INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2011; **67**:1004–1023
Published online 3 September 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.2403

# The Fixed-Mesh ALE approach for the numerical simulation of floating solids

Joan Baiges*,†, Ramon Codina and Herbert Coppola-Owen

*Technical University of Catalonia, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain*

SUMMARY

In this paper, we propose a method to solve the problem of floating solids using always a background mesh for the spatial discretization of the fluid domain. The main feature of the method is that it properly accounts for the advection of information as the domain boundary evolves. To achieve this, we use an arbitrary Lagrangian–Eulerian framework, the distinctive characteristic being that at each time step results are projected onto a fixed, background mesh. We pay special attention to the tracking of the various interfaces and their intersections, and to the approximate imposition of coupling conditions between the solid and the fluid. Copyright © 2010 John Wiley & Sons, Ltd.

Received 23 July 2009; Revised 9 July 2010; Accepted 17 July 2010

KEY WORDS: fixed mesh; free surface; floating solid; immersed boundary methods; ALE strategy; fluid–structure interaction

## 1. INTRODUCTION

In the classical Arbitrary Eulerian Lagrangian (ALE) approach to solve CFD problems, the mesh in which the computational domain is discretized is deformed. This is done according to a prescribed motion of part of its boundary, which is transmitted to the interior nodes in a way as smooth as possible so as to avoid mesh distortion. In this work, we present an ALE-type strategy with a different motivation. In the Fixed Mesh-ALE (FM-ALE) method, instead of assuming that the computational domain is defined by the mesh boundary, we assume that there is a boundary function that defines where the flow takes place. When this boundary function moves, the flow domain changes, and that must be taken into account at the moment of writing the conservation equations that govern the flow, which need to be cast in the ALE format. However, our purpose here is to explain how to use always a background fixed mesh. That requires a virtual motion of the mesh nodes followed by a projection of the new node positions onto the fixed mesh.

The FM-ALE method for flow problems in moving domains was originally proposed in [1] and it is extensively described in [2]. In [3] it is applied to free surface problems, and it is extended to solid mechanics and fluid–structure interaction problems in [4]. Here we apply it to the problem of simulating a floating rigid body in a liquid. In this case, the boundary function is defined by the boundary of the solid, and its motion is determined by the dynamics of this solid. In turn, this dynamical behavior is governed by the action exerted by the fluid on the rigid body surface. Apart from the problem of dealing with a moving computational domain, an additional difficulty is the presence of the free surface of the fluid on which the solid floats. This free surface may be treated

*Correspondence to: Joan Baiges, Technical University of Catalonia, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.
†E-mail: jbaiges@cimne.upc.edu

as such or as an interface with a lighter fluid, for example air. In this work we adopt the former approach, using an approximate imposition of boundary conditions at the intersection between the fluid surface and the elements of the finite element mesh.

Several works have been already developed in the field of the simulation of floating solids. In [5], floating bodies are simulated by means of the quasi arbitrary Lagrangian–Eulerian finite element method (QALE-FEM method) by means of a moving mesh, but the solid displacements in the numerical examples are small and there is no need to remesh. In [6], the FEM is used to simulate the interaction between waves and a floating body, but again it focuses in the case in which the solid body displacements are small, and the domain movement can be treated by moving the finite element mesh and using an ALE framework. A fixed grid strategy for the simulation of solids falling into water, which uses the cut-cell technique for fixed grids, has been used in [7], where the impact of a cylindrical object on a water surface is studied, the main difference with respect to the present approach being the way newly created nodes are treated. Floating bodies can also be treated with the Chimera strategy described in [8], provided the free surface is considered as the interface between the fluid analyzed and a fictitious one, for example air. The flow problem would become in this case a two-phase flow rather than a free surface problem, and a possible way to deal with it is explained in [9]. ALE approaches are also possible for the simulation of free surface, fluid–structure interaction problems, as done for example in [10], but they require rebuilding the finite element mesh when this mesh gets too distorted. In [11], the phase-field method is used to analyze the wetting phenomena of the impact of a sphere with a free surface.

The novelty of the present work with respect to the previous ones is the use of fixed-mesh strategy, which correctly takes into account the movement of the fluid domain at the time of computing the ALE convective terms and time derivatives. We want to stress that this idea is independent of the way to impose boundary conditions on the moving boundary. The way to impose this prescription is often used to classify a particular fixed-mesh method. Since the physical boundary is contained in the domain actually discretized, these methods are often called *immersed boundary methods*. Moreover, since the fixed grid used is often Cartesian, these formulations can be found under the keywords *Cartesian grid methods* (see for example the reviews [12–14]). These methods are developed for constant-in-time domains, and then extended in a more or less *ad hoc* way to time-dependent domains. In spite of the fact that we want to distinguish between the way to deal with moving domains and the way of approximately imposing the boundary conditions on the moving boundary, we will briefly describe the particular approach we use.

The paper is organized as follows. We present the general framework in Section 2, where we describe the fluid and solid mechanics problems that have to be solved in a domain that evolves in time, and also the level set approach we adopt for tracking the free surface. In Section 3, we describe the numerical approximation we adopt for solving each of the subproblems involved in the computations, including the spatial and time discretization, and the stabilization methods used to deal with the fluid mechanics problems. Section 4 introduces the FM-ALE algorithm, and the basic algorithmic steps are briefly schematized. We pay special attention to the features that are particular to the problem of floating solids: approximate imposition of boundary conditions and the interaction between the solid and the free surface boundaries. Section 5 corresponds to the final algorithm of the method, and Section 6 presents a numerical example in which we show the behavior of the proposed methodology. Some conclusions close the paper in Section 7.

## 2. PROBLEM STATEMENT

Let us consider a region $\Omega^0 \subset \mathbb{R}^d$ $(d=2,3)$ where a flow will take place during a time interval $[0,T]$. However, we consider the case in which the fluid at time $t$ occupies only a subdomain $\Omega(t) \subset \Omega^0$ (note in particular that $\Omega(0) \subset \Omega^0$). Suppose also that the boundary of $\Omega(t)$ is defined by part of $\partial\Omega^0$ and a moving boundary that we call $\Gamma_f(t) = \partial\Omega(t) \setminus \partial\Omega^0 \cap \partial\Omega(t)$. This moving part



DOI: 10.1002/fld

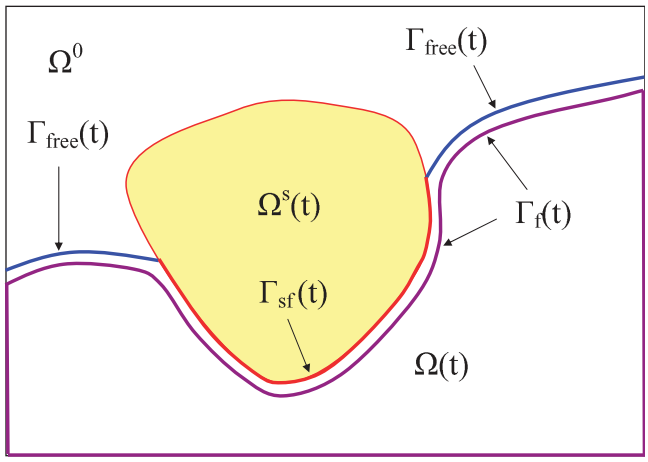


Figure 1. Setting.

of $\partial\Omega(t)$ may correspond to the boundary of a moving solid immersed in the fluid or can be determined by a level set function. The setting of the problem is described in Figure 1, where also the solid domain $\Omega^s(t)$ and the fluid–structure interface $\Gamma_{sf}$ have been depicted.

In order to cope with the time-dependency of $\Omega(t)$, we use the ALE approach. The incompressible Navier–Stokes formulated in $\Omega(t)$, accounting also for the motion of this domain, can be written as follows: find a velocity $\boldsymbol{u}:\Omega(t)\times(0,T)\longrightarrow\mathbb{R}^d$ and a pressure $p:\Omega(t)\times(0,T)\longrightarrow\mathbb{R}$ such that

$$\rho\left[\frac{\partial\boldsymbol{u}}{\partial t}+(\boldsymbol{u}-\boldsymbol{u}_{\text{dom}})\cdot\nabla\boldsymbol{u}\right]-\nabla\cdot(2\mu\nabla^{\text{S}}\boldsymbol{u})+\nabla p=\rho\boldsymbol{f}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u}=0, \tag{2}$$

where $\nabla^{\text{S}}\boldsymbol{u}$ is the symmetrical part of the velocity gradient, $\rho$ is the fluid density, $\mu$ is the viscosity, $\boldsymbol{f}$ is the vector of body forces and $\boldsymbol{u}_{\text{dom}}$ is the domain velocity.

Boundary conditions are of the form

$$\boldsymbol{u}=\bar{\boldsymbol{u}}\quad\text{on }\Gamma_{\text{D}},$$

$$\boldsymbol{n}\cdot\boldsymbol{\sigma}=\bar{\boldsymbol{t}}\quad\text{on }\Gamma_{\text{N}},$$

where $\boldsymbol{n}$ is the external normal to the boundary, $\boldsymbol{\sigma}=-p\boldsymbol{I}+2\mu\nabla^{\text{S}}\boldsymbol{u}$ is the Cauchy stress tensor and $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{t}}$ are the given boundary data.

When dealing with the fluid part of the domain, we also have to take into account the movement of the free surface. This movement is dealt with by means of a level set function, as explained for example in [3, 15].

In the level set method we define a smooth function $\Psi$ over $\Omega^0$, which allows us to determine $\Omega(t)$. In our case we define $\Omega(t)$ as the region over which $\Psi(\boldsymbol{x},t)$ is positive. The position of the fluid front will be defined by the iso-value contour $\Psi=0$. The evolution of this level set function is computed by means of the transport equation: find $\Psi:\Omega^0\times(0,T)\longrightarrow\mathbb{R}$ such that

$$\frac{\partial\Psi}{\partial t}+\boldsymbol{u}\cdot\nabla\Psi=0, \tag{3}$$

with the additional requirement that the advection velocity at the free surface coincides with that of the fluid. The procedure used to compute the level set advection velocity is described in Section 4.2.3.

