For more details on the implementation, see [49]

The outline of this paper is as follow: in the subsequent section we present the chosen formulations for the structure and fluid sub-problem. In particular, the fluid formulation is given in Arbitrary-Lagrangian Eulerian framework, along with the modifications to account for the free-surface flow. In Section 4, we describe the coupling between the fluid and the structure sub-problems. In Section 5, we give and comment the results of illustrative numerical examples dealing with free-surface flow impacting a structure in two and three-dimensions as well as comparison with existing works. The concluding remarks are given in the last section.

2 Description of the structure

2.1 Structure equations of motion

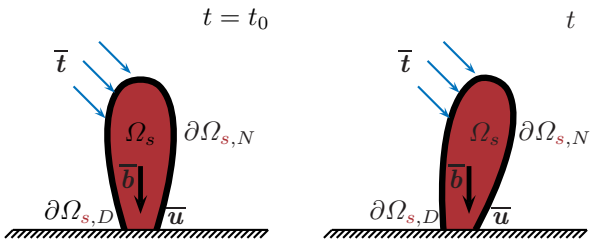


Figure 1: Solid problem

The structure motion is based on a Lagrangian description. Namely, we consider a structure domain Ω_s, with imposed displacements $\overline{\boldsymbol{u}}$ on its Dirichlet boundary $\partial\Omega_{s,D}$, moving under the traction forces $\overline{\boldsymbol{t}}$ imposed on its Neumann boundary $\partial\Omega_{s,N}$ and a body load $\overline{\boldsymbol{b}}$ applied in the whole domain Ω_s (see Fig. 1). The motion is dynamic and evolves in time on the segment $[0, T]$. The governing equation for a structure describes the momentum conservation and is also known as the Cauchy equation. The strong form of the structural problem written with respect to the deformed configuration is as follows:
Given: $\overline{\boldsymbol{u}}$ on $\partial\Omega_{s,D} \times [0,T]$, $\overline{\boldsymbol{t}}$ on $\partial\Omega_{s,N} \times [0,T]$ and $\overline{\boldsymbol{b}}$ in $\Omega_s \times [0,T]$.
Find: $\boldsymbol{u} \in \Omega_s \times [0,T]$ so that:

$$\nabla \cdot \boldsymbol{\sigma} + \rho_s \left(\overline{\boldsymbol{b}} - \partial_t^2 \boldsymbol{u} \right) = \boldsymbol{0} \quad \text{in } \Omega_s \times [0,T] \tag{1}$$

where ρ_s denotes the material density of the solid domain, $\boldsymbol{u}$ its displacement field and $\partial_t^2 \boldsymbol{u}$ the accelerations. The Cauchy stress tensor $\boldsymbol{\sigma}$ in the deformed configuration can be linked to the second Piola-Kirchhoff stress tensor $\mathbf{P}$ formulated in the initial configuration through the gradient $\mathbf{F}$ of the deformation and its Jacobian (see [41]).

To close this Partial Differential Equations system one needs to link the displacements (or one of its derived field) and the stresses together with constitutive behavior law. An elastic material model based on St.-Venant-Kirchhoff constitutive equation or neo-Hookean constitutive equation is assumed in the two presented example to link the stress Cauchy tensor $\boldsymbol{\sigma}$ and the Green-Lagrange strain tensor $\mathbf{E}$:

$$\mathbf{F}^{-1}\mathbf{J}\boldsymbol{\sigma}\mathbf{F}^{-T} = \mathcal{C} : \mathbf{E} \tag{2}$$

where $\mathcal{C}$ denotes the constitutive fourth-order elasticity tensor[1]. It is also possible to use more complex behavior, such as non-linear plasticity, but this is not explored herein. The non-linearity of the problem here comes from the large displacement described by the following relation between Green-Lagrange tensor $\mathbf{E}$ and the deformation gradient $\mathbf{F} = \mathbf{I} + \nabla \boldsymbol{u}$:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T\mathbf{F} - \mathbf{I}\right) \tag{3}$$

2.2 Structure discretization using Finite Element Method

It is *a priori* impossible to find directly the exact solution to the problem defined above. The idea is to find the best approximation of the solution in a space where the solution can be found numerically. The FEM formulations [8, 82, 41] derived from the equilibrium equation (1) rely on the associated weak forms of this problem:
Given: $\overline{\boldsymbol{t}}$ on $\partial\Omega_{s,N} \times [0,T]$ and $\overline{\boldsymbol{b}}$ in $\Omega_s \times [0,T]$.
Find $\boldsymbol{u} \in \mathcal{U}$ such that, for all $\delta\boldsymbol{u} \in \mathcal{U}_0$:

$$\begin{aligned} \mathcal{G}_s(\boldsymbol{u};\delta\boldsymbol{u}) &:= \int_{\Omega_s} \rho_s \partial_t^2 \boldsymbol{u}\cdot\delta\boldsymbol{u} + \int_{\Omega_s} \boldsymbol{\sigma} : \nabla\delta\boldsymbol{u} - \int_{\Omega_s} \overline{\boldsymbol{b}}\cdot\delta\boldsymbol{u} - \int_{\partial\Omega_s} \overline{\boldsymbol{t}}\cdot\delta\boldsymbol{u} \\ &= 0 \end{aligned}$$

where $\mathcal{U}$ and $\mathcal{U}_0$ are functional spaces for the solution and its variation.

The solid domain Ω_s is then discretized in a finite number of elements $\mathcal{T}_h = (\kappa_e)_{e=1,\dots,n_{el}}$ so that the whole space is covered by the finite elements that do not intersect. The research space associated with the solution is restrained to the space of continuous piecewise polynomial functions defined in each finite element domain. This space is denoted:

$$\mathcal{U}^h = \mathcal{U} \cap \left\{ \boldsymbol{u} \in \mathcal{C}^0\left(\Omega_s\right) \middle| \boldsymbol{u}|_\kappa \in \mathcal{P}^p\left(\kappa\right), \forall \kappa \in \mathcal{T}_h \right\} \tag{4}$$

where $\mathcal{P}^p\left(\kappa\right)$ is the space of polynomials of order p on κ. The same restriction holds on the associated vector space. The FE problem is defined as:
Given: $\overline{\boldsymbol{t}}$ on $\partial\Omega_{s,N} \times [0,T]$ and $\overline{\boldsymbol{b}}$ in $\Omega_s \times [0,T]$.
Find $\mathbf{u} \in \mathcal{U}^h$ such that, for all $\delta\mathbf{u} \in \mathcal{U}_0^h$:

$$\mathcal{G}_s(\mathbf{u};\delta\mathbf{u}) = 0 \tag{5}$$

This semi-discrete problem can be written in a matrix form using the real valued vectors $\mathbf{u} \in \mathbb{R}^{n_{d-o-f}}$:

$$\mathcal{R}_s(\mathbf{u}_s;\boldsymbol{\lambda}) := \mathbf{M}_s\ddot{\mathbf{u}}_s + \mathbf{f}_s^{\text{int}}\left(\mathbf{u}_s\right) - \mathbf{f}_s^{\text{ext}}\left(\boldsymbol{\lambda}\right) = \mathbf{0} \tag{6}$$

[1] We note in passing that the formulation developed herein for fluid-structure interaction problem would also apply to more elaborate inelastic constitutive models (see [41])

with $\mathbf{M}$ the mass matrix, $\mathbf{K}$ the stiffness matrix associated with a potentially nonlinear problem, and $\mathbf{f}$ the projected loading forces. Here the $\boldsymbol{\lambda}$ represents the boundary forces computed from the fluid flow problem and imposed on the fluid-structure interaction interface, and therefore a restriction of $\overline{\boldsymbol{t}}$. Each matrix and vector of this semi-discrete equation are properly defined by assembling locally computed array on each element with the polynomial basis N_e of $\mathcal{P}(\kappa_e)$:

$$\begin{aligned} \mathbf{M}_{s,e} &= \int_{\kappa_e} \rho_s N_e^T N_e \\ \mathbf{K}_{s,e}(\mathbf{u}_e)\mathbf{u}_e &= \int_{\kappa_e} \nabla N : \boldsymbol{\sigma}(\mathbf{u}_e N_e) \\ \mathbf{f}_{s,e} &= \int_{\kappa_e} N_e^T \overline{\boldsymbol{b}}_e + \int_{\partial\kappa_e} N_e^T \overline{\boldsymbol{t}}_e \end{aligned} \tag{7}$$

The time integration of the structure problem can be carried out by method such as standard time stepping schemes [6, 40] or by the so called Finite Element in time as introduced in [1]. In [19], the use of Finite Element in time for the structure part in a fluid-structure interaction context is shown to increase the overall computational with no noticeable advantages. The Generalized HHT-α method [14] is used herein. The time interval $[0, T]$ is discretized into a finite number of time-steps t_N such as $t_0 = 0$ and $t_{N_{\max}} = T$. In a typical time step size $\Delta t = t_{N+1} - t_N$, and time derivatives are approximated with:

$$\begin{aligned} \mathbf{u}_{N+1} &= \mathbf{u}_N + \Delta t \dot{\mathbf{u}}_N + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{u}}_N + \beta \ddot{\mathbf{u}}_{N+1} \right] \\ \dot{\mathbf{u}}_{N+1} &= \dot{\mathbf{u}}_N + \Delta t \left[(1-\gamma)\ddot{\mathbf{u}}_N + \gamma \ddot{\mathbf{u}}_{N+1} \right] \\ \mathbf{u}_{N+\alpha_f} &= (1-\alpha_f)\mathbf{u}_N + \alpha_f \mathbf{u}_{N+1} \\ \ddot{\mathbf{u}}_{N+\alpha_m} &= (1-\alpha_m)\ddot{\mathbf{u}}_N + \alpha_m \ddot{\mathbf{u}}_{N+1} \end{aligned} \tag{8}$$

In the semi-discrete form of the solid equation of motion in Eq. (6), the acceleration $\ddot{\mathbf{u}}$ and the displacement $\mathbf{u}$ are evaluated at $t_{N+\alpha_f}$ and $t_{N+\alpha_m}$ For the elastic linear case, it is shown [14] that there are optimum values for the parameters β, γ, α and α for a given spectral radius $\rho_\infty \in [0, 1]$.

