Fast fluid-structure interaction simulations using a displacement-based finite element model equipped with an explicit streamline integration prediction

P.B. Ryzhakov[a,b], J. Marti[a,b], S.R. Idelsohn[a,c], E. Oñate[a,b]

[a] *Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE) Gran Capitán s/n, 08034 Barcelona, Spain*
[b] *Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*
[c] *Institució Catalana de Recerca i Estudis Avanats (ICREA), Barcelona, Spain*

Abstract

We propose here a displacement-based updated Lagrangian fluid model developed to facilitate a monolithic coupling with **a wide range of structural elements described in terms of displacements**. The novelty of the model consists in the use of the explicit streamline integration for predicting the end-of-step configuration of the fluid domain. It is shown that this prediction considerably alleviates the time step size restrictions faced by the former Lagrangian models due to the possibility of an element inversion within one time step. The method is validated and compared with conventional approaches using three numerical examples. Time step size and corresponding Courant numbers leading to optimal behavior in terms of computational efficiency are identified.

Keywords: incompressible flows, Navier-Stokes, fluid-structure interaction, Particle Finite Element Method, Lagrangian, coupled problems

1. Introduction

Fluid models based on Lagrangian descriptions of motion have proven to be advantageous for treating free-surface flows and problems that involve large motion of interfaces, such as fluid-structure interaction (FSI) problems. Since in Lagrangian approaches the computational mesh follows the fluid

Email address: pryzhakov@cimne.upc.edu (P.B. Ryzhakov)

movement, tracking the moving boundaries does not require any additional techniques being an intrinsic feature of the model. Lagrangian fluid and fluid-structure interaction models have been developed both in the Finite Element Method (FEM) [1], [2], [3], [4], [5], [6], [7] and the Smooth Particle Hydrodynamics (SPH) contexts [8], [9], [10], [11].

The main drawback of the Lagrangian fluid models based on finite elements consists in the danger of mesh degradation when severe domain deformations are faced. It is usually alleviated by performing re-meshing at every time step of the transient problem. This considerably improves the mesh quality and diminishes the amount of highly distorted elements. However, even if re-meshing is performed, it is impossible to ensure that within one time step no element becomes inverted [12]. Element inversion results in a negative Jacobian of the corresponding element. This leads to divergence of the non-linear iterative procedure and, consequently, to the failure of the simulation.

Using variable time steps may serve as a remedy to the element inversion. The critical time step is defined as the one leading to element degradation (zero Jacobian). It can be estimated for each mesh element based upon the historical velocity value $\mathbf{v}_n$, considering that the movement of each of its nodes can be approximated as $\mathbf{v}_n \Delta t$, where Δt is the time step. The smallest value of the estimated critical time is thus identified. The safe time step to be used in the simulation is usually taken as 0.5 t_{crit}. Thus, the danger of element inversion is alleviated. Such implementation has shown to be an acceptable remedy and is used in practice [13, 14, 3]. Nevertheless, safe time step sizes might become excessively small and result in extensive computational times.

Recently, a conceptually different methodology aiming at working with large time steps has been proposed in [15, 16, 17] [1]. The idea consists in obtaining the position of the mesh at the new time step using the explicit streamline integration of the nodal positions. Once the mesh position at the current step is obtained, the Navier-Stokes equations are solved obtaining velocity and pressure without further moving the computational mesh. This method can be viewed as a decoupling of the convection step from the rest of the solution of governing equations. The technique has proven to be highly efficient (as the explicit step does not involve the Jacobian computation) and

[1] it has been developed both for Lagrangian and Eulerian formulations

even allowed for nearly real-time computations.

Despite its obvious advantage the actual implementation of the above-described technique done in [15, 16, 17] suffers from one drawback. The end-of-step position of the nodes is approximated explicitly and is never corrected. Ideally, after solving the Navier-Stokes equations for the primary variables (typically, the velocity and the pressure), one must correct the position of the mesh and iterate between the mesh position update and Navier-Stokes solution until convergence is achieved.

In this paper we propose a technique that ensures that the mesh position at the end of each time step respects the Navier-Stokes equations and no approximation in the mesh position is introduced. The idea consists in using the displacement instead of velocity as the primary kinematic variable of the model. This ensures that the solution of Navier-Stokes equations automatically provides the corrected mesh position. Thus no approximation error is introduced neither in the mesh position nor in the evaluation of the discrete operators dependent on the mesh configuration. Noting that a displacement-based fluid can be naturally coupled to a wide range of structural models (as the majority of structures are described in terms of displacements), we extend the approach to the field of FSI. Sharing the same kinematic variable in sub-domains of a multi-physics problems facilitates implementation of the solvers that simultaneously advance in time the fluid and the structure, known as monolithic solvers.