For the solid part, we consider only the case of rigid bodies. The solid also evolves in time, and we denote its domain by $\Omega_s(t)$. As usual in solid mechanics problems, we face the problem



DOI: 10.1002/fld

in a purely Lagrangian way. Let us denote by $\boldsymbol{x}_{\mathrm{rb}}$ the position vector of the center of mass of the rigid body, and by $\boldsymbol{\theta}_{\mathrm{rb}}$ the Euler angles. The motion equations for the rigid body are: find $\boldsymbol{x}_{\mathrm{rb}}:(0,T)\longrightarrow\mathbb{R}^d$ and $\boldsymbol{\theta}_{\mathrm{rb}}:(0,T)\longrightarrow\mathbb{R}^d$ such that

$$m\,\frac{\mathrm{d}^2\boldsymbol{x}_{\mathrm{rb}}}{\mathrm{d}t^2}=\boldsymbol{F}, \tag{4}$$

$$\boldsymbol{I}\,\frac{\mathrm{d}^2\boldsymbol{\theta}_{\mathrm{rb}}}{\mathrm{d}t^2}=\boldsymbol{T}, \tag{5}$$

where $m$ is the mass of the rigid body, $\boldsymbol{I}$ is the inertia tensor that we have considered to be constant in time for simplicity, $\boldsymbol{F}$ is the force vector at the center of mass and $\boldsymbol{T}$ is the torque at the center of mass.

Initial and boundary conditions have to be appended to problem (1)–(2) and initial conditions to (4)–(5). In order to impose these conditions, we redefine $\Gamma_{\mathrm{f}}(t)$ as $\Gamma_{\mathrm{f}}(t)=\Gamma_{\mathrm{free}}(t)\cup\Gamma_{\mathrm{sf}}(t)$, where $\Gamma_{\mathrm{free}}(t)$ is the part of $\Gamma_{\mathrm{f}}(t)$ corresponding to the free surface and $\Gamma_{\mathrm{sf}}(t)$ is the part of $\Gamma_{\mathrm{f}}(t)$ corresponding to the interaction between the fluid and the structure.

In $\Gamma_{\mathrm{free}}$ boundary conditions are of Neumann type, specifically we prescribe tractions to zero, neglecting surface tension. In $\Gamma_{\mathrm{sf}}$ we must impose the usual conditions in fluid–structure interaction problems, which for rigid bodies are continuity of the velocity field and transmission of the forces and torques exerted on the solid body by the fluid.

On the rest of the boundary of $\Omega(t)$, the usual Dirichlet and Neumann boundary conditions can be considered.

## 3. NUMERICAL APPROXIMATION

### *3.1. The time-discrete problem*

Let us start introducing some notation. Consider a uniform partition of $[0,T]$ into $N$ time intervals of length $\delta t$. Let us denote by $f^n$ the approximation of a time-dependent function $f$ at time level $t^n=n\delta t$. We will also denote

$$\delta f^{n+1}=f^{n+1}-f^n,$$

$$\delta_t f^{n+1}=\frac{f^{n+1}-f^n}{\delta t},$$

$$f^{n+\theta}=\theta f^{n+1}+(1-\theta)f^n,\quad \theta\in[\tfrac{1}{2},1].$$

We will use the trapezoidal rule to discretize problem (1)–(2) in time. Suppose we are given a computational domain $\Omega(t^n)$ at time $t^n$, with spatial coordinates labeled $\boldsymbol{x}^n$, and $\boldsymbol{u}^n$ and $p^n$ are known in this domain. The velocity $\boldsymbol{u}^{n+1}$ and the pressure $p^{n+1}$ in the domain $\Omega(t^{n+\theta})$ can then be found as the solution to the problem

$$\rho[\delta_t\boldsymbol{u}^{n+1}|_{\boldsymbol{x}^n}+(\boldsymbol{u}^{n+\theta}-\boldsymbol{u}^{n+\theta}_{\mathrm{dom}})\cdot\nabla\boldsymbol{u}^{n+\theta}]-\nabla\cdot(2\mu\nabla^{\mathrm{S}}\boldsymbol{u}^{n+\theta})+\nabla p^{n+1}=\rho\boldsymbol{f}^{n+\theta}, \tag{6}$$

$$\nabla\cdot\boldsymbol{u}^{n+\theta}=0, \tag{7}$$

where now $\delta_t\boldsymbol{u}^{n+1}|_{\boldsymbol{x}^n}=(\boldsymbol{u}^{n+1}(\boldsymbol{x})-\boldsymbol{u}^n(\boldsymbol{x}^n))/\delta t$, $\boldsymbol{x}$ being the spatial coordinates in $\Omega(t^{n+\theta})$. We are interested only in the cases $\theta=\frac{1}{2}$ and $\theta=1$ (implicit schemes are required). The same simple trapezoidal rule is used to temporarily discretize the equation corresponding to the advection of the level set function.



DOI: 10.1002/fld

For the temporal integration of the solid mechanics problem, we consider Newmark's method:

$$
\begin{aligned}
m\ddot{\boldsymbol{x}}_{\mathrm{rb}}^{n+1} &= \boldsymbol{F}^{n+1},\\
\ddot{\boldsymbol{x}}_{\mathrm{rb}}^{n+1} &= \frac{1}{\beta\delta t^2}[\boldsymbol{x}_{\mathrm{rb}}^{n+1}-\boldsymbol{x}_{\mathrm{rb}}^{n}-\dot{\boldsymbol{x}}_{\mathrm{rb}}^{n}\delta t]-\left(\frac{1}{2\beta}-1\right)\ddot{\boldsymbol{x}}_{\mathrm{rb}}^{n},\\
\dot{\boldsymbol{x}}_{\mathrm{rb}}^{n+1} &= \frac{\gamma}{\beta\delta t}[\boldsymbol{x}_{\mathrm{rb}}^{n+1}-\boldsymbol{x}_{\mathrm{rb}}^{n}]+\left(1-\frac{1}{2\beta}\right)\delta t\ddot{\boldsymbol{x}}_{\mathrm{rb}}^{n},
\end{aligned}
\tag{8}
$$

and

$$
\begin{aligned}
\boldsymbol{I}\ddot{\boldsymbol{\theta}}_{\mathrm{rb}}^{n+1} &= \boldsymbol{T}^{n+1},\\
\ddot{\boldsymbol{\theta}}_{\mathrm{rb}}^{n+1} &= \frac{1}{\beta\delta t^2}[\boldsymbol{\theta}_{\mathrm{rb}}^{n+1}-\boldsymbol{\theta}_{\mathrm{rb}}^{n}-\dot{\boldsymbol{\theta}}_{\mathrm{rb}}^{n}\delta t]-\left(\frac{1}{2\beta}-1\right)\ddot{\boldsymbol{\theta}}_{\mathrm{rb}}^{n},\\
\dot{\boldsymbol{\theta}}_{\mathrm{rb}}^{n+1} &= \frac{\gamma}{\beta\delta t}[\boldsymbol{\theta}_{\mathrm{rb}}^{n+1}-\boldsymbol{\theta}_{\mathrm{rb}}^{n}]+\left(1-\frac{1}{2\beta}\right)\delta t\ddot{\boldsymbol{\theta}}_{\mathrm{rb}}^{n},
\end{aligned}
\tag{9}
$$

where $\beta$ and $\gamma$ are parameters to be chosen. Most usual values are $\beta=\frac{1}{4}$ and $\gamma=\frac{1}{2}$, which provide a second-order stable and non-dissipative scheme.

If $\theta\neq 1$ in the trapezoidal rule in (6)–(7), results for the solid body unknowns at $t^{n+\theta}$ must be interpolated, velocity continuity is imposed at $t^{n+\theta}$. Another possibility would be to transport the Navier–Stokes equations from $\Omega(t^{n+\theta})$ to $\Omega(t^{n+1})$, or to use a second-order backward difference scheme for the time integration of the Navier–Stokes equations, in which all the terms are defined at $\Omega(t^{n+1})$. Some of these possibilities are discussed in [16].

### *3.2. The fully discrete problem*

The next step is to consider the spatial discretization of problem (6)–(7) (there is no need to spatially discretize problem (8)–(9) since we are considering rigid bodies). As for the time discretization, different options are possible. Here we simply describe the stabilized finite element formulation employed in our numerical simulations.

Let $\{\Omega^e\}^{n+1}$ be a finite element partition of the domain $\Omega(t^{n+1})$, with index $e$ ranging from 1 to the number of elements $n_{\mathrm{el}}$ (which may be different at different time steps). We denote with a subscript $h$ the finite element approximation to the unknown functions, and by $\boldsymbol{v}_h$ and $q_h$ the velocity and pressure test functions associated with $\{\Omega^e\}^{n+1}$, respectively.

An important point is that we are interested in using equal interpolation for the velocity and the pressure. This allows as to use the same elements both for the pressure and the velocity unknowns, which results in a simpler implementation of the overall numerical problem. Therefore, the corresponding finite element spaces are assumed to be built up using the standard continuous interpolation functions.

In order to overcome the numerical problems of the standard Galerkin method, a stabilized finite element formulation is applied. This formulation is presented in [17]. It is based on the subgrid-scale concept introduced in [18], which reduces to the Galerkin/least-squares method [19] when linear elements are used. We apply this stabilized formulation together with the finite difference approximation in time (6)–(7).