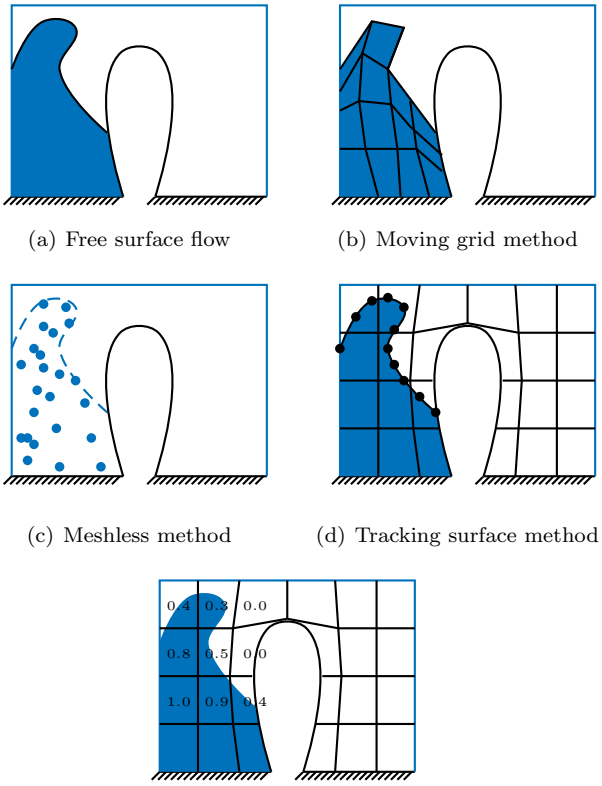
$$\beta = \frac{(1+\alpha_m-\alpha_f)^2}{4}, \gamma = \frac{1}{2} + \alpha_m - \alpha_f, \alpha_f = \frac{1}{1+\rho_\infty} \text{ and } \alpha_m = \frac{2-\rho_\infty}{1+\rho_\infty} \tag{9}$$

The spectral radius controls the numerical damping of the time integration scheme. The damping decreases with smaller values of ρ_∞ which is maximum for $\rho_\infty = 0$. For $\rho_\infty = 1$ the method is the classic trapezoidal rule. Other time integration schemes can be easily derived from this general formulation [36].

3 Description of the flow with free-surface

3.1 Free-surface flow equations

To model free surface flows, a traditional approach is to consider simplified model such as the Nonlinear Shallow Water Equations first introduced by Saint-Venant in 1837. Extensive literature on how to solve those equations analytically or numerically can be found in a number of previous works (*e.g.* see [71, 70]). If this approach gives good results for the propagation phase of the wave, in [22]



(a) Free surface flow (b) Moving grid method

(c) Meshless method (d) Tracking surface method

(e) Tracking volume method (VOF)

Figure 2: Techniques to solve a free-surface flow

these models are shown to be unable to represent accurately the impact process of the wave, as well as its sloshing process (see Fig. 2(a)).

For this reason, another appropriate model has to be chosen near a structure hit by water. To represent the complex flow an idea could be to consider a moving fluid domain where the shape of the wave is described by the motion of the domain. However, the domain motion is so complicated that this strategy is not possible to use the traditional ALE formulation given above.

Several alternative approaches are thus proposed:

Meshed Lagrangian approach: (see Fig. 2(a)) are used to represent the free surface fluid flows by using constant re-meshing; therefore, efficient mesh generators are needed in order to compute the new topology at each iteration. For instance, the PFEM method [42] uses one such fast meshing algorithm and solve this kind of problems.

Meshless Lagrangian approach: (see Fig. 2(b)) such as the SPH has the advantage not to require the re-meshing step of the previous strategies [7, 15, 60, 74, 75].

Surface tracking methods (ALE): (see Fig. 2(d)) has the advantage that

the representation of the fluid domain and its free surface is independent from the representation of the flow field. Hence, the resolution of the surface and that of the flow field may be chosen independently from one another. Of course, the level of details should be comparable in order to resolve the fluid motion properly. However, this freedom is helpful, for example, in order to improve the accuracy of the evaluation of the surface tension force (*e.g.* see [56]).

The relative efficiency of this method will not be discussed herein. As the result of the initial choice made to couple a FV code, non of them was used. We use an Arbitrary Lagrangian-Eulerian description of a two-phase flow (taking into account both water and surrounding air) discretized by Finite Volume (see [72, 33, 5]). In this Volume-Of-Fluid (V.O.F.) method, an indicator function (volume fraction, level set or phase-field) is used to represent the interface (see Fig. 2(e)); the main remaining issue is how to convect the interface without diffusing, dispersing or wrinkling it [21, 33, 55, 72].

Remark: The incompressible hypothesis can be discussed, especially for highly aerated waves. For more details on compressible multiphasic simulations, the reader is invited to consults works of [21].

For complex flows (with jets, cavitation and aeration in the sloshing wave) it is natural to consider the Navier-Stokes equations for two immiscible and incompressible flows (water and air for instance) occupying transient domains $\Omega_i(t)$ so that the whole fluid domain considered $\Omega_f(t) = \Omega_1(t) \cup \Omega_2(t)$. The interface between the both domains Ω_1 and Ω_2 is denoted Γ. In space-time domain $\Omega_f(t) \times [0,T]$, the Navier-Stokes equations formulated in an ALE framework apply. For an arbitrary motion of the total fluid domain Ω_f described by a displacement field $\boldsymbol{u}_m$, it can be written as:

$$\begin{array}{rcll} \rho\partial_t \boldsymbol{v} + \rho\left(\boldsymbol{v} - \dot{\boldsymbol{u}}_m\right)\nabla\cdot\boldsymbol{v} - \nabla\cdot 2\mu\mathbf{D}(\boldsymbol{v}) & = & -\nabla p + \boldsymbol{f}_\Gamma + \rho\boldsymbol{g} & \text{in } \Omega_f(t)\times[0,T] \\ \nabla\cdot\boldsymbol{v} & = & 0 & \text{in } \Omega_f(t)\times[0,T] \end{array} \tag{10}$$

where p denotes the pressure field, $\boldsymbol{v}$ the velocity. We also introduce $\boldsymbol{g}$ that depicts the gravity field, $\boldsymbol{f}_\Gamma$ the surface tension forces. It can be expressed as $\boldsymbol{f}_\Gamma = \sigma\kappa\delta_\Gamma$, where σ is the surface tension, κ the curvature of the free-surface (*i.e.*interface between Ω_1 and Ω_2) and δ_Γ the mass distribution concentrated at the surface (equivalent to a Dirac distribution). Fluid material properties are the dynamic viscosity μ and the density ρ. To write a unique formulation in the whole domain $\Omega_f(t)$ we express the local material property values as the function of ι:

$$\rho = \iota\rho_1 + (1-\iota)\rho_2 \quad \text{and} \quad \mu = \iota\mu_1 + (1-\iota)\mu_2 \qquad \text{in } \Omega_f(t)\times[0,T] \tag{11}$$

where the characteristic function or fluid volume fraction ι is defined as:

$$\iota(\boldsymbol{x},t) = \begin{cases} 1, & \text{for} \quad \boldsymbol{x}\in\Omega_1(t) \\ 0, & \text{for} \quad \boldsymbol{x}\in\Omega_2(t) \end{cases} \tag{12}$$

Thus, the fluid volume fraction ι and the mass distribution δ_Γ are linked by the relation:

$$\nabla\iota = \delta_\Gamma\boldsymbol{n} \tag{13}$$

To close the set of equations we have to write the conservation of ι. When no reaction between phases occurs, the fluid volume fraction evolves only by

advection:

$$\partial_t \iota + (\boldsymbol{v} - \dot{\boldsymbol{u}}_m) \cdot \nabla \iota = 0 \tag{14}$$

The conservation equation system on the whole domain Ω_f in d dimensions can be written as a function of the $2d + 2$ unknowns: $\boldsymbol{u}_m$, $\boldsymbol{v}$, p and ι which are sought in their respective spaces $\mathcal{U}$, $\mathcal{V}$, $\mathcal{P}$ and $\mathcal{I}$ that denote the restriction of Hilbert and Sobolev spaces with suitable boundary condition. Thus, we aim to find: $(\boldsymbol{u}_m, \boldsymbol{v}, p, \iota) \in \mathcal{U} \times \mathcal{V} \times \mathcal{P} \times \mathcal{I}$, such that equations (10) and (11) are verified.

3.2 Arbitrary moving domain description

A difficulty in fluid-Structure interaction problems pertains to unsteadily moving domains for the fluid part, as the fluid-structure interface follows the deformations of the structure. Traditional Computational fluid Dynamic programs solve the fluid equations on a fixed (Eulerian) grid. The classical approach to overcome this difficulty is to consider the so-called Arbitrary Lagrangian Eulerian (ALE) method where the whole grid is moved inside the fluid domain following the movement of the boundary [17, 30, 37, 39, 54, 78, 56]. However, this leads to a new difficulty: knowing the new shape of the boundary, how can one keep the quality and the validity of the fluid inner mesh? In more mathematical terms, this means building a suitable map for the domain motion given its interface displacement. The fluid displacement $\boldsymbol{u}_m$ is arbitrary inside the domain Ω_f, but has to fulfill the condition:

$$\boldsymbol{u}_m = \overline{\boldsymbol{u}}_m \qquad \text{on } \partial\Omega_f(t) \times [0, T] \tag{15}$$

Thus, the fluid displacement is an arbitrary extension of $\overline{\boldsymbol{u}}_m|_{\partial\Omega_f}$ inside the fluid domain Ω_f:

$$\boldsymbol{u}_m = \text{Ext}\left(\overline{\boldsymbol{u}}_m\Big|_{\partial\Omega_f}\right) \tag{16}$$

Let us consider a domain $\mathcal{T}^h\left(\Omega_{f,t_N}\right)$ which represent the mesh configuration at a time t_N. This domain is moving according to the imposed condition at the boundaries, in terms of either imposed displacement or velocity. The issue is to build a new valid mesh $\mathcal{T}^h\left(\Omega_{f,t_{N+1}}\right)$ at $t_{N+1} = t_N + \Delta t$ knowing the initial valid mesh $\mathcal{T}^h\left(\Omega_{f,t_N}\right)$ and imposed boundary conditions.

It can be proved that the mesh motion problem is formally equivalent to a solid body under large deformations. The cost to solve the resulting set of non-linear equations is tremendous and this strategy cannot be applied within an efficient 3D CFD code. A traditional way to overcome this difficulty is to consider simplified solid equations for the mesh motion:

Spring analogy aims at linking each point of the mesh by fictitious spring. In [46], authors show this yields failure modes. The failure modes of this kind can be eliminated by introducing non-linear and torsional springs. However, the cost induced by the improvement of this imperfect system by nature can be deemed as too large.

Pseudo-solid equations can be considered as a simplification of the 3D non-linear problem to the linear one; the analogy is a mechanical problem with small deformations.

Laplacian smoothing operator is different strategy where we consider the Laplace smoothing equation:

$$\nabla \cdot (\gamma \nabla \boldsymbol{u}_m) = 0 \qquad \text{in} \quad \Omega_f \tag{17}$$

A study of some mesh motion algorithms for the examples presented herein is given in [47][2]. The main conclusion of this work can be summarized as:

Laplace smoothing operator with a non constant diffusion coefficient can handle large deformation of the mesh while the rotations remains small.

Pseudo-solid formulation is of interest when the motion of the mesh is undergoing large deformation and is mainly governed by rotation.

When the mesh motion are too large (this is not the case herein), strategies such as Cartesian Cut Cells [12], Immersed Fictitious Elements [78], or even meshless methods [60] can be used.