Note that the monolithic FSI models can be defined using kinematic descriptions different than the one used in this paper. These include the ones using Arbitrary Lagrangian/Eulerian for both domains (e.g. [18],[19]), Eulerian fluids with Lagrangian structures (e.g. [20]) and ALE fluids with Lagrangian structures (e.g.[21], [22]). ALE methods are generally restricted to moderate mesh deformations, while those that employ Eulerian (fixed-grid) fluid formulations require additional techniques for boundary tracking. Moreover, the fluid-structure contact, naturally accounted for in the mesh-matching interfaces of fully Lagrangian approaches, must be explicitly modeled in other approaches. While being advantageous for many problems, these above-mentioned non-Lagrangian frameworks lie outside the scope of the present paper as here we strive to define a new step in the development of purely Lagrangian monolithic solvers. A comprehensive review of existing FSI methods can be found in [23], [24].

The paper is organized as follows. We first introduce the governing equations for the fluid. The equations are discretized in space and time. Next the

explicit streamline integration scheme for approximating new domain configuration is specified. Then the algorithm based upon this prediction and the subsequent solution of the displacement-based fluid equations is presented. Following that we address the use of the proposed methodology in FSI simulations. A monolithic fluid-structure interaction algorithm is outlined. The paper concludes with three examples. In the first one the proposed fluid model is validated by comparing the simulation results with the analytic solution. In the second and third examples FSI problems are solved. The method is tested for a wide range of time steps and is compared to the previously proposed schemes. Convergence characteristics are addressed and feasible simulation settings are identified.

2. Numerical model

2.1. Governing equations for the fluid at continuum level

Let us consider a fluid domain Ω with a fixed boundary Γ_d and a free surface Γ_n (see Fig. 1). We shall consider viscous incompressible Newtonian fluids being the most common one in the majority of engineering applications. The governing system consists of momentum and mass conservation and corresponds to the Navier-Stokes equations. **Since we strive to develop a fluid model that facilitates coupling with multiple structural elements (typically described in terms of displacements), displacement rather than velocity is chosen as the kinematic variable.** Using the displacement $\mathbf{d}$ as the primary variable the momentum equations can be written in vector form as:

$$\rho \frac{D^2\mathbf{d}}{Dt^2} - \mu\Delta\left(\frac{D\mathbf{d}}{Dt}\right) + \nabla p = \mathbf{f} \tag{1}$$

where $\mathbf{d} = [d_x, d_y]^T$ (in 2D), p is the pressure, t is time, $\mathbf{f}$ is the body force, ρ is the density and μ is the dynamic viscosity of the fluid and Δ is the Laplacian operator. D stands for the material derivative.

Mass conservation equation used here corresponds to a commonly used quasi-incompressible approximation. The advantages of using the quasi-incompressible rather than fully incompressible fluid formulation for the implementation of tightly coupled FSI solvers have been numerously demonstrated [25], [10], [7], [26], [3], [1],[6].

The quasi-incompressibile assumption allows us to directly relate the the time rate of change in pressure to volumetric strain rate as:

$$\frac{\partial p}{\partial t} = -\kappa \nabla \cdot \frac{D\mathbf{d}}{Dt} \tag{2}$$

where κ is the bulk modulus of the fluid.

The governing equations are completed with the boundary conditions (b.c.). At fixed walls Γ_d, the homogeneous Dirichlet b.c. is prescribed:

$$\mathbf{d} = 0 \qquad \text{at } \Gamma_d \tag{3}$$

At the free surface Γ_n (see dashed line in Fig. 1) the following Neumann b.c. is used :

$$\boldsymbol{\sigma} \cdot \mathbf{n} = 0 \qquad \text{at } \Gamma_n \tag{4}$$

where $\boldsymbol{\sigma}$ is the stress tensor.

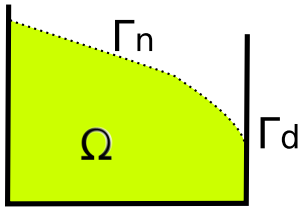


Figure 1: Fluid domain Ω.

2.2. Finite element formulation

Equal order linear interpolations for the displacement and the pressure over 3-noded triangles (2D) are used here for the spatial discretization of the

119 governing equations Eqs. (1), (2), i.e.:

$$\mathbf{d}(\mathbf{x}) = \sum_{I=1}^{3} \bar{\mathbf{d}}_I \mathbf{N}_I(\mathbf{x}) = \mathbf{N}^T \begin{pmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \\ d_{3x} \\ d_{3y} \end{pmatrix} \tag{5}$$

$$p(\mathbf{x}) = \sum_{I=1}^{3} \bar{p}_I N_I(\mathbf{x}) = \mathbf{N}_p^T \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \tag{6}$$

120 where

$$\mathbf{N} = \begin{pmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{pmatrix}^T \tag{7}$$

$$\mathbf{N}_p = \begin{pmatrix} N_1 & N_2 & N_3 \end{pmatrix}^T \tag{8}$$

121 N_I are the standard linear FE shape functions, over-bar distinguishes the
122 vectors of nodal quantities and I stands for the nodal index.