The bottom line of the method is to test the continuous equations by the standard Galerkin test functions plus perturbations that depend on the operator representing the differential equation being solved. In our case, this operator corresponds to the linearized form of the time-discrete Navier–Stokes equations (6)–(7). In this case, the method consists of finding $\boldsymbol{u}_h^{n+1}$ and $p_h^{n+1}$ such that

$$
m_1^{n+\theta}(\delta_t\boldsymbol{u}_h^{n+1}|_{\boldsymbol{x}^n},\boldsymbol{v}_h)+a^{n+\theta}(\boldsymbol{u}_h,\boldsymbol{v}_h)+c^{n+\theta}(\boldsymbol{u}_h-\boldsymbol{u}_{\mathrm{dom}};\boldsymbol{u}_h,\boldsymbol{v}_h)+b_1^{n+\theta}(p_h,\boldsymbol{v}_h)=l_1^{n+\theta}(\boldsymbol{v}_h), \tag{10}
$$

$$
m_2^{n+\theta}(q_h,\delta_t\boldsymbol{u}_h^{n+1}|_{\boldsymbol{x}^n})+b_2^{n+\theta}(q_h,\boldsymbol{u}_h)+s^{n+\theta}(q_h,p_h)=l_2^{n+\theta}(q_h), \tag{11}
$$



DOI: 10.1002/fld

for all test functions $\boldsymbol{v}_h$ and $q_h$, with the former vanishing on the Dirichlet part of the boundary $\Gamma_{\mathrm{D}}$. The different forms appearing in these equations are given by

$$m_1(\delta_t \boldsymbol{u}_h, \boldsymbol{v}_h) = \int_\Omega \boldsymbol{v}_h \cdot \rho \delta_t \boldsymbol{u}_h \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_{u1} \cdot \rho \delta_t \boldsymbol{u}_h \, \mathrm{d}\Omega,$$

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h) = \int_\Omega 2\nabla^{\mathrm{S}} \boldsymbol{v}_h : \mu \nabla^{\mathrm{S}} \boldsymbol{u}_h \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_{u1} \cdot (-2\nabla \cdot (\mu \nabla^{\mathrm{S}} \boldsymbol{u}_h)) \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \zeta_{u2} \nabla \cdot \boldsymbol{u}_h \, \mathrm{d}\Omega,$$

$$c(\boldsymbol{a}; \boldsymbol{u}_h, \boldsymbol{v}_h) = \int_\Omega \boldsymbol{v}_h \cdot (\rho \boldsymbol{a} \cdot \nabla \boldsymbol{u}_h) \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_{u1} \cdot (\rho \boldsymbol{a} \cdot \nabla \boldsymbol{u}_h) \, \mathrm{d}\Omega,$$

$$b_1(p_h, \boldsymbol{v}_h) = -\int_\Omega p_h \nabla \cdot \boldsymbol{v}_h \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_{u1} \cdot \nabla p_h \, \mathrm{d}\Omega,$$

$$m_2(q_h, \delta_t \boldsymbol{u}_h) = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_p \cdot \rho \delta_t \boldsymbol{u}_h \, \mathrm{d}\Omega,$$

$$b_2(q_h, \boldsymbol{u}_h) = \int_\Omega q_h \nabla \cdot \boldsymbol{u}_h \, \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_p \cdot (\rho \boldsymbol{a} \cdot \nabla \boldsymbol{u}_h - 2\nabla \cdot (\mu \nabla^{\mathrm{S}} \boldsymbol{u}_h)) \, \mathrm{d}\Omega,$$

$$s(q_h, p_h) = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_p \cdot \nabla p_h \, \mathrm{d}\Omega,$$

$$l_1(\boldsymbol{v}_h) = \int_\Omega \boldsymbol{v}_h \rho \cdot \boldsymbol{f} \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_{u1} \cdot \boldsymbol{f} \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{N}}} \boldsymbol{v}_h \cdot \bar{\boldsymbol{t}},$$

$$l_2(q_h) = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{\zeta}_p \rho \cdot \boldsymbol{f} \mathrm{d}\Omega,$$

where the functions $\boldsymbol{\zeta}_{u1}$, $\zeta_{u2}$ and $\boldsymbol{\zeta}_p$ are computed within each element as

$$\boldsymbol{\zeta}_{u1} = \tau_u [\rho(\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{dom}}) \cdot \nabla \boldsymbol{v}_h + 2\nabla \cdot (\mu \nabla^{\mathrm{S}} \boldsymbol{v}_h)], \tag{12}$$

$$\zeta_{u2} = \tau_p \nabla \cdot \boldsymbol{v}_h, \tag{13}$$

$$\boldsymbol{\zeta}_p = \tau_u \nabla q_h, \tag{14}$$

and the parameters $\tau_u$ and $\tau_p$ are also computed element-wise as [20]

$$\tau_u = \left[ \frac{4\mu}{h^2} + \frac{2\rho |\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{dom}}|}{h} \right]^{-1}, \quad \tau_p = 4\mu + 2\rho |\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{dom}}| h,$$

where $h$ is the element size for linear elements and half of it for quadratics.

A similar formulation is applied to solve the advection of the level set function.

## 4. THE FM-ALE METHOD

In this section we describe the general FM-ALE method, which we will apply to solve the fluid part of the problem. See [2] for a more detailed explanation of the FM-ALE method in the general framework of moving domains, and [4] for the application of the FM-ALE method to solid mechanics and FSI problems.



DOI: 10.1002/fld

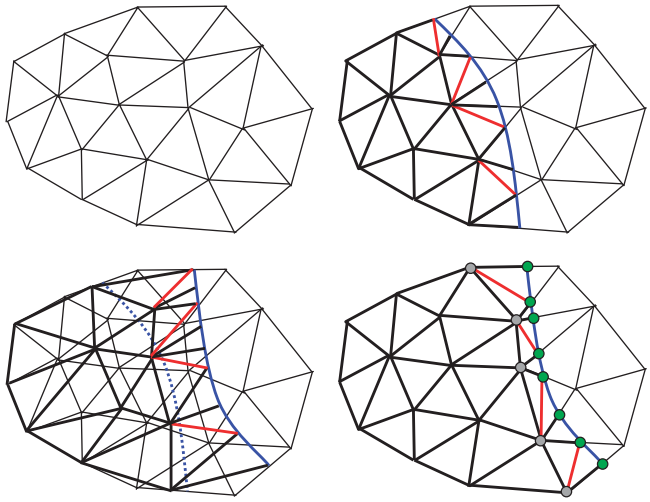

Figure 2. Two-dimensional FM-ALE schematic. Top-left: original finite element mesh $M^0$ of $\Omega^0$. Top-right: finite element mesh $M^n$ of $\Omega(t^n)$, with the elements represented by a thick line and the elements of $M^0$ represented by thin line. Elements cut by the interface are split for integration purposes. Bottom-left: updating of $M^n$ to $M^{n+1}_{\mathrm{ALE}}$ using the classical ALE strategy. The previous position of $\Gamma^n_f$ is depicted using a dotted line. Bottom-right: Mesh $M^{n+1}$ of $\Omega(t^{n+1})$, represented by a thick line. Elements intersected by the boundary function are again split for integration purposes.

In this section and the ones that follow it, the numerical schemes will be particularized for $\theta$=1.

### 4.1. *The general algorithm*

Suppose that $\Omega^0$ is meshed with a finite element mesh $M^0$ and that at the time level $t^n$, the domain $\Omega(t^n)$ is meshed with a finite element mesh $M^n$. Let $\boldsymbol{u}^n_h$ be the velocity already computed on $\Omega(t^n)$. The purpose is to obtain the region the solid occupies at time $t^{n+1}$, $\Omega(t^{n+1})$, and to compute the various unknown fields. If the classical ALE method is used, $M^n$ would deform to another mesh defined at $t^{n+1}$. In the FM-ALE approach we do not use this mesh to compute the unknowns of the problem, but instead we re-mesh in such a way that the new mesh is, essentially, $M^0$ once again.

The steps of the algorithm to achieve the goal described are the following:

1. Define $\Gamma^{n+1}_{\mathrm{f}}$ by updating the function that defines it.
2. Deform *virtually* the mesh $M^n$ to $M^{n+1}_{\mathrm{ALE}}$ using the classical ALE concepts and compute the mesh velocity $\boldsymbol{u}^{n+1}_{\mathrm{dom}}$.
3. Write down the ALE solid mechanics equations on $M^{n+1}_{\mathrm{ALE}}$.
4. *Split the elements* of $M^0$ cut by $\Gamma^{n+1}_{\mathrm{f}}$ to define a mesh on $\Omega(t^{n+1})$, $M^{n+1}$.
5. *Project* the ALE solid mechanics equations from $M^{n+1}_{\mathrm{ALE}}$ to $M^{n+1}$.
6. Solve the equations on $M^{n+1}$ to compute the unknowns.

A global idea of the meshes involved in the process is represented in Figure 2. Note in particular that at each time step, two sets of nodes have to be appropriately dealt with, namely, the newly created nodes and the boundary nodes. Contrary to other fixed grid methods, newly created nodes are treated in a completely natural way using the FM-ALE approach: the value of the velocity there is directly given by the projection step from $M^{n+1}_{\mathrm{ALE}}$ to $M^{n+1}$. Boundary nodes are actually not used. Instead, boundary conditions are imposed approximately to avoid the complexity of adding nodes that do not belong to the background mesh $M^0$.



DOI: 10.1002/fld

### 4.2. *Details on some of the steps*

*4.2.1. Splitting of elements.* Mesh $M^{n+1}$ is obtained by splitting the elements of $M^0$ cut by $\Gamma_{\mathrm{f}}^{n+1}$. Meshes $M^{n+1}$ and $M^0$ only differ in the subelements created after the splitting just mentioned. Mesh $M^{n+1}$ could be thought as a local refinement of mesh $M^0$ to make it conform the boundary $\Gamma_{\mathrm{f}}^{n+1}$. As in other fixed grid methods, this computational complication can be avoided by prescribing boundary conditions on $\Gamma_{\mathrm{f}}^{n+1}$ in an approximate way. Nevertheless, the local refinement from $M^0$ to $M^{n+1}$ is retained to perform the numerical integration of the different terms appearing in (10)–(11).