3.3 Finite Volume discretization

The continuous equations are transformed into a set of algebraic equations that can be solved numerically. Indeed, the goal of this work is first to emphasize the possibility of coupling different codes, with different methods (FEM for solid, FVM for fluid) and second to follow the trends of the computational scientific software market. And, as FVM, contrary to FEM, leads to intrinsically conservative methods at the local stage they are often preferred in commercial and non-commercial softwares (Phoenics, fluent, flow3D, Star-CD, Code_Saturne, OpenFOAM [45] are FVM based, Adina is FEM based). There are a great number of books of reference on the topic, among them [28, 64].

The FV formulation can be written directly using an integrated form of the conservation equation written in (10). Another possibility is to consider the restriction of the solution space of weak form problems. For homogeneity with the solid method we chose to describe the FV strategy in this framework. The weak form of the Navier-Stokes equation can be written as follows [35]. Find $(\boldsymbol{u}_m, \iota, \boldsymbol{v}, p) \in \mathcal{U}\times\mathcal{I}\times\mathcal{V}\times\mathcal{P}$, such that, for all $(\delta\boldsymbol{u}_m, \delta\iota, \delta\boldsymbol{v}, \delta p) \in \mathcal{U}_0\times\mathcal{I}_0\times\mathcal{V}_0\times\mathcal{P}_0$:

$$\begin{aligned}
\mathcal{G}_f \quad := \quad & \int_{\Omega_f} \nabla \cdot (\gamma \nabla \boldsymbol{u}_m) \delta \boldsymbol{u}_m + \int_{\Omega_f} \left(\partial_t \iota + (\boldsymbol{v} - \dot{\boldsymbol{u}}_m) \nabla \iota\right) \delta \iota \\
& \int_{\Omega_f} \rho \partial_t \boldsymbol{v} \cdot \delta \boldsymbol{v} + \int_{\Omega_f} \rho \nabla (\boldsymbol{v} - \dot{\boldsymbol{u}}_m) \otimes \boldsymbol{v} \cdot \delta \boldsymbol{v} - \int_{\Omega_f} \nabla \cdot \mu_f \mathbf{D}(\boldsymbol{v}) \cdot \delta \boldsymbol{v} \\
& + \int_{\Omega_f} p \nabla \cdot \delta \boldsymbol{v} + \int_{\Omega_f} \nabla \cdot \boldsymbol{v} \delta p + [\text{B.C. in a weak form}] \\
= \quad & 0
\end{aligned} \tag{18}$$

For this method, the whole volume Ω_f is divided into a set of of discrete elements, here called discrete volumes $(\kappa_{f,e})_{e=1,n_{el}}$ covering the whole domain $(\Omega_f = \cup_{e=1}^{n_{el}} \kappa_{f,e})$ without overlapping $(\cap_{e=1}^{n_{el}} \kappa_{f,e} = \emptyset)$. For a Finite Element discretization, the solution spaces are restricted to suitable spaces of piecewise

[2] See also the reference thread in http://www.cfd-online.com/Forums/openfoam-solving/57916-moving-mesh-problem-openfoam-141-a.html OpenFOAM forum

polynomial functions over the set of discrete elements. For a Finite Volume discretization used herein, the test functions are chosen in the space of characteristic discrete volume functions. For instance, the velocity can be approximated as:

$$\mathcal{V}^h = \mathcal{V} \cap \left\{ \boldsymbol{v} \middle| \boldsymbol{v}|_\kappa \in \text{span}(\iota_\kappa), \forall \kappa \in \mathcal{T}^h(\Omega_f) \right\} \tag{19}$$

where ι_κ is the characteristic function of the element defined as:

$$\begin{array}{rccl} \iota_\kappa : & \Omega_f & \longrightarrow & \mathbb{R} \\ & \boldsymbol{x} & \longrightarrow & \left\{ \begin{array}{lll} 1 & \text{if} & \boldsymbol{x} \in \kappa \\ 0 & \text{if} & \boldsymbol{x} \in \Omega / \kappa \end{array} \right. \end{array} \tag{20}$$

The same kind of restriction holds for the solution spaces of equation (18). Therefore, the function are piecewise constant by elements, and with the restriction of the weak formulation reduces to: find $(\boldsymbol{u}_m^h, \iota^h, \boldsymbol{v}^h, p^h) \in \mathcal{U}^h \times \mathcal{I}^h \times \mathcal{V}^h \times \mathcal{V}^h$, such that,for all $(\delta\boldsymbol{u}_m^h, \delta\iota^h, \delta\boldsymbol{v}^h, \delta p^h) \in \mathcal{U}_0^h \times \mathcal{I}_0^h \times \mathcal{V}_0^h \times \mathcal{V}_0^h$: $\mathcal{G}_f((\boldsymbol{u}_m^h, \iota^h, \boldsymbol{v}^h, p^h); \delta\boldsymbol{u}_m^h, \delta\iota^h, \delta\boldsymbol{v}^h, \delta p^h) = 0$. The divergence terms in equation (18) can be written in terms of flux at the boundary of volume controls using the Gauss theorem. Hence, the weak formulation can be written as:

$$\begin{aligned}
\boldsymbol{0} &= \sum_\kappa \left\{ \oint_{\partial\kappa} \mathrm{d}\boldsymbol{\Gamma} \cdot (\gamma \nabla \boldsymbol{u}_m) \right\} - [\text{B.C.}] \\
\boldsymbol{0} &= \sum_\kappa \left\{ \int_\kappa \partial_t \iota + \oint_{\partial\kappa} \mathrm{d}\boldsymbol{\Gamma} \cdot (\boldsymbol{v} - \dot{\boldsymbol{u}}_m)\, \iota \right\} - [\text{B.C.}] \\
\boldsymbol{0} &= \sum_\kappa \left\{ \int_\kappa \rho \partial_t \boldsymbol{v} + \oint_{\partial\kappa} \rho\, \mathrm{d}\boldsymbol{\Gamma} \cdot (\boldsymbol{v} - \dot{\boldsymbol{u}}_m) \otimes \boldsymbol{v} - \oint_{\partial\kappa} \mathrm{d}\boldsymbol{\Gamma} \cdot \mu_f \mathbf{D}(\boldsymbol{v}) + \oint_{\partial\kappa} p\, \mathrm{d}\boldsymbol{\Gamma} \right\} - [\text{B.C.}] \\
\boldsymbol{0} &= \sum_\kappa \left\{ \oint_{\partial\kappa} \mathrm{d}\boldsymbol{\Gamma} \cdot \boldsymbol{v} \right\} - [\text{B.C.}]
\end{aligned}$$

where $\mathrm{d}\boldsymbol{\Gamma}$ is the elementary surface vector. Note that there is no continuity requirement for the solution (contrary to classical FE), and therefore the approximate solutions are not properly defined at the interface. Their flux can be computed without being imposed by the restriction of the solution space to the FV space. The only difficulty is now to build an accurate representation of the fluxes at the boundaries from a piecewise constant field.

On each control volume, three levels of numerical approximations are applied to build the boundary fluxes:

interpolation: to express variable values at the control volume surface in terms of nodal values (depending on where the variable is stored).

differentiation: to build convective and diffusive fluxes the value of the gradient of the quantity of interest – or at least its approximation – is required.

integration: to approximate surface and volume integral using quadrature formulæ.

The details of constructing an accurate representation of the derived fields (especially of the convection terms that require often special cares) will not be exposed here. Some schemes seek to stabilize the solution at the expense of accuracy, others, such as the MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) exhibits good stability properties for a high-order scheme [28, 66].

The fluid mesh motion considers that $\mathbf{u}_m$ is imposed by the motion of the interface $\mathbf{u}$ and can be written in a semi discrete form as:

$$\mathcal{R}_m(\mathbf{u}_m;\mathbf{u}) := \mathbf{K}_m\mathbf{u}_m - \mathbf{D}_m\mathbf{u} = \mathbf{0} \tag{21}$$

where $\mathbf{D}_m$ is a projection/restriction operator and $\mathbf{K}_m$ governs the extension of the boundary displacement either by a diffusion process or a pseudo-solid equation (see Section 3.2 above). The volume fraction $\boldsymbol{\iota}$, the d components of velocity $\mathbf{v}$ and pressure p are coupled through with a set of non-linear equations. Written in a matrix forms, it gives the following semi-discrete problem:

$$\mathcal{R}_f(\boldsymbol{\iota},\mathbf{v}_f,\mathsf{p}_f;\mathbf{u}_m) := \begin{bmatrix} \mathbf{M}_\iota\dot{\boldsymbol{\iota}} + \mathbf{N}_\iota(\mathbf{v}_f - \dot{\mathbf{u}}_m)\boldsymbol{\iota} \\ \mathbf{M}_f(\boldsymbol{\iota})\dot{\mathbf{v}}_f + \mathbf{N}_f(\boldsymbol{\iota},\mathbf{v}_f - \dot{\mathbf{u}}_m)\mathbf{v} + \mathbf{K}_f(\boldsymbol{\iota})\mathbf{v}_f + \mathbf{B}_f\mathsf{p}_f - \mathbf{f}_f(\boldsymbol{\iota}) \\ \mathbf{B}_f^T\mathbf{v}_f \end{bmatrix} = \mathbf{0} \tag{22}$$

where $\mathbf{M}_\iota$ and $\mathbf{N}_\iota$ are the matrices associated to the advection problem of the fluid volume fraction, $\mathbf{M}_f$ is a positive definite mass matrix, $\mathbf{N}_f$ is an unsymmetrical advection matrix, $\mathbf{K}_f$ is the conduction matrix describing the diffusion terms, and $\mathbf{B}_f$ stands for the gradient matrix, whereas $\mathbf{f}_f$ is the discretized nodal loads on the flow. This matrix form also takes into account the boundary conditions; special care has to be taken concerning the discretization of boundary conditions – and especially normal flux – when using the Finite Volume Method [34].

3.4 Semi-implicit non-linear solver for the fluid problem

One way to solve this problem is to consider a monolithic application. Another way is to consider a split between the mesh motion, the volume fraction advection, the momentum and the continuity equations, and to solve it thanks to an operator split-like procedure often termed as the *segregated approach* [64]. This approach is favored for its computational efficiency compared to the monolithic one. Indeed, even with a simple fixed point iteration strategy its cost is less important than that of the monolithic approach for large size problems [28]. In the work presented herein, the later will be used.

Let us note that for a given motion of the fluid domain, the coupling between the mesh deformation and the Navier-Stokes equation is weak, in the sense that no variable like velocity $\boldsymbol{v}$ or pressure p influences the fluid domain deformation under imposed boundary displacements. The coupling between the mesh motion problem and the fluid momentum equation can therefore be insured explicitly.