123 **In the subsequent derivation** we use a Backward Euler time integra-
124 tion scheme for the sake of simplicity[2]. The corresponding time approxima-
125 tions **considering Lagrangian framework**are

$$\bar{\mathbf{d}}_{n+1} = \bar{\mathbf{v}}_{n+1} \Delta t \tag{9}$$

$$\bar{\mathbf{v}}_{n+1} = \bar{\mathbf{v}}_n + \bar{\mathbf{a}}_{n+1} \Delta t \tag{10}$$

126 and, thus

$$\frac{D\bar{\mathbf{d}}}{Dt} = \bar{\mathbf{v}}_{n+1} = \frac{\bar{\mathbf{d}}_{n+1}}{\Delta t} \tag{11}$$

$$\frac{D^2\bar{\mathbf{d}}}{Dt^2} = \bar{\mathbf{a}}_{n+1} = \frac{\bar{\mathbf{d}}_{n+1}}{\Delta t^2} - \frac{\bar{\mathbf{v}}_n}{\Delta t} \tag{12}$$

127 where $\bar{\mathbf{v}}$ and $\bar{\mathbf{a}}$ are the velocity and the acceleration vectors, respectively.

[2] All the arguments presented in the paper hold for any implicit time integration scheme. In the implementation carried out in this work the second order accurate Newmark-Bossak scheme is used.

Given $\bar{\mathbf{d}}_n$ and $\bar{\mathbf{p}}_n$ at t_n, the discrete problem consists in finding $\bar{\mathbf{d}}_{n+1}$ and $\bar{\mathbf{p}}_{n+1}$ at t_{n+1} as the solution of

$$\bar{\mathbf{r}}_m = 0 \tag{13}$$
$$\bar{\mathbf{r}}_c = 0 \tag{14}$$

where $\bar{\mathbf{r}}_m$ and $\bar{\mathbf{r}}_c$ are the residua of the momentum and continuity equations, defined as:

$$\bar{\mathbf{r}}_m = \mathbf{F}_{n+1} - \rho\mathbf{M}\left(\frac{\bar{\mathbf{d}}_{n+1}}{\Delta t^2} - \frac{\bar{\mathbf{v}}_n}{\Delta t}\right) + \mu\mathbf{L}\frac{\bar{\mathbf{d}}_{n+1}}{\Delta t} - \mathbf{G}\bar{\mathbf{p}}_{n+1} \tag{15}$$
$$\bar{\mathbf{r}}_c = \kappa\mathbf{D}\bar{\mathbf{d}}_{n+1} - \mathbf{M}_p\left(\bar{\mathbf{p}}_{n+1} - \bar{\mathbf{p}}_n\right) \tag{16}$$

where $\mathbf{M}$, $\mathbf{L}$, $\mathbf{G}$ and $\mathbf{D}$ are mass, Laplacian, gradient and divergence matrices, respectively. **The pressure mass matrix is distinguished by the "p" subscript.** $\bar{\mathbf{v}}$ and $\bar{\mathbf{p}}$ are the nodal velocity and the nodal pressure vectors, respectively and $\bar{\mathbf{F}}$ is the body force vector. In the Lagrangian framework the material derivative coincides with the local derivative and, thus, the convective term vanishes from the governing equations.

The matrices and vectors are assembled from the elemental contributions defined as

$$\mathbf{M} = \sum_{elem}\int_{\Omega_e}\mathbf{N}\mathbf{N}^T d\Omega \tag{17}$$
$$\mathbf{L} = \sum_{elem}\int_{\Omega_e}\nabla\mathbf{N}\nabla\mathbf{N}^T d\Omega \tag{18}$$
$$\mathbf{G} = -\sum_{elem}\int_{\Omega_e}\nabla\mathbf{N}\mathbf{N}_p d\Omega \tag{19}$$
$$\bar{\mathbf{F}} = \sum_{elem}\rho\int_{\Omega_e}\mathbf{N}\mathbf{g} d\Omega \tag{20}$$
$$\mathbf{M}_p = \sum_{elem}\int_{\Omega_e}\mathbf{N}_p\mathbf{N}_p^T d\Omega \tag{21}$$
$$\mathbf{D} = -\mathbf{G}^T \tag{22}$$

Note that the discrete operators given by Eqs. (17)-(22) correspond to the current configuration, i.e. $\Omega_e = \Omega_e(t_{n+1}) = \Omega^i_{n+1}$ **where n and i are the time step and non-linear iteration indices, respectively**. This domain configuration is defined by the nodal positions $\bar{\mathbf{x}}(t_{n+1})$. Thus, the governing equations system (Eqs. 15, 16) is non-linear and must be solved iteratively. The discrete operators must be updated at every non-linear iteration according to the newly obtained mesh position ($\bar{\mathbf{x}}^{i+1}_{n+1} = \bar{\mathbf{x}}^i_{n+1} + \delta\bar{\mathbf{d}}$).