However, depending on how $\Gamma_{\mathrm{f}}^{n+1}$ intersects $M^0$, the resulting subelements size could be very small compared with the size of elements adjacent to $\Gamma_{\mathrm{f}}^{n+1}$. This results in an ill-conditioning of the system of equations to be solved. In order to avoid this issue, we work with a slightly deformed mesh at each time step constructed as follows: exterior nodes very close to $\Gamma_{\mathrm{free}}^{n+1}$ (closer than $0.1h$ for example) are displaced in a direction orthogonal to $\Gamma_{\mathrm{f}}^{n+1}$ until they match exactly the body surface. The splitting of this mesh will avoid ill-conditioned elements. A more detailed explanation of this procedure can be found in [4].

*4.2.2. Approximate imposition of boundary conditions.* As we have already mentioned, it is very convenient from the implementation standpoint to prescribe boundary conditions approximately. We summarize next a strategy to prescribe Dirichlet boundary conditions on a generic immersed boundary, that we denote by $\Gamma$. This strategy provides optimal order of accuracy and proves to be suitable for both flow and fluid–structure interaction problems. See [21] for more details.

Let $u_h$ be the unknown solution of a problem posed in $\Omega \subset \Omega^0$ for which we want to prescribe a condition on $\Gamma$. Let $\Omega_\Gamma$ be the set of elements cut by $\Gamma$, which is split as $\Omega_\Gamma = \Omega_{\Gamma,\mathrm{in}} \cup \Omega_{\Gamma,\mathrm{out}}$ , where $\Omega_{\Gamma,\mathrm{in}} = \Omega \cap \Omega_\Gamma$ and $\Omega_{\Gamma,\mathrm{out}}$ is the interior of $\Omega_\Gamma \setminus \Omega_{\Gamma,\mathrm{in}}$. Note that $\Omega = \Omega_{\mathrm{in}} \cup \Omega_{\Gamma,\mathrm{in}}$. For simplicity, we will assume that the intersection of $\Gamma$ with the element domains can be exactly represented by the classical isoparametric mapping. For the notation to be used, see Figure 3.

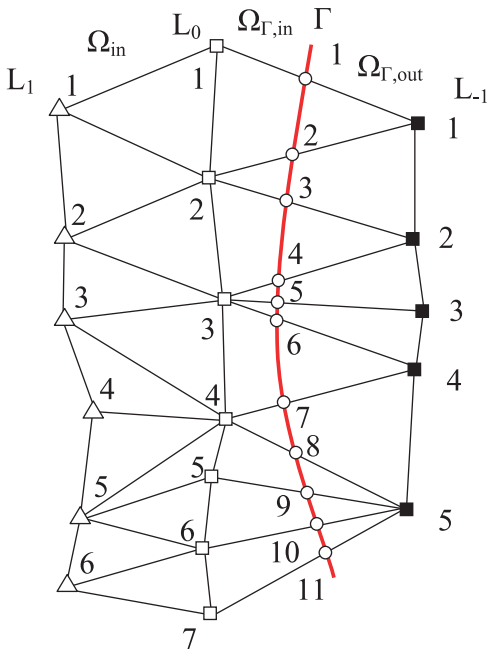


Figure 3. Immersed boundary sketch in a 2D example. The subdomain to the left of the $\Gamma$ boundary corresponds to the fluid, whereas the subdomain to the right corresponds to the solid.



DOI: 10.1002/fld

Suppose that the unknown $u_h$ is interpolated as

$$u_h(\boldsymbol{x}) = \sum_{a=1}^{n_{\text{in}}} I^a_{\text{in}}(\boldsymbol{x}) U^a_{\text{in}} + \sum_{b=1}^{n_{\text{out}}} I^b_{\text{out}}(\boldsymbol{x}) U^b_{\text{out}}$$
$$= \boldsymbol{I}_{\text{in}}(\boldsymbol{x})\boldsymbol{U}_{\text{in}} + \boldsymbol{I}_{\text{out}}(\boldsymbol{x})\boldsymbol{U}_{\text{out}},$$

where $I^a_{\text{in}}(\boldsymbol{x})$ and $I^b_{\text{out}}(\boldsymbol{x})$ are the standard interpolation functions, $n_{\text{in}}$ is the number of nodes in $\Omega_{\text{in}}$, the domain where the problem needs to be solved (including layer $L_0$) and $n_{\text{out}}$ the number of nodes in layer $L_{-1}$ (see Figure 3).

The objective is to compute $\boldsymbol{U}_{\text{out}}$. Suppose that $u_h$ needs to be prescribed to a given function $\bar{u}$ on $\Gamma$. The main idea is to compute $\boldsymbol{U}_{\text{out}}$ by minimizing the functional

$$J_2(\boldsymbol{U}_{\text{in}}, \boldsymbol{U}_{\text{out}}) = \int_\Gamma (u_h(\boldsymbol{x}) - \bar{u}(\boldsymbol{x}))^2 = \int_\Gamma (\boldsymbol{I}_{\text{in}}(\boldsymbol{x})\boldsymbol{U}_{\text{in}} + \boldsymbol{I}_{\text{out}}(\boldsymbol{x})\boldsymbol{U}_{\text{out}} - \bar{u}(\boldsymbol{x}))^2. \tag{15}$$

Suppose now that the problem for $u_h$ in $\Omega_{\text{in}}$ leads to an algebraic equation of the form

$$\boldsymbol{K}_{\text{in,in}}\boldsymbol{U}_{\text{in}} + \boldsymbol{K}_{\text{in,out}}\boldsymbol{U}_{\text{out}} = \boldsymbol{F}_{\text{in}}. \tag{16}$$

The domain integrals in matrices $\boldsymbol{K}_{\text{in,in}}$ and $\boldsymbol{K}_{\text{in,out}}$ extend only over $\Omega$. The nodal values $\boldsymbol{U}_{\text{out}}$ are merely used as degrees of freedom to interpolate $u_h$ in the domain $\Omega$. If (16) is supplemented with the equation resulting from the minimization of functional (15), the system to be solved is finally

$$\begin{bmatrix} \boldsymbol{K}_{\text{in,in}} & \boldsymbol{K}_{\text{in,out}} \\ \boldsymbol{N}_\Gamma & \boldsymbol{M}_\Gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{\text{in}} \\ \boldsymbol{U}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{\text{in}} \\ \boldsymbol{f}_\Gamma \end{bmatrix}, \tag{17}$$

where

$$\boldsymbol{M}_\Gamma = \int_\Gamma \boldsymbol{I}^{\text{t}}_{\text{out}}(\boldsymbol{x})\boldsymbol{I}_{\text{out}}(\boldsymbol{x}), \quad \boldsymbol{f}_\Gamma = \int_\Gamma \boldsymbol{I}^{\text{t}}_{\text{out}}(\boldsymbol{x})\bar{u}(\boldsymbol{x}), \quad \boldsymbol{N}_\Gamma = \int_\Gamma \boldsymbol{I}^{\text{t}}_{\text{out}}(\boldsymbol{x})\boldsymbol{I}_{\text{in}}(\boldsymbol{x}).$$

It is important to note that this implementation maintains the connectivity of the background mesh. There are a number of other methods for imposing Dirichlet boundary conditions on fixed meshes which could have been used, see for example the strategies proposed in the *Immersed Boundary Method* [22], the *Fictitious Domain Method* [23, 24] and the *hybrid Cartesian/immersed boundary methods* [25–27]. The only difficulties in the imposition of boundary conditions near boundaries with irregular geometry are associated with the fact that we consider the interface between the 'inside the domain' region and the 'outside the domain' region of each cut element to be a single line segment. This introduces limitations in the tracking of the solid body geometry, especially when sharp corners are present or, in the current work, in those elements that are cut by the solid body boundary and the level set function at the same time. This is a common limitation of fixed mesh methods that can be addressed by coding more complex subelement integration subroutines, although this has not been done in the current work.

*4.2.3. Tracking of* $\Gamma_{\text{f}}$. As explained before, the free surface is tracked by means of a level set function. However, there still remain some points to be clarified about how this process is exactly carried out. The main particularity of our problem is that the fluid boundary $\Gamma_{\text{f}}$ is represented not only by $\Gamma_{\text{free}}$ but also by $\Gamma_{\text{sf}}$ (see Figure 1). Theoretically, if the advection velocity of $\Psi$ is that of the rigid body in $\Gamma_{\text{sf}}$, $\Gamma_{\text{free}} \cap \Gamma_{\text{sf}} = \Gamma_{\text{sf}}$, but in practice both boundaries will rarely exactly coincide. A strategy has to be devised to deal with this lack of coincidence of the functions that define the boundary of the fluid domain, which is due to numerical approximation errors.