The only remaining question is the choice of velocity in the time step $\Delta t = t_{N+1} - t_N$. As the mesh motion is arbitrary and does not rely on any physical phenomenon, it is *a priori* possible to take any velocity evolution on the window $[T_N, T_{N+1}]$ so that the initial mesh deformation is equal to $\boldsymbol{u}_{m,N}$ and the final mesh deformation is equal to $\boldsymbol{u}_{m,N+1}$.

The Geometric Conservation Law demands a numerical scheme to reproduce exactly and independently from the mesh motion a constant solution. This condition can be found in the literature for ALE formulation discretized either by the Finite Volume [17] or the Stabilized Finite Element methods [30]. It is proven [23] that the velocity of the dynamic mesh needs to be computed for all first– and second–order time accurate methods like implicit Euler or Crank-

Nicholson that the mesh motion velocity so that:

$$\dot{\boldsymbol{u}}_m = \frac{\boldsymbol{u}_{m,N+1} - \boldsymbol{u}_{m,N}}{\Delta t} \tag{23}$$

The volume fraction function is supposed to be sharp at the interface between water and gaze, and therefore, standard FV discretization that can be strongly diffusive cannot be applied, as they would smear the interface. A way to guarantee a sharp and bounded solution is to solve it with a numerical scheme designed for the multi-dimensional advection equation [55]. We will not enter into the details of such a treatment, but let us note that, in our case, the treatment of the volume fraction require time sub-cycles, as an explicit treatment termed as MULES (Multidimensional Universal Limiter with Explicit Solution) is used (see for instance [72, 5]).

For the coupling between the pressure equation and the momentum equilibrium, strategy based upon ACM (Artificial Compressibility Method) [13], or pressure correction techniques such as SIMPLE (Semi-Implicit Method for Pressure Linked Equations) [63, 73] or PISO (Pressure Implicit with Splitting of Operators) [43, 44] are traditionally used in CFD. They often rely on the use of suitable relaxation parameters [28] in order to reach convergence for the stiff coupled problem. In [68], a comparison between ACM and pressure-correction techniques is given and in [2], a comparison of two pressure-correction algorithms showed the overall better performances of PISO-like algorithms over SIMPLE ones.

In this work, PISO-like algorithms are used to solve our CFD problem. The semi-discrete form of the Navier-Stokes equation is discretized in time using implicit or explicit integration schemes, such as Euler explicit and implicit, or a second order Crank-Nicholson scheme. The discretized momentum Eq. (22) is split in the following way when an implicit integration scheme is used:

$$\mathcal{A}_f(\mathbf{v}_{N+1})\mathbf{v}_{N+1} - \mathcal{H}_f(\mathbf{v}_N, \mathbf{v}_{N+1}) = -\mathbf{B}_f \mathsf{p}_{N+1} \tag{24}$$

where $\mathcal{A}_f$ stands for time derivative terms in a cell (and is therefore diagonal) and $\mathcal{H}_f$ takes into account all neighboring velocity and source terms in elements. The incompressibility condition can be re-written in a discrete form using the previous split as:

$$\mathbf{B}_f^T \frac{1}{\mathcal{A}_f(\mathbf{v}_{N+1})} \mathbf{B}_f \mathsf{p}_{N+1} - \mathbf{B}_f^T \frac{1}{\mathcal{A}_f(\mathbf{v}_{N+1})} \mathcal{H}_f(\mathbf{v}_N, \mathbf{v}_{N+1}) = \mathbf{0} \tag{25}$$

For the PISO algorithm, the coupling between the incompressibility condition and the momentum equilibrium parts of the Navier-Stokes equation is assured in an iterative way as detailed in **Algorithm 1**.

The PISO algorithm is not fully implicit, as corrective term of the velocity is introduced explicitly. Hence, in the correction step, it is supposed that the influence of the transported term is negligible compared to the pressure gradient correction terms. Therefore, even with an implicit time integration scheme, the stability of the PISO algorithm remains conditional, and when the Courant Number becomes too large (it means that the transport due to the flow over a cell is not well captured) the PISO algorithm fails to converge.

Remark: in the algorithm given, the non-orthogonal correctors applied to build more accurate flux terms when the mesh is not orthogonal grid are not detailed. For further details, see [28, 79].

Algorithm 1 Pressure Implicit Splitting of Operators algorithm

1: Given: initial velocity $\mathbf{v}_0$ and pressure p_0.
2: **while** $T < T_{\max}$ **do**
3: Momentum predictor:

$$\mathcal{A}_f(\mathbf{v}_{N+1}^{(0)})\mathbf{v}_{N+1}^{(0)} - \mathcal{H}_f(\mathbf{v}_N, \mathbf{v}_{N+1}^{(0)}) = -\mathbf{B}_f \mathsf{p}_N$$

4: **for** $(k) = 0$ to $(k) < (k_{\max})$ **do**
5: Pressure correction:

$$\mathbf{B}_f^T \cdot \frac{1}{\mathcal{A}_f(\mathbf{v}_{N+1}^{(k)})} \mathbf{B}_f^T \mathsf{p}_{N+1}^{(k)} = \mathbf{B}_f^T \cdot \frac{\mathcal{H}_f(\mathbf{v}_N, \mathbf{v}_{N+1}^{(k)})}{\mathcal{A}_f(\mathbf{v}_{N+1}^{(k)})}$$

6: Explicit velocity correction

$$\mathbf{v}_{N+1}^{(k+1)} = \mathbf{v}_{N+1}^{(k)} - \frac{1}{\mathcal{A}(\mathbf{v}_{N+1}^{(k)})} \mathbf{B}_f \mathsf{p}_{N+1}^{(k)}$$

7: **end for**
8: **end while**

4 Implicit coupling using algorithm Direct Force-Motion Transfer and Block-Gauss-Seidel with Aitken's relaxation

By enforcing the continuity of primal variables at the interface we can eliminate the energy errors that characterize the explicit interface matching. When coupling incompressible flow with structure, the implicit interface matching is required for stability reason, as proved in [48]. This ought to be done by iterating on the following residual to reduce its value below the chosen tolerance:

$$\boldsymbol{r}_{N+1} := \boldsymbol{u}_{s,N+1} - \boldsymbol{u}_{f,N+1} \simeq 0 \quad \leq \texttt{TOL} \tag{26}$$

In this way we obtain an implicit algorithm requiring more than one iteration to enforce the interface matching condition. The chosen order of iterations, corresponds to the Block-Gauß-Seidel algorithm for fluid-structure interaction problem [58]. Let us note that not only the value at synchronization points T_n or T_{n+1}, but also the interpolated evolution of variables have to be exchanged in the entire time-interval $t \in [T_n, T_{n+1}]$ when the time steps are not matching between fluid and structure sub-problems.

Contrary to explicit algorithms which generate spurious energy at the interface, the present implicit interface matching algorithm enforce the same evolution of the primal variables at the fluid-structure interface. In other words, an iterative solution for primal (displacements) continuity as well as the dual (forces) equilibrium equations at the interface is performed by using the Picard iteration:

$$\boldsymbol{u}_{N+1}^{(k+1)} = \mathcal{G}\left(\boldsymbol{u}_{N+1}^{(k)}\right); \qquad \mathcal{G} = {\mathcal{S}_s}^{-1} \circ -\mathcal{S}_f \tag{27}$$

where $\mathcal{S}_f$ and $\mathcal{S}_s$ are Steklov-Poincaré operators for fluid and structure defined as define in [18]. These operators can be formulated using transfer operators and equations (6), (22) and (21):

$$\mathcal{S}_s = \mathcal{T}_s^{\boldsymbol{\lambda}} \circ \mathcal{R}_s \circ \mathcal{T}_s^{\boldsymbol{u}}; \qquad \mathcal{S}_f = \mathcal{T}_f^{\boldsymbol{\lambda}} \circ \mathcal{R}_f \circ \mathcal{R}_m \circ \mathcal{T}_s^{\boldsymbol{u}} \tag{28}$$

where the transfer of structure displacement to fluid-structure interface displacement is $\mathcal{T}_s^{\boldsymbol{u}}$, the transfer of fluid displacement to interface displacement $\mathcal{T}_s^{\boldsymbol{\lambda}}$, the transfer of structure stresses to the interface $\mathcal{T}_f^{\boldsymbol{u}}$ and the transfer of fluid stresses to the interface $\mathcal{T}_f^{\boldsymbol{\lambda}}$.

The Picard iterations will continue until convergence of interface residual is achieved:

$$\boldsymbol{r}_{N+1}^{(k)} = \boldsymbol{u}_{s,N+1}^{(k)} - \boldsymbol{u}_{f,N+1}^{(k)} = \mathcal{G}\left(\boldsymbol{u}_{N+1}^{(k)}\right) - \boldsymbol{u}_{N+1}^{(k)} \tag{29}$$

Such a Block-Gauß-Seidel algorithm with implicit interface matching, further denoted as DFTM-BGS, can be presented as a natural generalization of the explicit algorithms. We can thus write:

Algorithm 2 Direct Force-Motion Transfer Block-Gauß-Seidel

Given: initial time $T = T_0$, final time $T_{\max}$, window size Δt, initial interface displacement $\boldsymbol{u}_0$.
while $T < T_{\max}$ **do**
  $(k) = 0$
  Predict displacement: $\boldsymbol{u}_{N+1}^{(0)} = \mathcal{P}(\boldsymbol{u}_N^{(k_{\max})}, \dot{\boldsymbol{u}}_N^{(k_{\max})}, \boldsymbol{u}_{N-1}^{(k_{\max})}, \dots)$
  repeat
    Perform Picard iteration: $\mathcal{G}\left(\boldsymbol{u}_{N+1}^{(k)}\right)$
    Compute residual: $\boldsymbol{r}_{N+1}^{(k)} = \mathcal{G}\left(\boldsymbol{u}_{N+1}^{(k)}\right) - \boldsymbol{u}_{N+1}^{(k)}$
    Update interface primal variable: $\boldsymbol{u}_{N+1}^{(k+1)} = \boldsymbol{u}_{N+1}^{(k)} + \boldsymbol{r}_{N+1}^{(k)}$
    do $(k) \leftarrow (k) + 1$
  until $\|\boldsymbol{r}_N^{(k-1)}\| \geq \texttt{TOL}$
  $N \leftarrow N + 1$ and $T \leftarrow T + \Delta t$
end while

It is clear that this fixed-point algorithm based on Picard iterations has the main drawback that the search directions for $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ variables at the interface do not exploit any information from the fixed-point function $\mathcal{G}$ nor the Steklov-Poincaré operators $\mathcal{S}_f$ and $\mathcal{S}_s$. Therefore, quite a few iterations may be needed to reach the convergence.

The stability of such a coupling algorithm is studied in [48]. We give a formal proof of potential numerical instability due to the *added-mass effect* also observed in [31, 53]. In order to improve the convergence of the DFMT-BGS method, we can use a relaxed update:

$$\boldsymbol{u}_{N+1}^{(k+1)} = \boldsymbol{u}_{N+1}^{(k)} + \omega^{(k)}\, \boldsymbol{r}_{N+1}^{(k)} \tag{30}$$

Our favorite choice for constructing $\omega^{(k)}$ is using a secant methods which can keep the cost of each iteration as low as possible. The Aitken's relaxation strategy has been extensively used in fluid-structure interaction [18, 51, 77], and shown sufficient performances to be used in the following.