Since we aim at developing a monolithic FSI solver, the fluid model should have the same degrees of freedom as the structural one. Thus we inject the approximation for the pressure increment to be used in the linearization of the momentum equation (Eq. 15) **resulting in an equations system that is to be solved exclusively for nodal displacements**. From Eq. (2) we can obtain [14]:

$$\delta\bar{\mathbf{p}} = \sum_{elem} \int_{t_n}^{t_{n+1}} \left(\int_{\Omega_e} \mathbf{C}_K \mathbf{B} \bar{\mathbf{d}}(t) d\Omega_e \right) dt \approx \left[\sum_{elem} \int_{\Omega_e} \mathbf{C}_K \mathbf{B} d\Omega_e \right] \bar{\mathbf{d}}_{n+1} \tag{23}$$

where $\delta\bar{\mathbf{p}} = \bar{\mathbf{p}}_{n+1} - \bar{\mathbf{p}}_n$. The strain matrix $\mathbf{B}$ and the volumetric constitutive matrix $\mathbf{C}_K$ are defined (in 2D) as

$$\mathbf{B} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{pmatrix} \qquad \mathbf{C}_K = \begin{pmatrix} \kappa & \kappa & 0 \\ \kappa & \kappa & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{25}$$

(24)

Making use of Eq. 23, the resulting approximation of the pressure gradient in Eq. (19) is:

$$\mathbf{G}\bar{\mathbf{p}}_{n+1} = \mathbf{G}\left(\bar{\mathbf{p}}_n + \delta\bar{\mathbf{p}}\right) \approx \mathbf{G}\bar{\mathbf{p}}_n + \left[\sum_{elem} \int_{\Omega_e} \mathbf{B}^T \mathbf{C}_K \mathbf{B} d\Omega_e \right] \bar{\mathbf{d}}_{n+1} \tag{26}$$

Hence, the linearization of the pressure gradient with respect to displacement can be expressed as:

$$\frac{\partial \mathbf{G}\bar{\mathbf{p}}}{\partial \bar{\mathbf{d}}} \approx \sum_{elem} \int_{\Omega_e} \mathbf{B}^T \mathbf{C}_K \mathbf{B} d\Omega_e \tag{27}$$

and the tangent matrix of the momentum equation reads

$$\mathbf{H} = \frac{\partial \bar{\mathbf{r}}_m}{\partial \bar{\mathbf{d}}} = \frac{\mathbf{M}}{\Delta t^2} + \frac{\mu \mathbf{L}}{\Delta t} + \sum_{elem} \int_{\Omega_e} \mathbf{B}^T \mathbf{C}_K \mathbf{B} d\Omega_e \tag{28}$$

1. Solve $\mathbf{H}\delta\bar{\mathbf{d}} = \bar{\mathbf{r}}_m(\bar{\mathbf{d}}^i_{n+1}, \bar{\mathbf{p}}^i_{n+1})$
   for $\delta\bar{\mathbf{d}}$, where $\delta\bar{\mathbf{d}} = \bar{\mathbf{d}}^{i+1}_{n+1} - \bar{\mathbf{d}}^i_{n+1}$
2. Update the nodal positions $\bar{\mathbf{x}}^{i+1}_{n+1} = \bar{\mathbf{x}}^i_{n+1} + \delta\bar{\mathbf{d}}$
3. Update the nodal pressure as $\mathbf{M}_p\bar{\mathbf{p}}_{n+1} = \mathbf{M}_p\bar{\mathbf{p}}_n + \kappa\mathbf{D}\bar{\mathbf{d}}_{n+1}$
4. Go to 1 until convergence in $\delta\bar{\mathbf{d}}$

Table 1: Implicit solution of the governing equations system: iterative procedure.

The iterative procedure applied to the solution of the governing equations (Eqs. 15, 16) can be summarized as follows:

Note that for large values of bulk modulus, pressure instability may manifest. In order to stabilize pressure, pressure update equation (Eq. 16) is modified in the present implementation as: $\bar{\mathbf{r}}_c = \kappa\mathbf{D}\bar{\mathbf{d}}_{n+1} - \left(\mathbf{M}^l_p\bar{\mathbf{p}}_{n+1} - \mathbf{M}^c_p\bar{\mathbf{p}}_n\right)$**. The historical and the present pressure values are thus multiplied by consistent and lumped mass matrices, respectively. This technique is similar to the pressure stabilization method proposed by P. Bochev [27]. It is explained in detail in the context of quasi-incompressible formulations in [3], [12]. Thus, Step 3 in the algorithm above reads:** $\mathbf{M}^l_p\bar{\mathbf{p}}_{n+1} = \mathbf{M}^c_p\bar{\mathbf{p}}_n + \kappa\mathbf{D}\bar{\mathbf{d}}_{n+1}$

Element inversion

When solving the equations summarized in Table 1, a problem arises when the displacements $\bar{\mathbf{d}}^{i+1}_{n+1}$ become such that it leads to an element inversion. This typically happens already at the first iteration ($\bar{\mathbf{d}}^1_{n+1}$) whenever the time step size is large.