The first situation we consider is the one depicted in Figure 4. As we can see, $\Gamma_{\text{sf}}$ does not coincide with the free surface $\Gamma_{\text{free}}$, understood as the isovalue $\Psi = 0$ of the level set function. This problem can be solved in the following manner: let us define $\Gamma^*_{\text{free}}$ as the part of $\Gamma_{\text{free}}$ interior to $\Omega_{\text{s}}(t)$. Now we can define the fluid boundary as:

$$\Gamma_{\text{f}} = (\Gamma_{\text{free}} \setminus \Gamma^*_{\text{free}}) \cup \Gamma_{\text{sf}}$$



DOI: 10.1002/fld

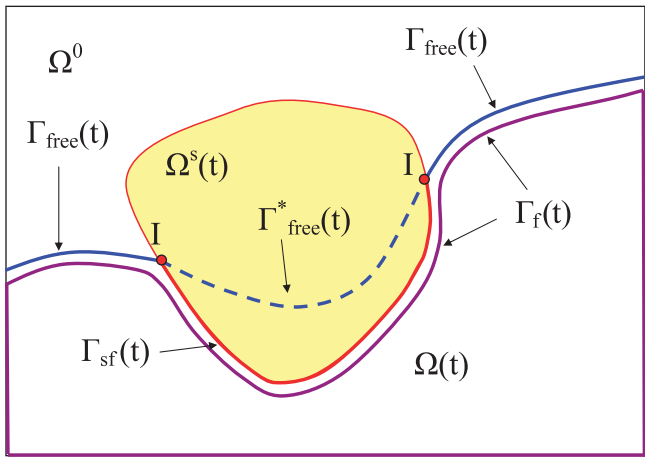


Figure 4. Lack of coincidence of the boundary defined by $\Gamma_{\text{free}}$ and $\Gamma_{\text{sf}}$.

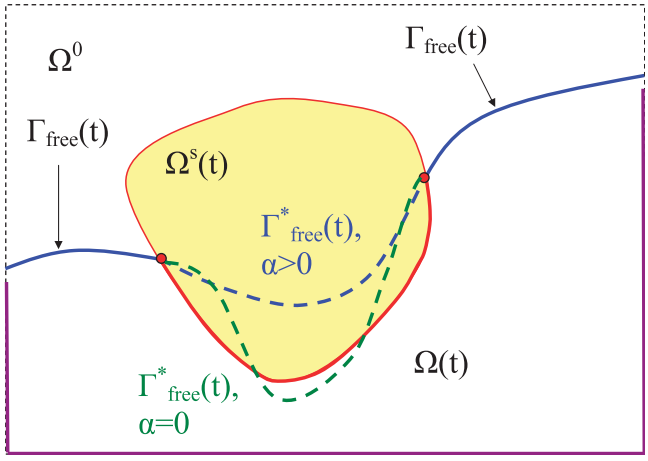


Figure 5. $\Gamma^*_{\text{free}}$ for different $\alpha$ parameters.

The second and more delicate problem occurs when $\Gamma_{\text{free}}$ gets *delayed* with respect to $\Gamma_{\text{sf}}$ due to the extra numerical diffusion that appears in the advection of the level set. This situation is outlined in Figure 5. As we are considering continuum mechanics, the only way the water surface can separate from the solid is slipping. In order to avoid an incorrect separation process of the fluid from the solid, we rely on the velocity in the non-computed *air domain*.

In the *air domain*, we do not solve the Navier–Stokes equations. This is the reason why we have to compute an artificial velocity to advect the level set function. To do this, we solve a modified Stokes problem with the following particularities: find a velocity $\boldsymbol{u}:\Omega^0\setminus\Omega(t)\longrightarrow\mathbb{R}^d$ and a pressure $p:\Omega^0\setminus\Omega(t)\longrightarrow\mathbb{R}$ such that

$$-\nabla\cdot(2\nu\nabla^{\mathrm{S}}\boldsymbol{u})+\nabla p=\boldsymbol{f}, \tag{18}$$

$$\nabla\cdot\boldsymbol{u}=-\alpha, \tag{19}$$

where $\alpha$ is a constant that is positive in the solid domain, and zero outside of it. Note that using the positive constant $\alpha$ in the interior of the solid body does not introduce any extra numerical error since (19) refers only to the computation of the *artificial* velocity. Slip boundary conditions are applied except for $\Gamma_{\text{free}}$ where the fluid velocity is imposed. The positive constant $\alpha$, which has units of $[T^{-1}]$, makes the solid body act as a sink, which allows us to avoid any numerically induced



DOI: 10.1002/fld

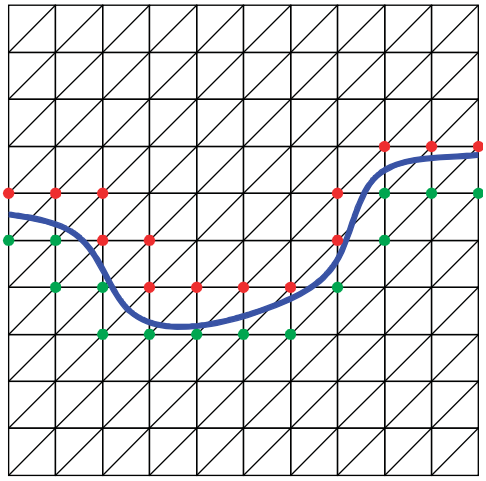

Figure 6. Highlighted nodes above the level set function: nodes in which we prescribe the level set function to be equal to the signed distance from the nodes to the free surface. Highlighted nodes below the level set function: nodes whose nodal values are such that the level set function is zero on the free surface (Approximate imposition of boundary conditions).

*delay* in the advection of the interface. Again, the subgrid-scale method is used to stabilize the problem and allow for equal velocity–pressure interpolation. This problem needs to be solved only in a region close to $\Gamma_{\mathrm{f}}$. In the rest of $\Omega^0 \setminus \Omega(t)$, the advection velocity can be more straightforwardly computed, for example by means of a linear extrapolation.

As usual when using the level set method, we need to reinitialize the level set function for every certain number of time steps. This reinitialization may move the position of the interface. In order to avoid this, a special procedure is used in the nodes of cut elements. It is a slight variation of the method presented in [28], which consists of the following: Suppose that we are given the free surface configuration in Figure 6. Now we divide the nodes belonging to the elements cut by the free surface in two sets, each set corresponding to one side of the free surface. In the first wet side, we prescribe the nodal values of the level set function to be equal to the (signed) distance between the node and the free surface.

We use the degrees of freedom of the nodes in the second side of the free surface to prescribe the reinitialized level set function to be zero valued on the free surface, by using the approximate imposition of boundary conditions of the previous section. On the rest of the nodes we prescribe the level set function to be the signed distance from the nodes to the free surface, although we could also use the more efficient procedure in [28], or other techniques as the one described in [15]. This algorithmic procedure allows the reinitialized level set function to very accurately track the free surface, that is, the free surface for the reinitialized level set function minimizes the distance between the interface position before and after the reinitialization. This guarantees that no significative mass loss is introduced during the reinitialization of the level set function, since the error in the reinitialization is of $\mathcal{O}(\delta t^2)$.

*4.2.4. The ALE Navier–Stokes equations on the fixed-mesh.* In order to write the ALE Navier-Stokes equations on the background mesh, we first need to define the mesh velocity in the virtual mesh. The mesh velocity on the boundary points can be computed from their position $\boldsymbol{x}_b^{n+1}$ and $\boldsymbol{x}_b^n$, where subscript $b$ refers to points on $\Gamma_{\mathrm{free}}$. Once the velocity at the nodes of $\Gamma_{\mathrm{free}}$ is known, it has to be extended to the rest of the nodes. A classical possibility is to solve the Laplace problem $\Delta \boldsymbol{u}_{\mathrm{dom}} = \boldsymbol{0}$ using $\boldsymbol{u}_{\mathrm{dom},b}^{n+1}$ as Dirichlet boundary conditions. However, it is also possible to restrict $\boldsymbol{u}_{\mathrm{dom}} \neq \boldsymbol{0}$ to the nodes next to $\Gamma_{\mathrm{free}}^{n+1}$, since in our approach mesh distortion does not accumulate from one time step to another.

Once we have been able to define the deformed mesh $M_{\mathrm{ALE}}^{n+1}$, we have to project the mesh velocity and the unknowns at $t^n$ from $M_{\mathrm{ALE}}^{n+1}$ to $M^{n+1}$. Let $P^{n+1}$ be the projection of finite element



DOI: 10.1002/fld

functions defined on $M_{\mathrm{ALE}}^{n+1}$ to $M^{n+1}$. There are several possibilities to define this projection operator, among them a simple interpolation scheme or the more involved $L^2$ projection with restrictions (for example conserving momentum or imposing that the resulting projected velocity field is divergence free). Some of the possibilities for this projection schemes are discussed in [29].

The problem to be solved at time step $n+1$ is to find a velocity $\boldsymbol{u}_h^{n+1}$ and a pressure $p_h^{n+1}$ such that

$$m_1^{n+1}(\delta t^{-1}(\boldsymbol{u}_h^{n+1}(\boldsymbol{x})-P^{n+1}(\boldsymbol{u}_{h,\mathrm{ALE}}^{n}(\boldsymbol{x}^n))),\boldsymbol{v}_h)+a^{n+1}(\boldsymbol{u}_h,\boldsymbol{v}_h)$$
$$+c^{n+1}(\boldsymbol{u}_h-P^{n+1}(\boldsymbol{u}_{\mathrm{dom,ALE}});\boldsymbol{u}_h,\boldsymbol{v}_h)+b_1^{n+1}(p_h,\boldsymbol{v}_h)=l_1^{n+1}(\boldsymbol{v}_h), \tag{20}$$

$$b_2^{n+1}(q_h,\boldsymbol{u}_h)+s^{n+1}(q_h,p_h)=l_2^{n+1}(q_h), \tag{21}$$

which again must hold for all velocity test functions $\boldsymbol{v}_h$ and pressure test functions $q_h$.

Problem (20)–(21) is posed on $M^{n+1}$ which, as it has been said, coincides with $M^0$ except for the splitting of the elements crossed by the interface. Even this difference can be avoided if instead of prescribing exactly the boundary conditions, an approximation is performed. Therefore, the goal of using a fixed mesh during the whole simulation has been achieved.

## 5. THE FINAL ALGORITHM

The resulting final algorithm for solving the fluid–structure interaction problem is shown in Algorithm 1. Note that we compute the body position and the free surface position only once, at the beginning of the time step and before the iterative process. To do this, we use the velocities at the previous time step. This approximation allows us to avoid recomputing the body position at each inner iteration, and obviously also reduce the computational effort at each time step.