The use of different different solvers, for the fluid and the structure part, do not provide in general a matching mesh at the interface. Furthermore, even for matching meshes, as the geometries of the domains are not the same on both sides of the interface, an optimal numbering of the nodes can lead to different orders for the interface nodes. In the examples proposed herein, only this latter point is of interest. Last but not least, different discretization techniques (Finite Element versus Finite Volume) or different order p of the polynomials can be used for constructing solution to fluid-structure interaction problem. In the domain of FE applied to mechanical engineering, extensive literature can be found on how to build a consistent interpolation for both subproblems at the interface [25]. For the fluid-structure interaction problems, an interesting review can be found in [16]. In our framework, it was decided not to favor any mesh-based representation of the interface, since, in the most general case, the fluid problem can also be solved by a meshfree-based method [15]. Namely, an interpolation strategy relying on radial basis function is here chosen. This method has already been employed for FSI in [60, 4].

The algorithm presented here is simple to implement. We use for this work the Communication Template Library (CTL, see [57, 61]) that allows to re-use existing codes in a generic way, either called as libraries on the same computer, or as remote executables through network. With the CTL, we are able to couple existing stand-alone software, in a quite straightforward way, even if they are programmed in different langages (`Fortran` for the structure part, `C++` for the fluid part), and to conserve the inner parallelism of each component. For more details on the implementation, the reader is invited to see [49].

5 Numerical simulations

5.1 Two-dimensional sloshing wave hitting a rigid structure

To validate the proposed V.O.F. fluid strategy, the classical dam break example is computed. It is studied from the experimental [10], the analytical [71] as well as the numerically point of view [72, 22], to describe the collapse of a water column hitting a rigid structure.

The geometry of the problem is represented in Fig. 3: it mainly consists on a $400mmmm$ box that contains a water column (initially $146mmmm$) collapsing and hitting a $16mmmm$ rigid obstacle. The high density fluid representing water has the following properties: density $\rho_1 = 1 \times 10^3 kg.m^{-3}$ and kinematics viscosity $\nu_1 = 1 \times 10^{-6} m.s^{-2}$. The low density fluid – air – is described with a density $\rho_2 = 1 kg.m^{-3}$ and a kinematics viscosity $\nu_2 = 1.48 \times 10^{-5} m.s^{-2}$. The flow in each phase is considered to be Newtonian (no turbulence effect is taken into account). The surface tension between each phase is $0.07 kg.s^{-2}$, but regarding the characteristic size of the problem, the surface tension is of little influence and can be neglected.

The computation is run with three meshes, with 2268, 9072 and 36288 cells: around 46 and 50 cells in $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ directions for the coarse grid, 92 cells for the medium grid, and 92 cells are considered fine grid. At $t = 0$, the initial conditions are at rest, and the water column starts to collapse. The computation is run with an time step $\Delta t = 0.005$. The time step can be decreased in order

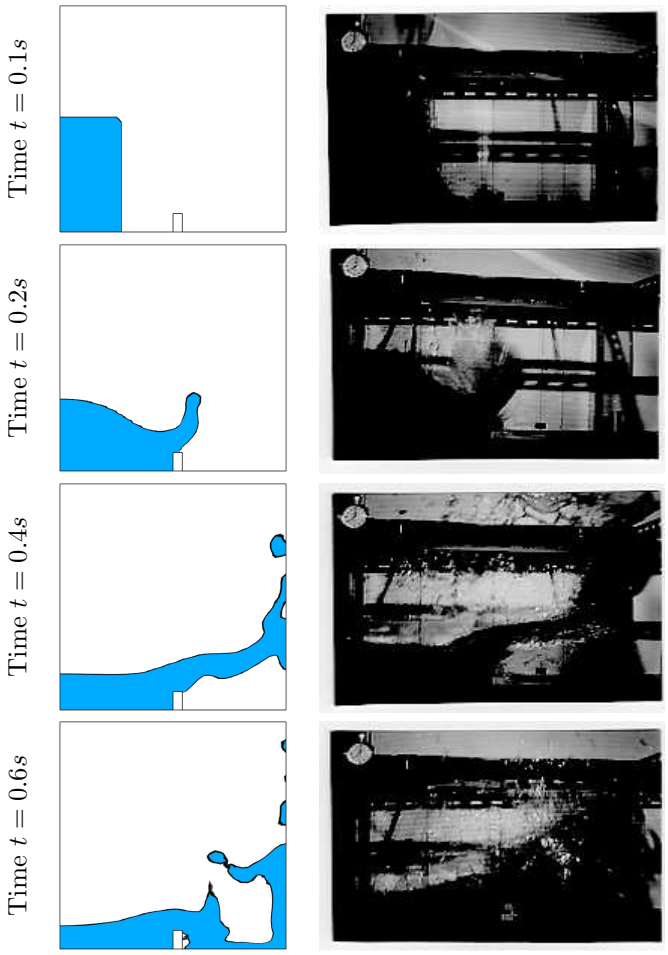


Figure 3: Evolution of the volume fraction $\iota < 0.3$ for the two-phase flow (fine mesh) and comparison with experimental results from [50]

to maintain the Courant–Friedrichs–Lewy condition under 0.5 during the whole simulation. For the coarsest grid considered, no decreasing of the time step is required whereas it is required for the finest grid after the water column hits the rigid obstacle.

In Fig. 3, the evolution of the flow for the finest grid, and a comparison with experimental results [50] is given. We give here only this qualitative comparison. To get more quantitative comparison, one can see the same dam-break problem without obstacle that as been largely used to validate the VOF strategy (see *e.g.* [72, 22]). As we are interested in fluid-structure interaction, the dam-break with an obstacle seems here more relevant with the examples proposed in the following sections.

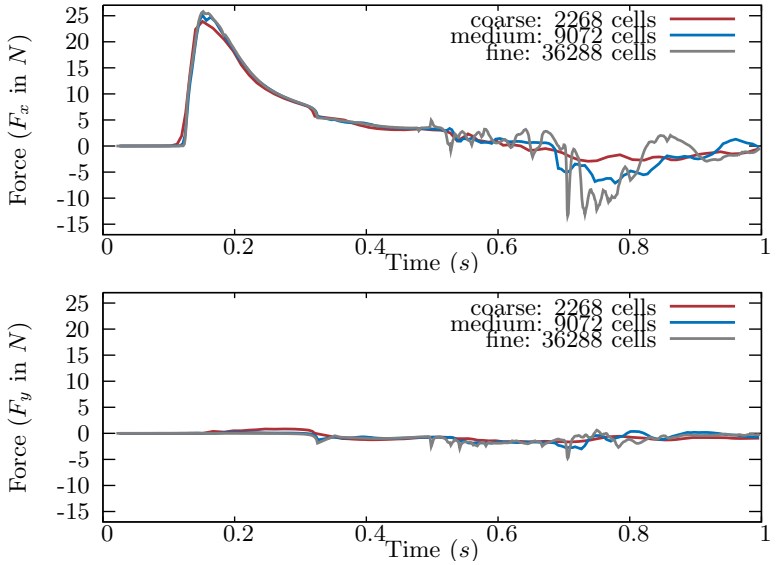


Figure 4: Horizontal and vertical forces acting upon the obstacle for coarse (2268 cells) and fine (9072 cells) grids

In Fig. 4 the forces acting upon the obstacle are also represented. Before the water column hit the structure, the three meshes give roughly the same response. After that, the fine mesh is able to capture the separation of the water phase in many drops and to represent more accurately their separated impact on the right side of the obstacle. It explains the difference observed for $t \geq 0.5$. The use of a turbulence model should be here probably required to obtain more physical results, but at this is not the topic of this paper (*e.g.* see [52, 67]), we will use only laminar models with the coarse and medium mesh densities for the fluid-structure interaction examples provided in the following.

5.2 Two-dimensional sloshing wave hitting a flexible structure

The problem solved herein is a modification of the dam-break problem presented in Sec. 5.1. At initial time $t = 0s$, the same water column starts to fall under the gravity loading. Instead of hitting a rigid structure, its geometry is modified in

order to get a more slender obstacle (Fig. 5). Furthermore, the material properties of the obstacle are modified in order to obtain a flexible elastic structure. This problem was studied experimentally and numerically [76, 3].

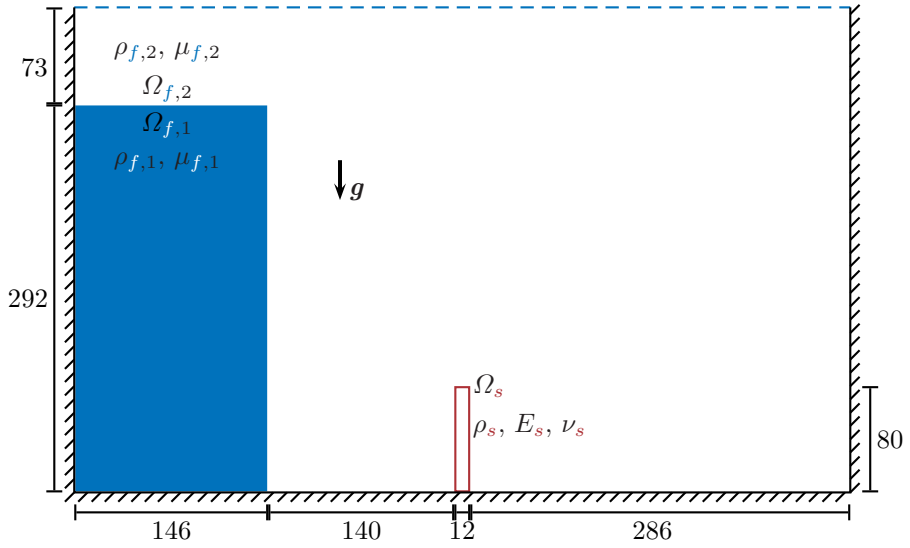


Figure 5: Dam break interacting with an obstacle: geometry (given in mm) and boundary conditions

All boundaries a non-slip boundary condition is applied, except for the upper boundary $z = H$ where the tank is open and a total pressure condition is used:

$$p + \frac{1}{2}\rho \|\boldsymbol{v}\|^2 = p_{\text{atm}} \tag{31}$$

with $p_{\text{atm}} = 0$.

The material properties are imposed as follows: the density and the kinematic viscosity are $\rho_{f,1} = 1 \times 10^3 kg.m^{-3}$ and $\nu_{f,1} = 1 \times 10^5 m.s^{-1}$ for the high density fluid, whereas $\rho_{f,2} = 1 kg.m^{-3}$ and $\nu_{f,2} = 1 \times 10^6 m.s^{-1}$ are considered for the low density fluid domain. Due to the scale of the simulation there is no need to consider surface tension.