This critical case implies that the area of an element becomes zero due to the movement of its nodes defined by $\bar{\mathbf{d}}^{i+1}_{n+1}$. Noting that the **determinant of an** elemental Jacobian is equal to twice the elemental area ($det\mathbf{J} = 2A_{el}$ **for triangular elements**), one can relate the element degradation to having a zero Jacobian and thus compute the critical time step. Thus, the degradation condition then reads

$$A_{el} = \frac{1}{2}det\mathbf{J} = \frac{1}{2}det\left(\frac{\partial\bar{\mathbf{x}}_{n+1}}{\partial\bar{\mathbf{x}}_n}\right) = 0 \tag{29}$$

Knowing the position and the velocity of the nodes of an element at time step t_n one can estimate the position of these nodes at time step t_{n+1}.

Assuming a first order prediction, that is: $\bar{\mathbf{x}}_{n+1} \approx \bar{\mathbf{x}}_n + \bar{\mathbf{v}}_n \Delta t$ and expanding Eq. (29) we arrive at

$$det\mathbf{J} = det\left(\frac{\partial \bar{\mathbf{x}}_{n+1}}{\partial \bar{\mathbf{x}}_n}\right) = det\left(\frac{\partial \bar{\mathbf{x}}_n}{\partial \bar{\mathbf{x}}_n} + \Delta t \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{\mathbf{x}}_n}\right) = det\left(\mathbf{I} + \Delta t \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{\mathbf{x}}_n}\right) \quad (30)$$

where $\mathbf{I}$ is the identity matrix. To find the critical time step, one must solve the equation: $det\left(\mathbf{I} + \Delta t_{crit} \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{\mathbf{x}}_n}\right) = 0$ for Δt_{crit}.

One can show that the critical time step determined by above equation is related to the elemental Courant number:

$$\begin{aligned}
det\left(I + \Delta t_{crit} \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{x}_n}\right) = det\left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \Delta t_{crit} \begin{pmatrix} \bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x} & \bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial y} \\ \bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial x} & \bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial y} \end{pmatrix}\right] = \\
det\left[\begin{pmatrix} 1 + \Delta t_{crit}\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x} & \Delta t_{crit}\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial y} \\ \Delta t_{crit}\bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial x} & 1 + \Delta t_{crit}\bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial y} \end{pmatrix}\right] = \\
\left(1 + \Delta t_{crit}\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x}\right)\left(1 + \Delta t_{crit}\bar{\mathbf{v}}_{Jy}\frac{\partial N_J}{\partial y}\right) - \\
\left(\Delta t_{crit}\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial y}\right)\left(\Delta t_{crit}\bar{\mathbf{v}}_{Jy}\frac{\partial N_J}{\partial x}\right) = \\
1 + \Delta t_{crit}\bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial y} + \Delta t_{crit}\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x} + \\
\Delta t_{crit}^2\left(\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x}\bar{\mathbf{v}}_{Jy}\frac{\partial N_J}{\partial y} - \bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial y}\bar{\mathbf{v}}_{Jy}\frac{\partial N_J}{\partial x}\right) = \\
1 + \Delta t_{crit}\bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x} + \Delta t_{crit}\bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial y} + O(t^2) \quad (31)
\end{aligned}$$

where I and J are the nodal indices. **Note that Einstein's summation convention is used.**

Thus, **neglecting the second order terms**, the critical time step can be determined as:

$$\Delta t_{crit}\left(\bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial y} + \bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x}\right) = -1 \quad (32)$$

Noting that the Courant number C can be computed as

$$C = \Delta t\left|\bar{\mathbf{v}}_{Iy}\frac{\partial N_I}{\partial y} + \bar{\mathbf{v}}_{Ix}\frac{\partial N_I}{\partial x}\right| \quad (33)$$

one concludes that the element degradation (zero Jacobian) corresponds to $C = 1$, meaning that a node can bypass the entire element within one time step.

Experience shows that the danger of inversion is highest for the elements encountered in the vicinity of the homogeneous Dirichlet boundary. This is illustrated in Fig. 2. An element with a bottom node belonging to the fixed boundary and two free nodes is displayed. Let us consider that for the given time step size the upper node has a displacement $\bar{\mathbf{d}}_3$ much larger than that of the middle node $\bar{\mathbf{d}}_2$. Thus within one time step the middle node "snaps through", the element becomes inverted and the solution diverges. As already mentioned, this can be avoided in the standard Lagrangian schemes only by reducing the time step size, leading to lengthy computational times.

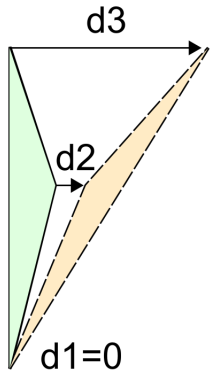


Figure 2: Element inversion.

In order to alleviate this problem we propose to predict the current configuration of the domain Ω^p_{n+1} (defined by the nodal positions $\bar{\mathbf{x}}^p_{n+1}$ and the corresponding connectivities, where super-index p stands for "prediction") by using an explicit time integration of the nodal position prior to beginning of the implicit solution (Table 1). This is explained next.