**Algorithm 1** Final algorithm.

```
for istep = 1:n do
    Initialize variables
    Obtain the fluid domain:
        Update the solid body position
        Advect the level set function Ψ
        Compute the fluid domain Ω^{n+1}
    Move the mesh virtually and compute the mesh velocity
    Compute the mesh M^{n+1} by splitting the elements of M^0
     Set i = 1 (iteration counter)
    while Not converged do
       Solve the Navier-Stokes Equations
       Check convergence
       i = i + 1
    end while
    Solve the rigid body motion equations
    If needed, reinitialize Ψ
end for
```

## 6. NUMERICAL EXAMPLES

In this section, we present two numerical examples that illustrate the behavior of the methodology proposed in this work.



DOI: 10.1002/fld

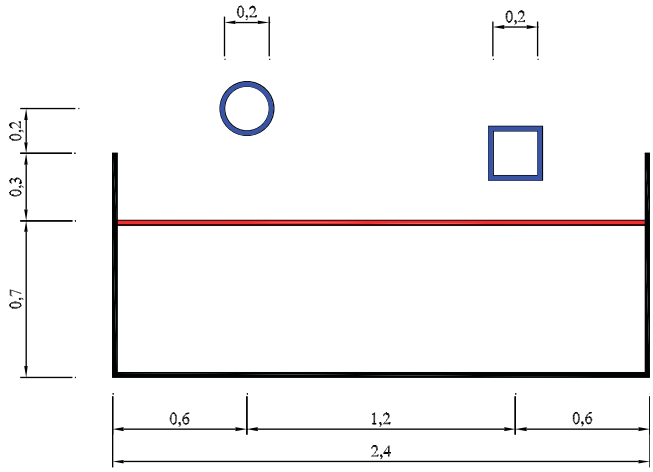


Figure 7. Initial configuration, example 1.

The first example we propose consists of two rigid bodies falling into an incompressible fluid. To run this example we use the FM-ALE method on the fluid part, and we track the free surface by means of a level set function. The initial configuration of the problem can be seen in Figure 7.

The hold-all domain is the rectangle $B=[0,2.4]\times[0,1]$. A background mesh of 7968 linear triangles has been used. The fluid density is $\rho=1$ and the viscosity is set to $\mu=0.001$. For the solid bodies, density is $\rho=0.75$.

Both the fluid and the structure are subject to a vertical gravity force of value $g=-10$. We apply slip boundary conditions both at the interface between the fluid and the deposit wall and at the interface between the solid bodies and the fluid. This means that only the velocity in the direction normal to the interface has to coincide between the fluid and the solid bodies. In the free surface, $\Gamma_{\text{free}}$ tractions in the normal direction are prescribed to zero.

The time step has been set to $\delta t=0.02$ and 150 time steps have been carried out. Regarding the advection of the level set function, the $\alpha$ parameter for the artificial mass sink explained in Section 4.2.3 is taken as $\alpha=2$.

Figures 8–11 show the results for various time steps. Let us remark that the solution obtained is smooth along all the computation, even for the first critical steps in which the rigid bodies *contact* the free surface. The irregular boundaries are due to the fact that for ease of post processing, we have plotted the solution in the elements cut by the boundary without taking into account that the boundary of the domain does not fit the boundary of the elements.

The kind of computation involved in this example, in which the fluid region undergoes very large deformations, would have implied the need for continuous remeshing if classical ALE methods were used. Our fixed mesh strategy avoids it by projecting the results to the background mesh at each time step.

The most critical situations in this problem, which are the instant in which *the fluid closes around the rigid body* and surrounds it completely, and the instant when *the solid body breaks the free surface*, are handled in a very natural way with the level set function strategy.

Coupling conditions between the fluid and the structure need to be imposed in an approximate way. To do this we use the strategy described in Section 4.2.2, which is used to impose velocity continuity in the direction normal to the interface.

In the second example, we simulate an oval body falling into an incompressible fluid. Again, we use the FM-ALE method to simulate the fluid part, and we track the free surface with a level set function. The initial configuration for this problem can be seen in Figure 12.

The hold-all domain is the rectangle $B=[0,1]\times[0,1]$. The fluid density is $\rho=1$, and the viscosity is set to $\mu=0.01$. For the solid body, the density is $\rho=0.5$. Both the fluid and the structure



DOI: 10.1002/fld

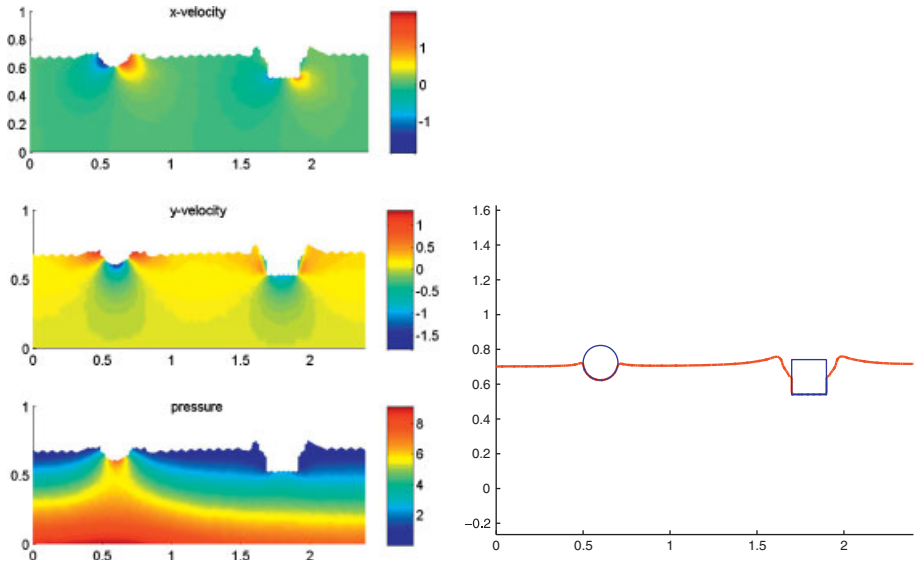


Figure 8. Unknown fields and free surface at $t=0.36$.

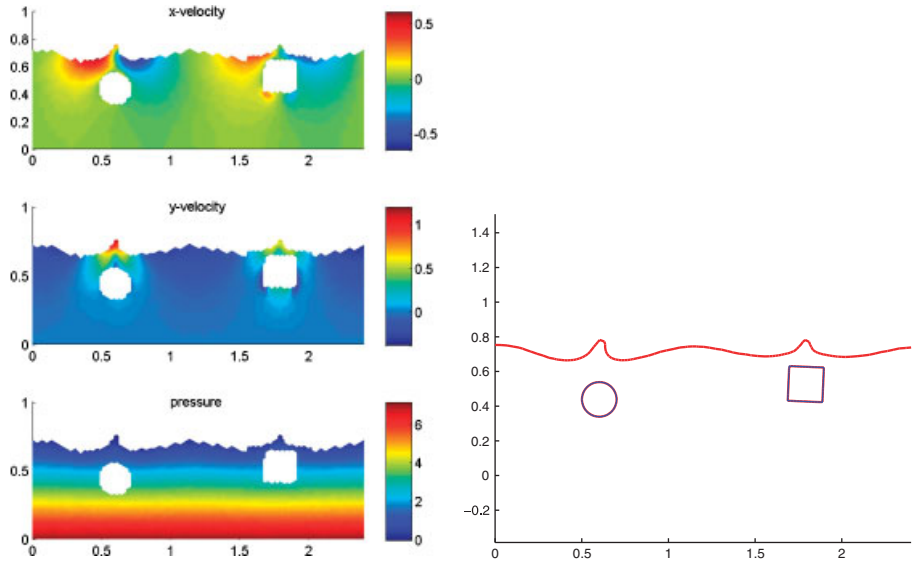


Figure 9. Unknown fields and free surface at $t=0.78$.

are subject to a vertical gravity force of value $g=-10$. The time step has been set to $\delta t=0.02$ and 500 time steps have been carried out. Again, the parameter $\alpha$ has been set to $\alpha=2$.

In this case, we have used three different meshes in order to compare the behavior of the method with different element sizes. In the first case, we have used a relatively coarse triangle mesh with 1890 nodes. In the second case we have used a finer mesh, with 5205 nodes and the last mesh consisted of 11 614 nodes. We compare the vertical displacement of the center of mass of the solid body in the three cases in Figure 13. We can see that we obtain a solution close to the converged one for the vertical displacement with the two finer meshes. We have also represented vertical and horizontal velocities for the solid body in Figure 14. Horizontal velocity for the solid body should be zero due to the problem symmetry. The numerical errors introduced in the geometry interpolation of the cut elements (the considered meshes are not symmetric) are the cause for



DOI: 10.1002/fld

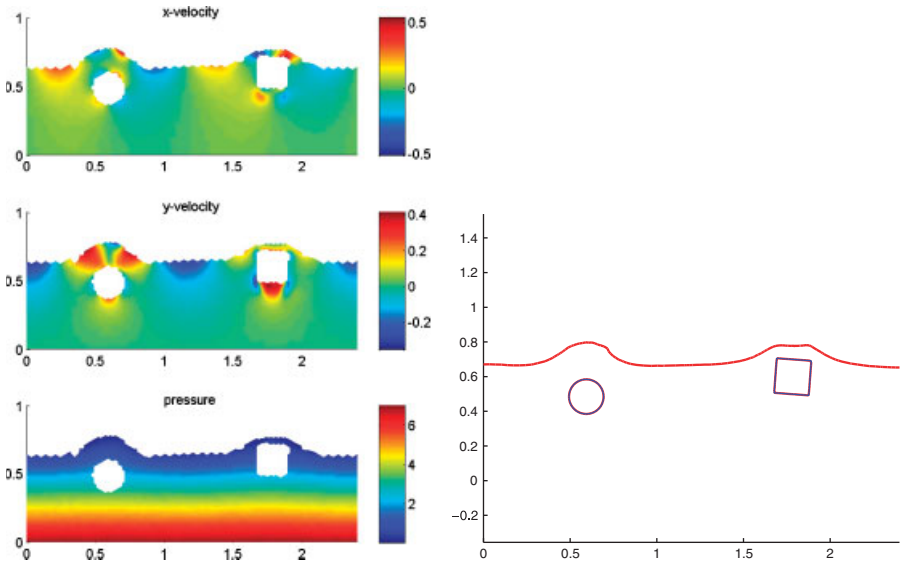


Figure 10. Unknown fields and free surface at $t=0.98$.