Mesh motions based on smoothing operator like the Laplacian equation fails for the used meshes since the points around the obstacle have difficulties to follow the large displacement and rotations of the mesh around the structure. For this reason, the mesh motion problem is solved by using a pseudo-elastic material model where the rigidity is a quadratic inverse function of the distance to the interface between solid and fluid.

The fluid problem is discretized with Finite Volume Method. The results are presented for two meshes with respectively 3340 and 13760 cells. The fluid is handled by second order space discretization and a Van Leer limiter is used for the advected terms. The time integration scheme employed herein is implicit Euler. Small time steps are required by the explicit nature of the coupling

between the phase function indicator problem, the momentum prediction and the pressure correction step.

For structure mechanics problem, it is proposed to use two dimensional elements with quadratic shape functions. Each element therefore contains 9 nodes. For the coarse grid, 51 nodes are considered while 165 nodes are used to discretize the fine grid. The total number of d-o-f given in Tab. 1. The model can handle finite deformations. The material properties used for the solid are: a neo-Hookean elastic material with Young modulus $E_s = 1 \times 10^6 Pa$ and Poisson's ratio $\nu_s = 0$. The chosen value of mass density $\rho_s = 2500 kg \cdot m^{-3}$.

The time integration is handled by a Generalized-α scheme with:

$$\rho_\infty = \frac{1}{2}; \beta = \frac{4}{9}; \gamma = \frac{5}{3} \quad \text{and} \quad \alpha = \frac{2}{3} \tag{32}$$

At each iteration, the linear system is solved by a direct solver for real value asymmetric matrices.

	fluid		solid		number of
Discretization	cells	d-o-f	nodes	d-o-f	time steps
Coarse	3440	17.2×10^3	51	102	1×10^4
Fine	13760	68.8×10^3	165	330	5×10^4

Table 1: Number of d-o-f for coarse and fine discretization of the two-dimensional dam-break problem

The computation is carried out run with a time step of 1×10^{-4} for the coarse and 2×10^{-5} for the fine discretization. The coupling scheme used here is DFMT-BGS with Aitken's relaxation, with the initial parameter value of $\omega = 0.25$. The predictor is of order 1, since computations with second order predictor fail for the finnest grid. The absolute tolerance considered is:

$$\|\boldsymbol{r}_N^{(k)}\| \leq 1 \times 10^{-6}$$

The total number of iteration required to reach the convergence criteria at each time step is given in Fig. 6. Note that no iteration is required before the water hits the structure since the effect of air flow can almost be deemed negligible for this structure. Subsequently, the number of iteration depends on the discretization density. Namely, the finest scale is able to represent fine water sloshing after the breaking of the dam, that will increase the number of iterations needed to ensure the convergence of the implicit coupling. This large number of iteration make the computation costly, and for the finest grid, the total time required to perform the whole coupled simulation on a single $3.0 GHz$ Intel processor is $143 \times 10^3 s$.

In Fig. 8 we represent the domain occuped by high density fluid as well as the stream-lines in the high and low density fluid domains (compare again the results in Fig. 3 for rigid obstacle). For first $0.1s$ of the simulations, the water column is falling under the gravity loading. There is no effect whatsoever on the structure until the high density fluid reaches it. The maximum amplitude of the motion is obtained at $t = 0.25s$, before the solid comes back to its initial position due to flow friction.

After one second of simulation, the fluid is not entirely at rest, but its main effects on the structure are mostly captured. Moreover, we are able to catch the

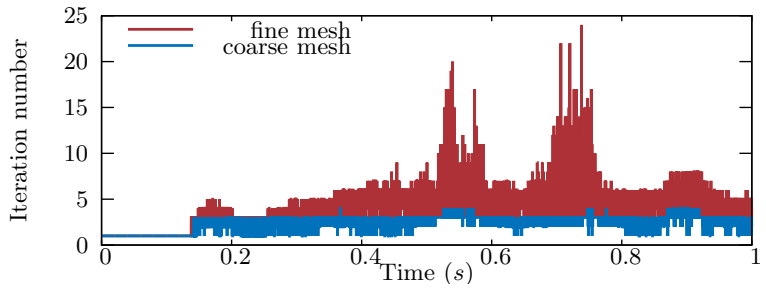


Figure 6: Number of iteration in order to make the DFMT-BGS algorithm converge for the two-dimensionnal dam-break example

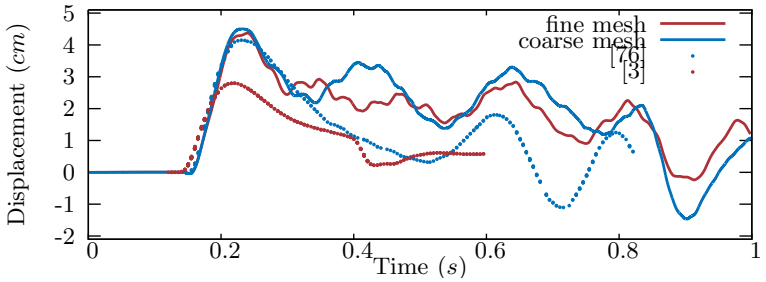


Figure 7: Obstacle extremity displacement in $\boldsymbol{e}_x$ direction for different meshes and comparison with results from literature

main impact with a good accuracy with both the coarse and the fine mesh. Our results are in accordance with the one obtained using a monolithic stabilized-FEM strategy described in [76]. In [3], a tight coupling strategy between a FV and a FEM solver is used, that probably explained the differences observed. After, the impact high density fluid is highly fragmented (see Fig. 8 for time greater than $0.3s$), and the results mostly depend on the ability of the chosen fluid mesh to represent these fine effects, as well as the use of turbulence fluid models not explored herein.

5.3 Three-dimensional sloshing wave impacting a flexible structure

The problem solved in this example is a modified three-dimensional representation of dam-breaking event that brings about a sloshing wave impact on a flexible structure presented in Fig. 9. At initial time $t = 0s$, a three-dimensional water column starts falling down under the gravity loading and eventually hits the obstacle placed in the way. The flexible obstacle is a slender plate-like body made of elastic material that can undergo large deformation. The chosen dimension of the problem, as well as the boundary conditions are given in Fig. 9.

Let us note that we propose to use open boundary conditions far from the obstacle in order to avoid the water bounces-back and hits again the structure after breaking off the walls. For that reason, only the left and bottom planes of the fluid domain are defined as non-slipping walls, while the others are defined

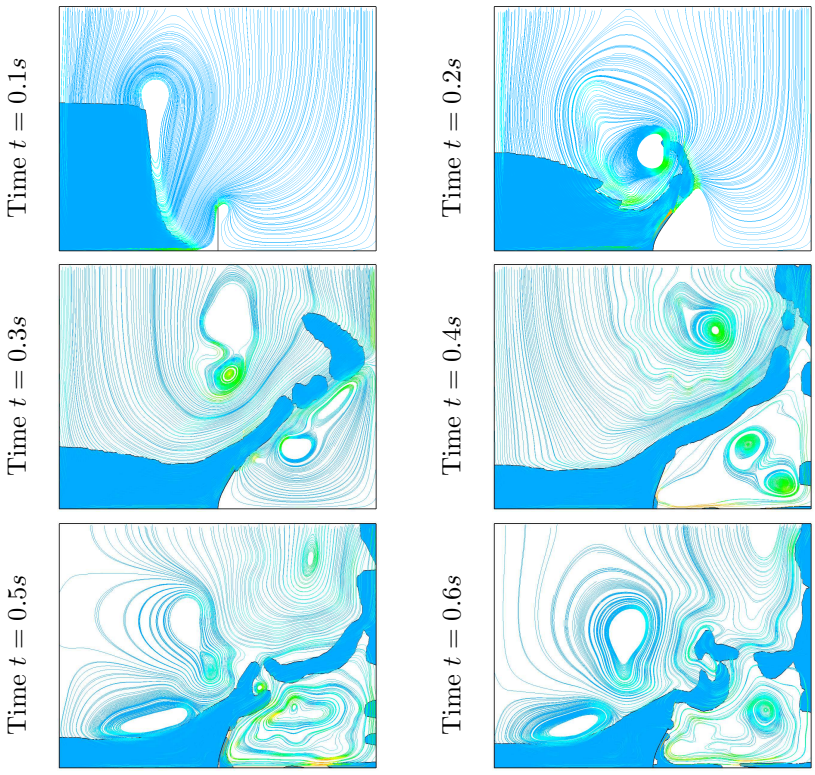


Figure 8: Bi-dimensional dam breaking problem. Evolution of the free surface, structure motion and streamlines for water and air.

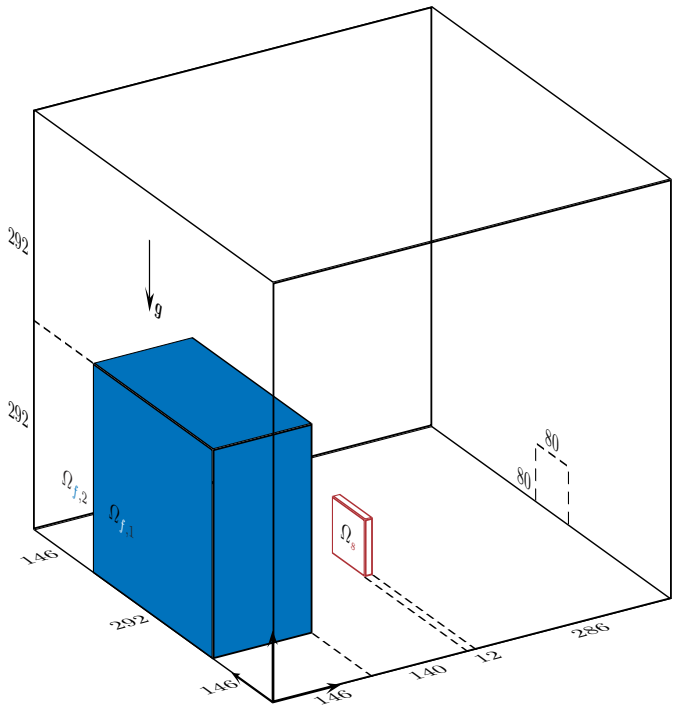


Figure 9: Three-dimensional water column impacting an obstacle: geometry (given in mm) and boundary conditions

with boundary condition of atmospheric pressure.

The material properties are chosen as follows: the density and the kinematic viscosity are $\rho_{f,1} = 1 \times 10^3 kg.m^{-3}$ and $\nu_{f,1} = 1 \times 10^5 m.s^{-1}$ for the high density fluid (water in the reservoir), versus $\rho_{f,2} = 1 \times 10^3 kg.m^{-3}$ and $\nu_{f,2} = 1 \times 10^6 m.s^{-1}$ for the low density fluid (air in the remaining part of the domain). The mesh motion problem is solved by using a Laplacian smoothing material where the diffusion coefficient is a quadratic inverse function of the distance to the interface between solid and fluid.