2.3. The eXplicit Integration following the Velocity Streamlines (X-IVS) for predicting the domain configuration Ω^p_{n+1}

By definition, a streamline is a curve that is instantaneously tangent to the velocity vector of the flow. The streamlines show the direction in which a massless fluid element will travel at any point in time. If one considers the streamline passing through a given node of the Lagrangian mesh, the streamline predicts where this node will move.

The idea of the X-IVS method proposed in [15] is to use the the velocity streamlines obtained at time step t_n to approximate the position of a particle (coinciding in our case with a node of the computational mesh) at t_{n+1} using the following expression:

$$\bar{\mathbf{x}}^p_{n+1} = \bar{\mathbf{x}}_n + \int_{t_n}^{t_{n+1}} \mathbf{v}_n \left(\mathbf{x}_\tau\right) d\tau. \tag{34}$$

where $\mathbf{x}_\tau$ is the function describing the movement of the particle from its position at t_n to that at t_{n+1} (τ: $t_n < \tau < t_{n+1}$).

Once the field $\mathbf{v}_n$ is discretized using piece-wise linear functions supported by the mesh **($\mathbf{v}_n = \sum_{I=1}^{3} \bar{\mathbf{v}}_I \mathbf{N}_I$, where I is the nodal index of the element where the particle is located)** at $t = t_n$ Eq. 34 can be written as

$$\bar{\mathbf{x}}^p_{n+1} = \bar{\mathbf{x}}_n + \left[\int_{t_n}^{t_{n+1}} \mathbf{N} \left(\mathbf{x}_\tau\right) dt\right] \bar{\mathbf{v}}_n. \tag{35}$$

Eq. (35) is linear and explicit in time. Only the information at time step t_n is used. This may be integrated analytically or numerically using any standard time integration schemes of high accuracy like explicit Runge-Kutta, or alternatively a sub-stepping technique. In this work the sub-stepping method is implemented. This is not an expensive operation taking into account that computations are explicit and then each particle may be evaluated separately from each other using a parallel computer.

Let us consider a mesh discretizing the domain of interest at t_n (see Fig. 3). At the beginning of the new time step t_{n+1} the historical velocity at the mesh nodes is known $\bar{\mathbf{v}}_n$. Let us consider a particle (marked by a red dot in Fig. 3) that at t_n coincides with a mesh node. The movement of this particle can be tracked by dividing the time step $\Delta t = t_{n+1} - t_n$ into a series of sub-steps ($\delta t = \frac{\Delta t}{m}$, where m is the number of sub-steps) and evaluating Eq. 35 incrementally as (see Fig. 3 where X-IVS integration using 6 sub-steps is schematically shown):

$$\bar{\mathbf{x}}^p_{n+1} = \bar{\mathbf{x}}_n + \sum_{i=1}^{m} \bar{\mathbf{v}}_n \left(\mathbf{x}_{n+\frac{i}{m}}\right) \delta t. \tag{36}$$

Note that for each sub-step one must identify the element where the node is located and use the corresponding shape functions and velocity values in Eq. 36.

Once all the sub-steps are completed, the prediction of the final position $\bar{\mathbf{x}}^p_{n+1}$ of the particle is obtained (see Fig 3). The segments connecting the intermediate positions of the particle at each sub-step δt define an approximation of the streamline originating from the selected node.

Note that the particles can move across several elements and through the free surface during a time step. If a particle crosses the free surface, then it leaves the streamline and follows a trajectory defined by the acting forces, being the simplest one the parabolic motion (due to gravity force only) or coupled with a water droplet drag model. An extended description of this technique may be consulted in [17].

Once X-IVS integration is applied to all the mesh nodes, an approximation for the new configuration Ω^p_{n+1} is obtained by creating a mesh connecting these nodes. The generation of the FE mesh is done using a Delaunay triangulation/tessellation [28]. For more details on the mesh generation applied in Lagrangian fluid formulations the reader is referred to [29] or [30].

The configuration Ω^p_{n+1} provides the first approximation that can be used then to solve the governing equations (Eqs. 13, 14) implicitly according to the algorithm presented Table 1. The implicit solution yields only the correction $\delta\bar{\mathbf{d}}$ for the position of the nodes, rather than $\bar{\mathbf{d}}_{n+1}$ (as it would be in case of using a standard scheme, i.e. without X-IVS prediction).

Thus, if X-IVS approximation of the new domain configuration is accurate, theoretically, one can work with arbitrarily large time steps without the danger of element inversion since X-IVS step convects the nodes with no connectivities and only then the mesh is created. Solving the governing equations implicitly ensures that the mesh configuration is corrected iteratively until reaching the true end-of-step position satisfying the governing equations. This feature is a strong advantage of the present approach over the previously proposed X-IVS-based Lagrangian models [17], [31] where the nodal positions predicted by X-IVS were not corrected. Moreover, using the displacement rather than velocity as the primary variable for the fluid domain facilitates the monolithic coupling with a large number of structural elements typically described in terms of displacements, **such as 2D and 3D solids, membranes, rotation-free shells or trusses**. In the following, the algorithm combining the proposed fluid model with displacement-based solid models is outlined.