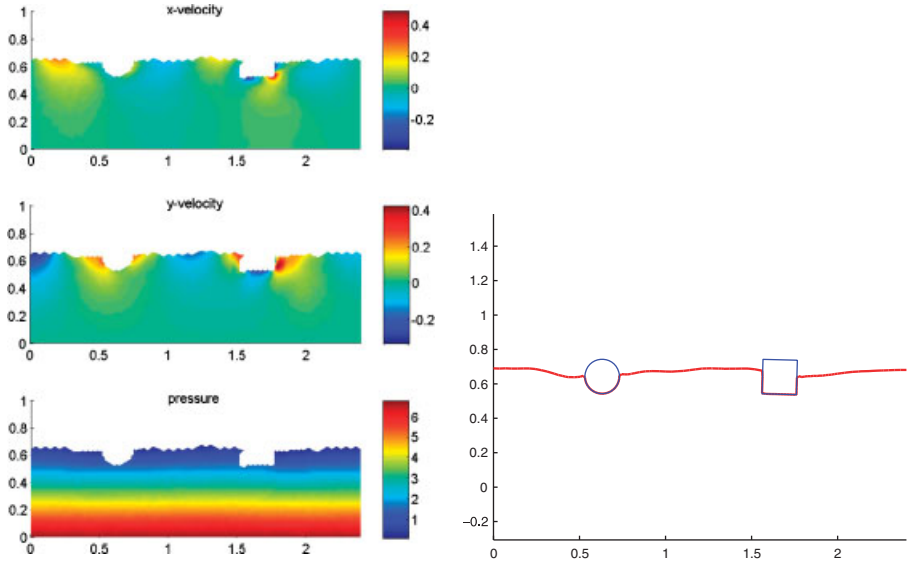


Figure 11. Unknown fields and free surface at $t=2.40$.

horizontal velocity to appear, although the horizontal velocity is small compared with the vertical velocity and the domain size. In any case, as already explained this local error could be removed by improving the numerical integration, which can be done by introducing appropriate subelements for integration purposes.

Fluid velocity, free surface and solid body configurations can be seen in Figures 15–17. As expected, in the final configuration, when the body is at rest, half of the body is inside the fluid domain and half of it is in the air domain, since $\rho_s/\rho_f=0.5$. Again, the lack of symmetry in the velocity fields in Figure 17, when both the fluid and the solid are close to rest, is due to the cumulative numerical errors of the geometry interpolation in the cut elements.

Figure 18 shows the time evolution of the total fluid mass. We can see that mass loss is larger in the coarse mesh case, and much smaller for the finer meshes although no method for correcting mass loss has been used.



DOI: 10.1002/fld

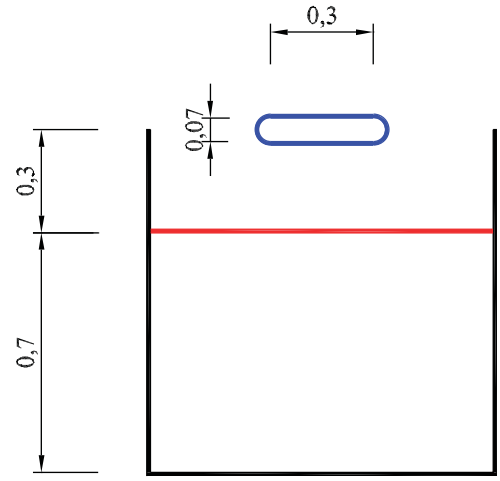


Figure 12. Initial configuration, example 2.

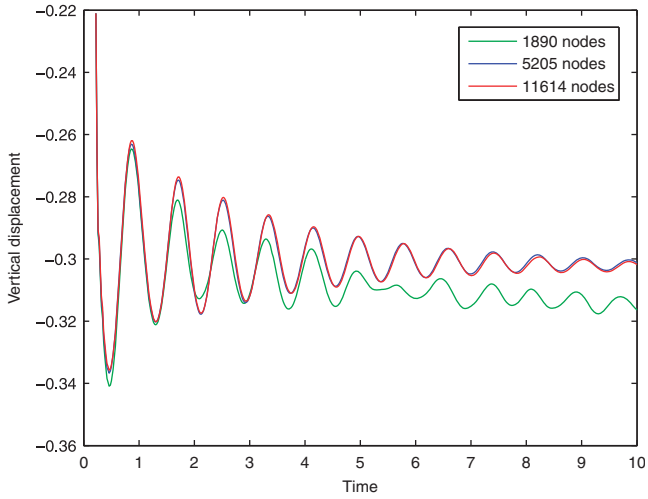


Figure 13. Time evolution of the vertical displacement.

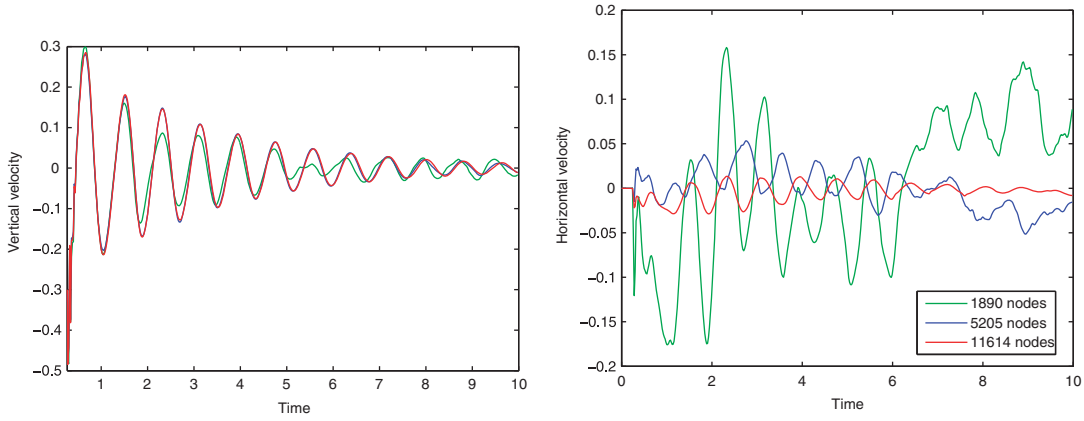


Figure 14. Time evolution of the vertical and horizontal velocities.



DOI: 10.1002/fld

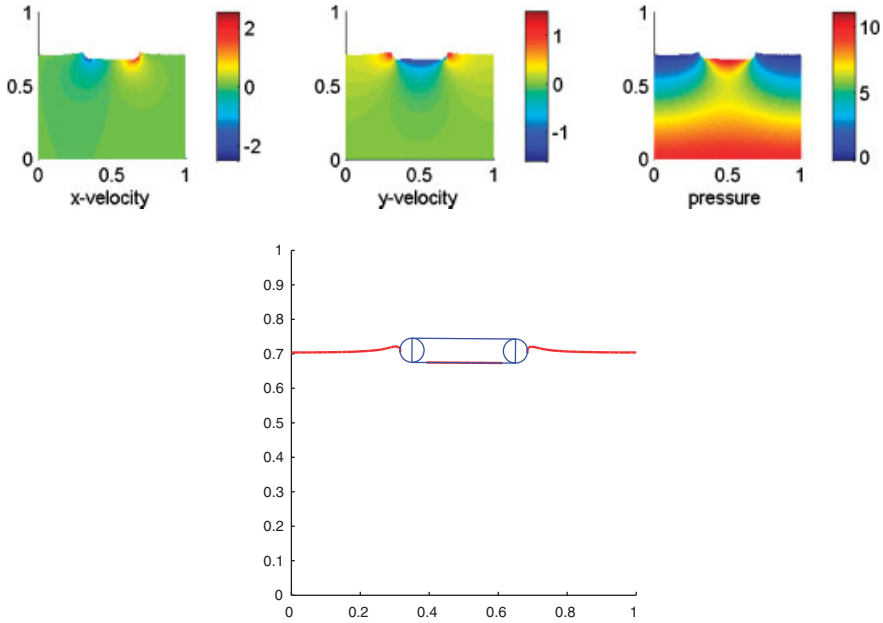


Figure 15. Unknown fields and free surface at $t = 0.28$.