The results are computed for two meshes with the chosen discretization and the number of cells given in Tab. 2. For the finest grid, around 64, 60 and 40 cells are used in $\boldsymbol{e}_x$, $\boldsymbol{e}_y$ and $\boldsymbol{e}_z$ direction. The mesh is refined gradually refined around the structure, and initially the cell dimensions are between 8.9×10^{-8} and 3.2×10^{-5}. For this finest grid, the maximum skewness of the mesh observed is 2.947, that does not generate too large errors. The fluid is handled by second order space discretization and a Van Leer limiter is used for the advection terms. The time integration scheme employed in this computation is implicit Euler. For such a scale of modeling it is not required to consider surface tension between the two fluids. For this problem the fluid computation is parallelized, but reduction of the CPU time is obtained by using a Generalized Algebraic-MultiGrid (GAMG) linear solver.

Note that small time steps are required for the explicit solution of the phase function indicator equation, as well as the half-implicit nature of the coupling between the momentum predictor and the pressure corrector.

For the structure part, we propose here to use three-dimensional elements with quadratic shape functions, where each element has 27 nodes. The material properties used for the solid are: a neo-Hookean elastic material with Young's modulus $E_s = 1 \times 10^6 Pa$ and Poisson's ratio $\nu_s = 0$ and a density $\rho_s = 2500 kg \cdot m^{-3}$, which can represent finite deformation. The time integration is carried out by a Generalized-α scheme with the same parameters as the one used for the 2D case.

	fluid		solid		number of
Discretization	cells	d-o-f	nodes	d-o-f	time steps
Coarse	13×10^3	63×10^3	363	1.1×10^3	1×10^5
Fine	104×10^3	520×10^3	2205	6.6×10^3	1×10^5

Table 2: Number of d-o-f for coarse and fine discretization of the three-dimensional dam-breaking problem

The computation of the coupled problem (with total number of d-o-f given in Tab. 2), is carried out by an implicit iterative scheme. The results of fluid and solid computations are matched for a time step of 1×10^{-4} for the coarse and 2×10^{-5} for the fine discretization. The coupling scheme used is DFMT-BGS with Aitken's relaxation. The initial parameter is $\omega = 0.25$ and the predictor is of order 1. The absolute tolerance considered is:

$$\|\boldsymbol{r}_N^{(k)}\| \leq 1 \times 10^{-6} \tag{33}$$

The number of iterations required to reach the convergence criteria is given in Fig. 10. Note that there is no iteration required before the water hits the

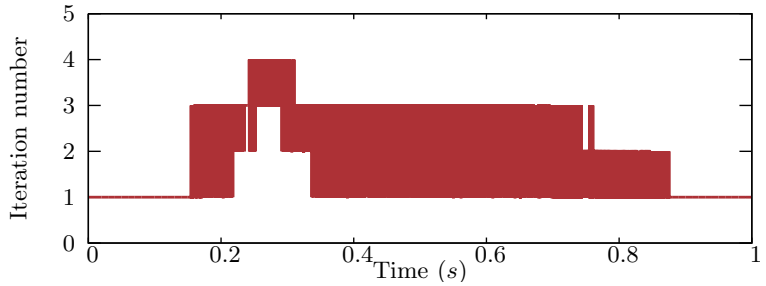


Figure 10: Number of iterations in order to make the DFMT-BGS algorithm converge for the three-dimensional dam-breaking problem

structure since the effect of air flow can almost be deemed negligible with respect to the structure. Subsequently, the number of iterations depends on the discretization density. In reaching the opposite wall, the water does not rebound on the wall but simply flows away. For the finest grid, the total time required to perform the whole coupled simulation on a single $3.0GHz$ Intel processor is $279 \times 10^3 s$.

In Fig. 12, the high density fluid domain is represented, as well as some part of the fluid mesh and the structure displacement. The first $0.1s$ of the simulations, the water column falls under the gravity loading. There is no effect whatsoever on the structure until the high density flow reaches its bottom. The maximum amplitude of the motion is obtained at $t = 0.25s$, before the solid comes back to its initial position and oscillates after the shock.

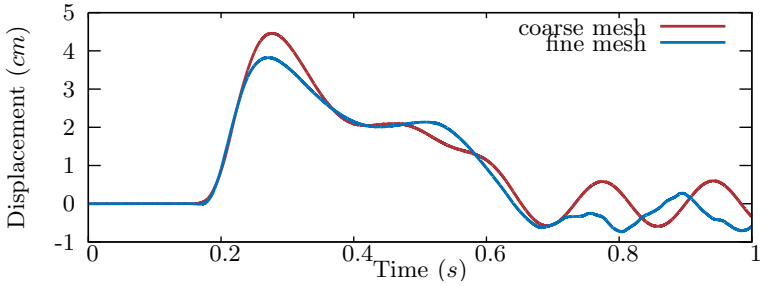


Figure 11: Three-dimensional dam break example: obstacle displacement in $\boldsymbol{e}_x$ direction of center point of the top face $(40; 6; 80)$

In Fig. 11 the motion of the extremity of the solid obstacle is plotted. Contrary of the two-dimensional example, small drops of high density fluid are not interacting with the obstacle after the main shock. Therefore, the motion of the flexible structure remains fairly smoothed and it is rather well described with the coarsest grid.

Conclusion

The presented work deals with a partitioned strategy for complex fluid-structure interaction problems in presence of free-surface flows. The proposed coupling

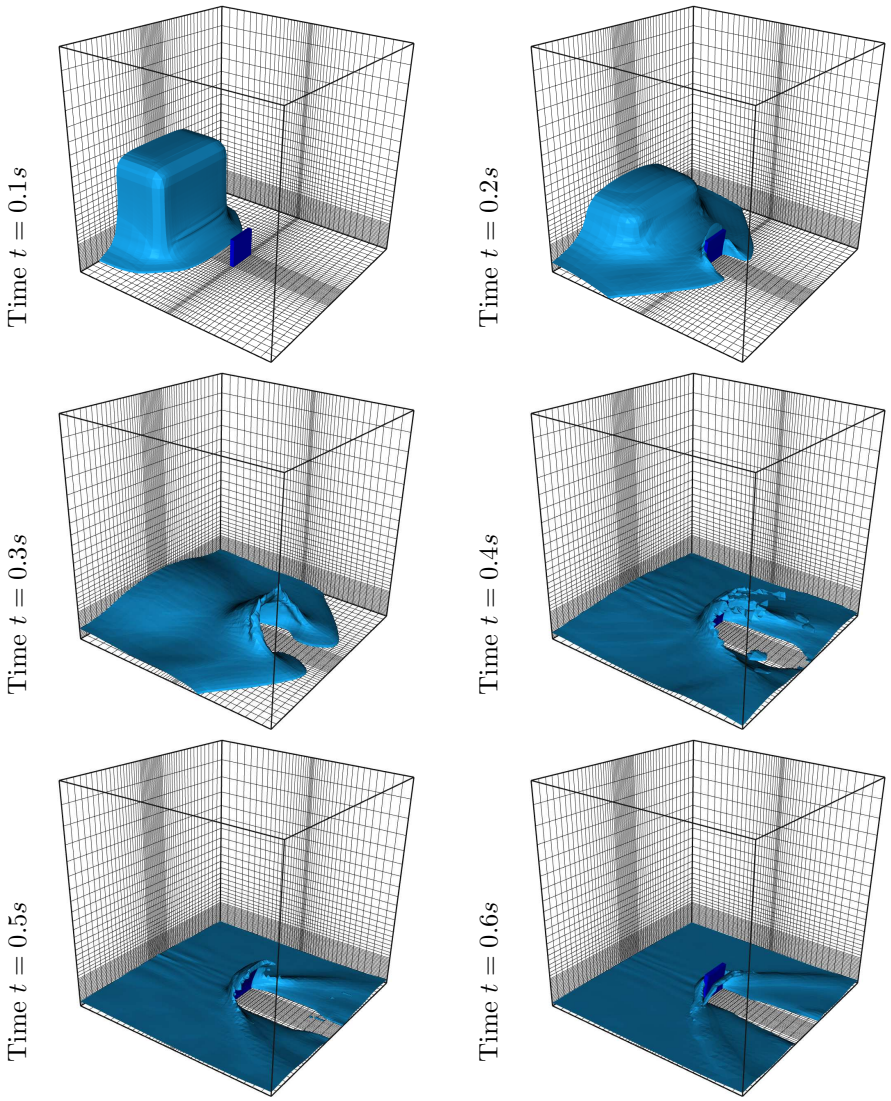


Figure 12: Tri-dimensional dam break problem. Evolution of the free surface and motion of the structure.

strategy is shown to provide reliable (convergent) results with implicit scheme computation for both structure and fluid component. The Direct Force-Motion Transfers algorithm is used to pass the pertinent information between structure and fluid motion, featuring also the implicit solution strategy based upon the fixed-point iterations with a dynamic relaxation parameter used to accelerate convergence [48].

Most importantly, the proposed solution method allows to perform coupled simulations and obtain reliable solution to complex fluid-structure interaction by using the existing codes, that were initially developed to support either fluid or structure motion computation. This is achieved thanks to the use of the component technology [49, 61] providing the coupling between existing software products. Therefore, the proposed solution method for fluid-structure interaction can utilize very different discretization strategies to obtain the optimal accuracy; The case in point concerns FE for the structure and FV for the fluid. The use of these popular methods for the fluid and solid parts allows to benefit from the advanced features of the two families of methods, each developed by the experts from the corresponding domain. Accordingly, on the fluid side, it is possible to use a very efficient semi-implicit solver for incompressible flow (PISO), inverse techniques (Algebraic Multigrid) or advanced models for free-surface flows. A very good performance of the proposed technology for fluid-structure interaction is illustrated with 2D and 3D models for dam breaking examples, which also involve flexible obstacles.

References

[1] J. H. Argyris and D. W. Scharpf. Finite elements in time and space. *Nuclear Engineering and Design*, 10(4):456–464, 1969.

[2] I. E. Barton. Comparison of SIMPLE- and PISO-type algorithms for transient flows. *International Journal for Numerical Methods in Fluid*, 26(4):459–483, 1998.

[3] R. Baudille and M. E. Biancolini. Modelling FSI problems in FLUENT: a dedicated approach by means of UDF programming. In *Proceedings of the European Automotive CFD Conference, Frankfurt, Germany*, 2005.

[4] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5(2):125–134, 2001.

[5] A. Behzadi, R. Issa, and H. Rusche. Modelling of dispersed bubble and droplet flow at high phase fractions. *Chemical Engineering Science*, 59(4):759–770, 2004.

[6] T. Belytschko. An overview of semidiscretization and time integration procedures. In T. Belytschko and T. J. R. Hughes, editors, *Computational methods for transient analysis*, pages 1–65, Amsterdam, North-Holland, 1983. Journal of Applied Mechanics.