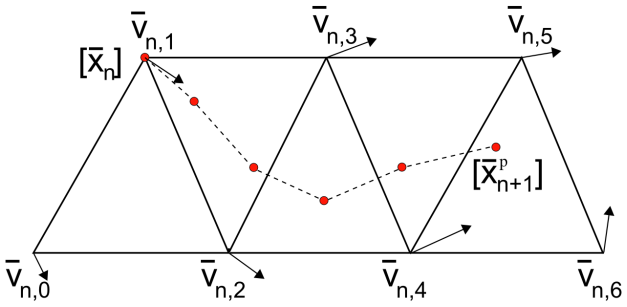


Figure 3: X-IVS streamline integration considering 5 sub-steps.

2.4. Fluid-structure interaction (FSI)

The presented approach can be easily incorporated into the FSI monolithic schemes presented in our previous works [3], [1], [13], [32], [7].

Let us consider a generic structural element. It can be a solid (2D or 3D), membrane, rotation-free shell ([33]) or any other FE structure. The only prerequisite is that the primary variable of the structural element must be the displacement. Note, that the time integration scheme chosen for the solid must be coincident with the one used for the fluid in order to define a monolithic scheme.

The discrete momentum equations for a solid in the absence of damping, using backward Euler time integration scheme can be written as

$$\mathbf{M}\bar{\mathbf{a}}_{n+1} + \mathbf{K}\bar{\mathbf{d}}_{n+1} = \mathbf{F}_{n+1} \tag{37}$$

where $\mathbf{K}$ is the stiffness matrix (the rest of matrices and vectors follow previously introduced definitions).

For applying a non-linear iteration procedure we define the dynamic residual and tangent stiffness

$$\bar{\mathbf{r}}_s = \mathbf{F} - \mathbf{M}\bar{\mathbf{a}}^i_{n+1} - \mathbf{K}\bar{\mathbf{d}}^i_{n+1} \tag{38}$$

$$\mathbf{H}_s = \frac{\partial \bar{\mathbf{r}}_s}{\partial \bar{\mathbf{d}}} \tag{39}$$

where subscript s stands for "structure". The tangent matrix and the residual corresponding to the fluid domain (given by Eq. 15. and Eq. 28, respectively) will be distinguished by subscript f.

The linearized monolithic FSI equations system is obtained by a standard FE assembly procedure, i.e. looping over all the elements (fluid and

structure). Structural elements contribute $\bar{\mathbf{r}}_s$ and $\mathbf{H}_s$ whereas fluid elements contribute $\bar{\mathbf{r}}_f$ and $\mathbf{H}_f$ to the unique FSI dynamic residual and tangent stiffness $\bar{\mathbf{r}}_{FSI}$ and $\mathbf{H}_{FSI}$, respectively. Nodes shared by the fluid and the solid contain the sum of the respective fluid and solid contributions.

The implementation procedure of the model for FSI problems is summarized in Table 2.

1. Find the approximation for the position of the mesh at time t_{n+1} as:
   - Use the X-IVS integration (Eq. 35) in the fluid domain. Result: $\bar{\mathbf{d}}^p_{n+1}$, $\mathbf{x}^p_{n+1}$.
   - Use a Forward Euler approximation for the nodal positions in the solid domain.
2. Re-mesh the fluid domain
3. Using $\bar{\mathbf{d}}^p_{n+1}$ compute the prediction for the fluid pressure $\bar{\mathbf{p}}^p_{n+1}$ (see step **2** in Table 1).
4. Start the non-linear loop (until convergence in $\delta\bar{\mathbf{d}}$)
   - Construct the monolithic FSI momentum equation using residual and tangent matrices defined by: Eqs. 38, 39 (for structural elements) and Eqs. 15, 28 (for fluid elements)
   - Solve the FSI momentum equation for $\delta\bar{\mathbf{d}}$. Compute: $\bar{\mathbf{d}}^{i+1}_{n+1}$, $\bar{\mathbf{x}}^{i+1}_{n+1}$
   - Move mesh according to $\bar{\mathbf{x}}^{i+1}_{n+1}$
   - Update fluid pressure. Result: $\bar{\mathbf{p}}^{i+1}_{n+1}$
5. Go to the next time step

Table 2: Implementation procedure of the displacement-based FSI formulation with explicit streamline integration prediction.

3. Examples

The examples chosen validate the model and compare it with the former approaches. Particular attention is paid to the choice of the time step size. The method is applied to both fluid and fluid-structure interaction problems.