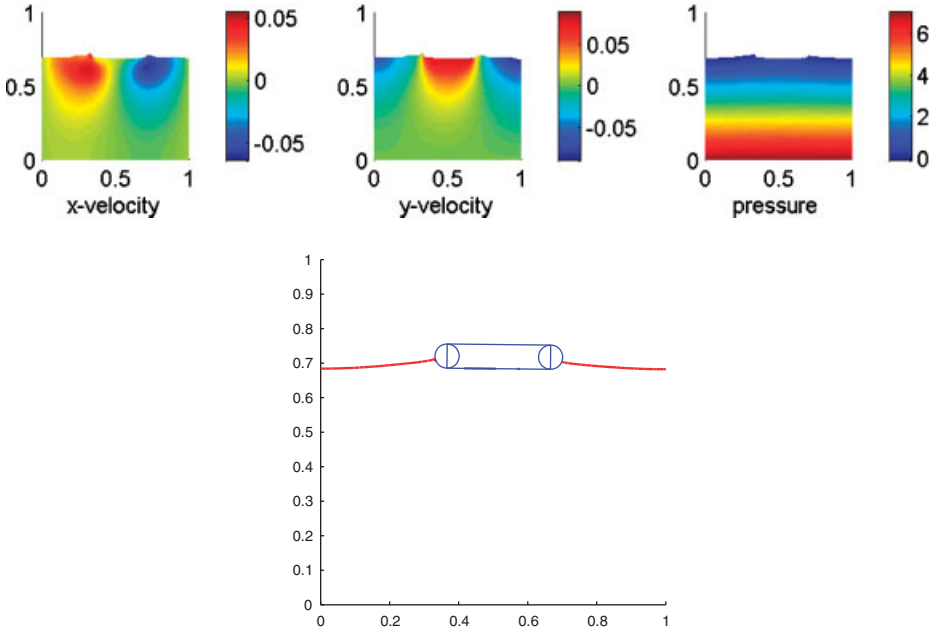


Figure 16. Unknown fields and free surface at $t = 2.5$.

## 7. CONCLUSIONS

In this paper, the FM-ALE approach has been applied to the problem of floating solids. The main feature of the method is its capability of using a fixed background mesh, but at the same time correctly taking into account the domain movement in the computation of the time derivatives. Moreover, values of the unknowns for the so-called *newly created nodes* are clearly and uniquely defined with the FM-ALE approach. For free surface problems, the FM-ALE method avoids the



DOI: 10.1002/fld

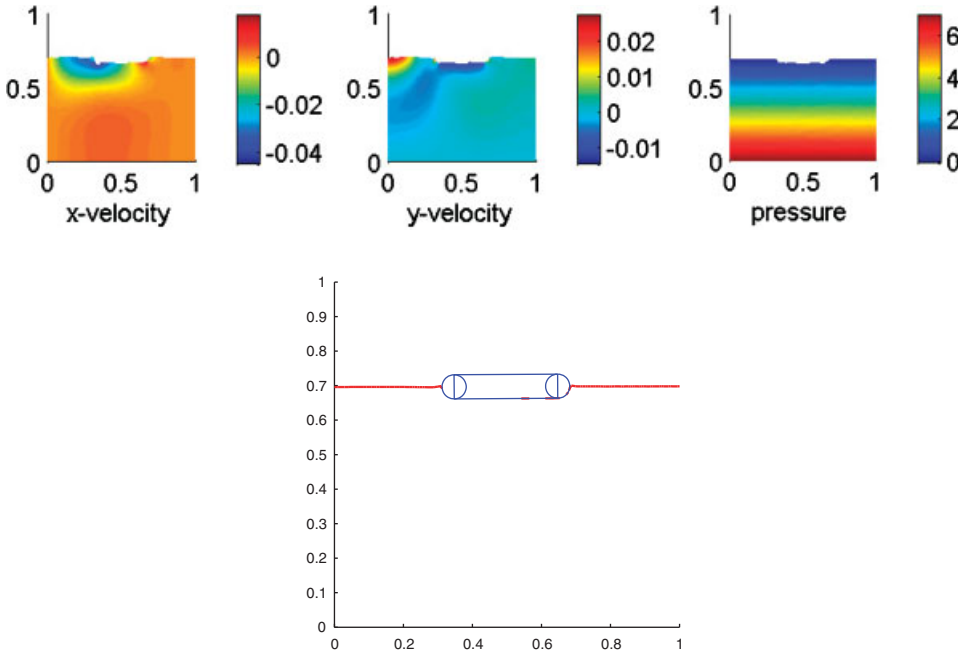


Figure 17. Unknown fields and free surface at $t = 10.0$.

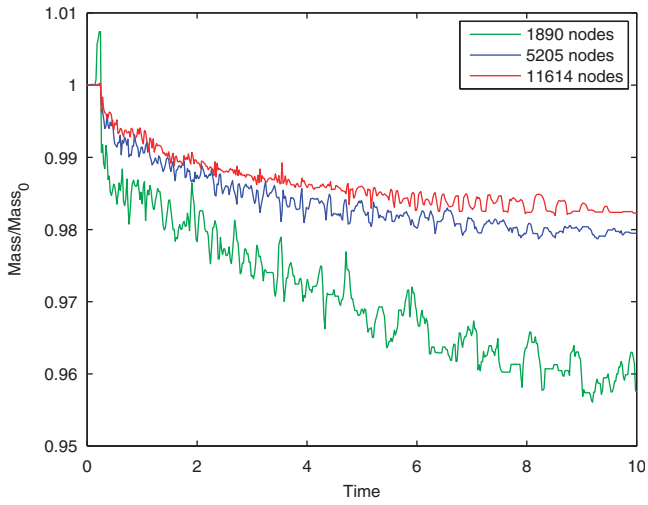


Figure 18. Time evolution of mass (with respect to initial mass).

need for remeshing which appears in the classical Lagrangian or ALE methods. Moreover, the free surface is tracked in a very natural way with the level set function strategy, allowing for the solid body *breaking the free surface* without any further algorithmic steps.

We have paid special attention to the interaction between the level set function and the solid boundary function, which defines the fluid domain: in order to avoid the delay of the level set function with respect to the solid boundary function, we have modified the Stokes problem to be solved in the empty part of the domain, imposing the velocity divergence to be negative inside the solid body.



DOI: 10.1002/fld

The proposed method has been used to solve the problem of rigid bodies falling into water, and has proved to be robust and provide smooth solution fields, even at the critical instant in which the solid body contacts the free surface.

## ACKNOWLEDGEMENTS

J. Baiges would like to acknowledge the support received from the *Ministerio de Ciencia e Innovación* of the Spanish Government and the *Col·legi d'Enginyers de Camins, Canals i Ports de Catalunya*.

H. Coppola-Owen would like to acknowledge the support received from the *Departament d'Universitats, Recerca i Societat de la Informació* at the *Generalitat de Catalunya* and the *European Social Fund* through a doctoral grant.

## REFERENCES

1. Houzeaux G, Codina R. A finite element model for the simulation of lost foam casting. *International Journal for Numerical Methods in Fluids* 2004; **46**:203–226.
2. Codina R, Houzeaux G, Coppola-Owen H, Baiges J. The Fixed-Mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics* 2009; **228**:1591–1611.
3. Coppola-Owen H, Codina R. A finite element model for free surface flows on fixed meshes. *International Journal for Numerical Methods in Fluids* 2007; **54**:1151–1171.
4. Baiges J, Codina R. The Fixed Mesh-ALE approach applied to solid mechanics and fluid–structure interaction problems. *International Journal for Numerical Methods in Engineering* 2009; **81**:1529–1557.
5. Yan S, Ma QW. Numerical simulation of fully nonlinear interaction between steep waves and 2D floating bodies using the QALE-FEM method. *Journal of Computational Physics* 2007; **221**:666–692.
6. Wang CZ, Wu GX. Analysis of second-order resonance in wave interactions with floating bodies through a finite-element method. *Ocean Engineering* 2008; **35**:717–726.
7. Lin P. A fixed-grid model for simulation of a moving body in free surface flows. *Computers and Fluids* 2007; **36**:549–561.
8. Houzeaux G, Codina R. A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:3343–3377.
9. Coppola-Owen H, Codina R. Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *International Journal for Numerical Methods in Fluids* 2005; **49**:1287–1304.
10. Li J, Hesse M, Ziegler J, Woods A. An arbitrary Lagrangian–Eulerian method for moving-boundary problems and its application to jumping over water. *Journal of Computational Physics* 2005; **208**:289.
11. Do-Quang M, Amberg G. The splash of a solid sphere impacting on a liquid surface: numerical simulation of the influence of wetting. *Physics of Fluids* 2009; **21**:022102.
12. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**(2):83–130.
13. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
14. Löhner R, Cebral JR, Camelli FF, Baum JD, Mestreau EL. Adaptive embedded/immersed unstructured grid techniques. *Archives of Computational Methods in Engineering* 2007; **14**:279–301.
15. Codina R, Soto O. A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. *International Journal for Numerical Methods in Fluids* 2002; **40**:293–301.
16. Badia S, Codina R. Analysis of a stabilized finite element approximation of the transient convection–diffusion equation using an ALE framework. *SIAM Journal on Numerical Analysis* 2006; **44**:2159–2197.
17. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
18. Hughes TJR. Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.
19. Franca LP, Frey SL. Stabilized finite element methods: II. The incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:209–233.
20. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
21. Codina R, Baiges J. Approximate imposition of boundary conditions in immersed boundary methods. *International Journal for Numerical Methods in Engineering* 2009; **80**(11):1379–1405.
22. Peskin CS. Flow patterns around heart vales: a numerical method. *Journal of Computational Physics* 1972; **10**:252–271.
23. Glowinski R, Pan T-W, Périaux J. A fictitious domain method for Dirichlet problems and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:203–303.
24. Glowinski R, Pan TW, Hesla TI, Joseph DD, Périaux J. A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow. *International Journal for Numerical Methods in Fluids* 1999; **30**:1043–1066.



DOI: 10.1002/fld

25. Gilmanov A, Sotiropoulos F. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics* 2005; **207**:457–492.
26. Ferziger JH, Tseng YH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
27. Mohd-Yusof J. Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries. *CTR Annual Research Briefs*, Stanford University, NASA AMES, 1997.
28. Hartmann D, Meinke M, Schröder W. Differential equation based constrained reinitialization for level set methods. *Journal of Computational Physics* 2007; **227**:6821–6845.
29. Houzeaux G, Codina R. Transmission conditions with constraints in finite element domain decomposition methods for flow problems. *Journal of Computational Physics* 2001; **17**:179–190.



DOI: 10.1002/fld