[7] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer methods in applied mechanics and engineering*, 139(1-4):3–47, 1996.

[8] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear finite elements for continua and structures*. Wiley, New-York, 2000.

[9] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, pages 199–259, 1990.

[10] G. N. Bullock, C. Obhrai, D. H. Peregrine, and H. Bredmose. Violent breaking wave impacts. Part I: Results from large-scale regular wave tests on vertical and sloping walls. *Coastal Engineering*, 54(8):602–617, 2007.

[11] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4506–4527, 2005.

[12] D. M. Causon, D. M. Ingram, C. G. Mingham, G. Yang, and R. V. Pearson. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in water resources*, 23(5):545–562, 2000.

[13] A. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.

[14] J. Chung and G. M. Hulbert. A family of single-step houbolt time integration algorithms for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2):1–11, 1994.

[15] R. Dalrymple and B. Rogers. Numerical modeling of water waves with the SPH method. *Coastal engineering*, 53(2-3):141–147, 2006.

[16] A. de Boer, A. H. van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1515–1525, 2007.

[17] I. Demirdžić and M. Perić. Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluid*, 8(9), 1988.

[18] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5797–5812, 2006.

[19] W. G. Dettmer and D. Perić. An analysis of the time integration algorithms for the Finite Element solution of incompressible Navier-Stokes equations based on a stabilised formulation. *Computer Methods in Applied Mechanics and Engineering*, 192:1177–1226, 2003.

[20] W. G. Dettmer and D. Perić. A fully implicit computational strategy for strongly coupled fluid-solid interaction. *Archives of Computational Methods in Engineering*, 14:205–247, 2007.

[21] F. Dias, D. Dutykh, and J. Ghidaglia. A two-fluid model for violent aerated flows. *Computers & Fluids*, 39(2):283–293, 2010.

[22] D. Dutykh and D. Mitsotakis. On the relevance of the dam break problem in the context of nonlinear shallow water equations. *Discrete and Continous Dynamical System – Series A*, Accepted, 2009.

[23] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.

[24] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182:499–515, 2000.

[25] C. Felippa and K. Park. Synthesis tools for structural dynamics and partitioned analysis of coupled systems. *NATO Advanced Research Workshop (eds. A. Ibrahimbegović and B. Brank)*, pages 50–111, 2004.

[26] M. Á. Fernández, J.-F. Gerbeau, A. Gloria, and M. Vidrascu. Domain decomposition based Newton methods for fluid-structure interaction problems. In *ESAIM: Proceedings*, volume 22, pages 67–82. edpsciences.org, 2008.

[27] M. Á. Fernández and M. Moubachir. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers and Structures*, 83(2-3):127–142, 2005.

[28] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springler-Verlag, Berlin, Germany, 3$^{\text{rd}}$ edition, 2002.

[29] C. Fochesato, S. Grilli, and F. Dias. Numerical modeling of extreme rogue waves generated by directional energy focusing. *Wave Motion*, 44:395–416, 2007.

[30] C. Förster, W. A. Wall, and E. Ramm. On the geometric conservation law in transient ow calculations on deforming domains. *International Journal for Numerical Methods in Fluid*, 50:1369–1379, 2006.

[31] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1291, 2007.

[32] L. P. Franca, T. J. R. Hughes, and R. Stenberg. Stabilized finite element methods. *Incompressible Computational Fluid Dynamics*, pages 87–107, 1993.

[33] J.-M. Ghidaglia, A. Kumbaro, and G. Le Coq. On the numerical solution to two-fluid models via a cell centered finite volume method. *European Journal of Mechanics/B Fluids*, 20(6):841–867, 2001.

[34] J.-M. Ghidaglia and F. Pascal. The normal flux method at the boundary for multidimensional finite volume approximations in CFD. *European Journal of Mechanics/B Fluids*, 24(1):1–17, 2005.

[35] R. Glowinski. Numerical methods for fluids (Part III). In P. Ciarlet and J. Lions, editors, *Handbook of numerical analysis*, volume 9. Elsevier North-Holland, 2003.

[36] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3), 1977.

[37] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.

[38] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193:2087–2014, 2004.

[39] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian-Eulerian finite element formulation from incompressible viscous flows. In *Interdisciplinary Finite Element Analysis: Proceedings of the US-Japan Seminar Held at Cornell University*, page 179. College of Engineering and School of Civil & Environmental Engineering of Cornell University, 1981.

[40] T. J. R. Hughes, K. S. Pister, and R. L. Taylor. Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17:159–182, 1979.

[41] A. Ibrahimbegovic. *Nonlinear solid mechanics: Theoretical formulations and finite element solution methods*. Springer, 2009.

[42] S. R. Idelsohn, E. Onate, and F. Del Pin. The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering*, 61:964–989, 2004.

[43] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1):40–65, 1986.

[44] R. I. Issa, B. Ahmadi-Befrui, K. R. Beshay, and A. D. Gosman. Solution of the implicitly discretised reacting flow equations by operator-splitting. *Journal of Computational Physics*, 93:388–410, 1991.

[45] H. Jasak. OpenFOAM: Open source CFD in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2), 2009.

[46] H. Jasak and Z. Tuković. Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30(2):1–18, 2007.

[47] C. Kassiotis. Which strategy to move the mesh in the Computational Fluid Dynamic code OpenFOAM? Technical report, WiRe / LMT–Cachan, Germany / France, 2008. http://perso.crans.org/kassiotis/openfoam/movingmesh.pdf.

[48] C. Kassiotis, N. Rainer, A. Ibrahimbegovic, and H. G. Matthies. Partitioned procedure for strongly coupled fluid-structure problems. Part I: implicit algorithm, stability analysis and validation examples. *Computer Methods in Applied Mechanics and Engineering*, Submitted, 2010.

[49] C. Kassiotis, N. Rainer, A. Ibrahimbegovic, and H. G. Matthies. Partitioned procedure for strongly coupled fluid-structure problems. Part II: implementation aspects, re-using existing software as components and nested parallel computations by using ctl. *Computer Methods in Applied Mechanics and Engineering*, Submitted, 2010.

[50] S. Koshizuka, H. Tamako, and Y. Oka. A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal*, 4(1):29–46, 1995.

[51] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.

[52] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2):269–289, 1974.

[53] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3039–3067, 2001.

[54] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134(1-2):71–90, 1996.

[55] R. J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *Society for Industrial and Applied Mathematics Journal on Numerical Analysis*, 33(2):627–665, 1996.

[56] Q. Ma and S. Yan. QALE-FEM for numerical modelling of non-linear interaction between 3D moored floating bodies and steep waves. *International Journal for Numerical Methods in Engineering*, 78(6):713–756, 2009.

[57] D. Markovič, R. Niekamp, A. Ibrahimbegovic, H. G. Matthies, and R. L. Taylor. Multi-scale modeling of heterogeneous structures with inelastic constitutive behavior : Mathematical and physical aspects. *International Journal of Engineering Computations*, 22:664–683, 2005.

[58] H. G. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*, 195:2028–2049, 2006.

[59] H. G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid-structure interaction. *Computers and Structures*, 81:805–812, 2003.

[60] J. Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30(1):543–574, 1992.

[61] R. Niekamp, D. Markovič, A. Ibrahimbegovic, H. G. Matthies, and R. L. Taylor. Multi-scale modelling of heterogeneous structures with inelastic constitutive behavior: Part II–software coupling implementation aspects. *Engineering Computations*, 26:6–28, 2009.

[62] F. Pascal and J.-M. Ghidaglia. Footbridges between finite volumes and finite elements with applications to CFD. *International Journal for Numerical Methods in Fluid*, 37:951–986, 2001.

[63] S. Patankar and D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787–1806, 1972.

[64] S. V. Patankar. *Numerical heat transfer and fluid flow.* Hemisphere Publishing Corporation, Washington, DC, 1980.

[65] D. Perić, W. G. Dettmer, and P. H. Saksono. Modelling fluid-induced structural vibrations: reducing the structural risk for stormy-winds. In A. Ibrahimbegovic, editor, *NATO Advanced Research Workshop*, ARW 981641, pages 239–268, Opatija, Croatia, 2006.

[66] S. Piperno, M. Remaki, and L. Fezoui. A nondiffusive finite volume scheme for the three-dimensional Maxwell's equations on unstructured meshes. *Society for Industrial and Applied Mathematics Journal on Numerical Analysis*, 39(6):2089–2108, 2002.

[67] S. B. Pope. *Turbulent flows.* Cambridge University Press, United-Kingdom, 2000.

[68] S. Rogers, D. Kwak, and C. Kiris. Steady and unsteady solutions of the incompressible Navier-Stokes equations. *American Institute of Aeronautics and Astronautics Journal*, 29(4):603–610, 1991.

[69] A. K. Slone, C. Bailey, and M. Cross. Dynamic solid mechanics using finite volume methods. *Applied Mathematical Modelling*, 27(2):69–87, 2003.

[70] R. J. Sobey. Linear and Nonlinear Wave Theory. Lecture note, Leichtweiß-Institut für Wasserbau, Braunschweig, 1998.

[71] J. J. Stoker. *Water Waves: The Mathematical Theory and Applications.* Wiley-Interscience, New-York, 1992.

[72] O. Ubbink and R. Issa. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics*, 153(1):26–50, 1999.

[73] J. Van Doormaal and G. Raithby. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numerical Heat Transfer, Part A: Applications*, 7(2):147–163, 1984.

[74] J. Vila. On particle weighted methods and smooth particle hydrodynamics. *Mathematical models and methods in applied sciences*, 9(2):161–210, 1999.

[75] D. Violeau and R. Issa. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. *International Journal for Numerical Methods in Fluids*, 53(2):277–304, 2006.

[76] E. Walhorn, A. Kölke, B. Hübner, and D. Dinkler. Fluid-structure coupling within a monolithic model involving free surface flows. *Computers and Structures*, 83(25-26):2100–2111, 2005.

[77] W. A. Wall, D. P. Mok, and E. Ramm. Partitioned analysis approach of the transient coupled response of viscous fluids and flexible structures. In *Solids, structures and coupled problems in engineering, proceedings of the European Conference on Computational Mechanics*, 1999.

[78] H. Wang, J. Chessa, W. K. Liu, and T. Belytschko. The immersed–fictitious element method for fluid-structure interaction: Volumetric consistency, compressibility and thin members. *International Journal for Numerical Methods in Engineering*, 74(1), 2008.

[79] H. G. Weller, G. Tabora, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998.

[80] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, Fluid Mechanics*, volume 3. Butterworth Heinemann, Oxford, 5^{th} edition, 2001.

[81] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, Solid Mechanics*, volume 2. Butterworth Heinemann, Oxford, 5^{th} edition, 2001.

[82] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, The Basis*, volume 1. Butterworth Heinemann, Oxford, 5^{th} edition, 2001.