The model was implemented within KRATOS Multi-Physics code, a C++ object oriented FE framework developed at CIMNE [34]. The convergence criteria for the non-linear iterations were set as: $\delta\bar{\mathbf{d}}_{n+1} \leq 10^{-9}$ (absolute tolerance) and $\frac{\delta\bar{\mathbf{d}}_{n+1}}{\bar{\mathbf{d}}_{n+1}} \leq 10^{-6}$. Conjugate Gradient (CG) linear solver was used to solve the linearized equations at each non-linear iterations. The CG tolerance was set to 10^{-9}.

3.1. Flow between two parallel plates

To validate the method a simple example dealing with a steady laminar flow between two parallel plates is chosen. For this test the analytic solution is known. The problem settings are taken from [6]. The test is sketched in Fig. 4. The fluid is moving in the horizontal direction parallel to the plates that have length L=10 m and are separated by the distance D=1 m. The fluid properties are: density $\rho = 1000$ kg/m^3, dynamic viscosity $\mu = 10^4\ Pa \cdot s$. The bulk modulus used for modeling the nearly incompressible behavior of the fluid was set to $\kappa = 10^7$ Pa [3]. A uniform pressure of 160000 Pa is applied at the inlet **nodes as a force term equal to the pressure multiplied by the normal to the inlet at a given node**. The analytic solution for the distribution of the horizontal velocity component in the vertical direction (along the y coordinate) is given by

$$u = \frac{1}{2\mu}\frac{\partial p}{\partial x}\left(y^2 - \frac{D^2}{4}\right) \tag{40}$$

The pressure gradient $\frac{\partial p}{\partial x} = \frac{P}{L} = 16000$ Pa/m.

The problem is discretized with an unstructured and nearly uniform mesh of size $h = 0.05$ m. Total simulation time is set to 1 s.

Fig. 5 shows the velocity along the cross-section at x=5 m. Result of the numerical simulation carried out using time step $Dt = 0.001$ s is compared

[3] **Note that for approximating the incompressible behavior the bulk modulus of the fluid $\kappa = 10^7$ Pa was used (if not mentioned otherwise), which is two orders of magnitude smaller than the physical value. Chosen value is sufficiently large for obtaining negligible variation of volume, but, on the other hand, small enough so as to prevent poor system conditioning. This typically manifests when using physical value of bulk modulus and large time steps due to the domination of the ill-conditioned term in the tangent matrix over the well-conditioned mass matrix scaled with a square inverse of the time step (see Eq. 28).**

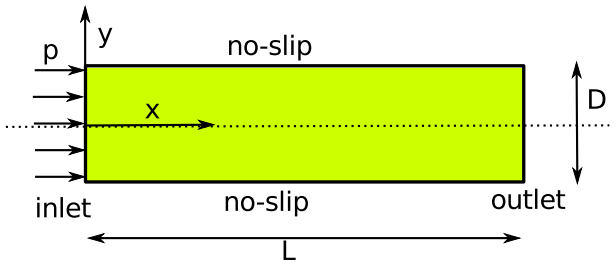
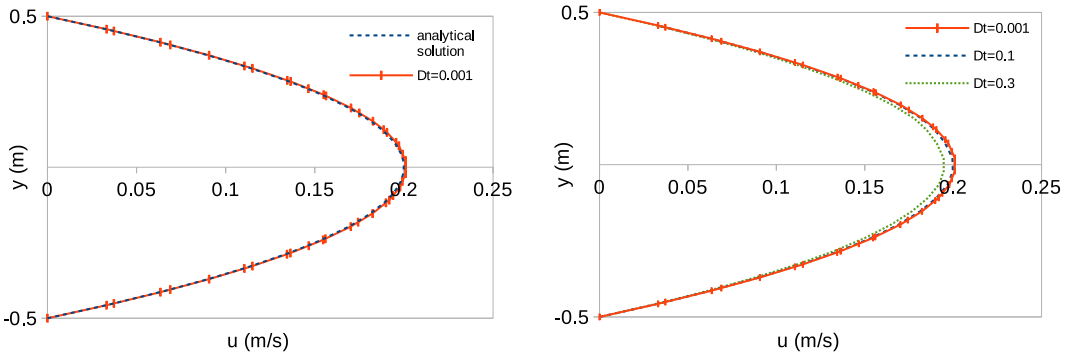


Figure 4: The model of viscous flow between two parallel plates

with the analytic solution given by Eq. 40 in Fig. 5(a). One can see that the solutions are practically coincident. Comparison of the results obtained using larger time steps is plotted in 5(b). One can see that only for $Dt = 0.3$ s some discrepancy with the reference solution can be distinguished.



(a) Comparison with the analytic solution

(b) Results obtained for different time step sizes

Figure 5: Velocity profile along the vertical coordinate at x=5 m.

Fig. 7 displays the error in horizontal velocity for different time step sizes. The error was computed as an integral of the difference between the numerical and the analytic solution for the horizontal velocity at t=1 s along the vertical cut at x=0.5. One can see that second order of convergence is obtained.

Note that as reported in [6], where the problem is solved with a similar Lagrangian methodology, but without the explicit streamline integration prediction, the time step size used was $Dt = 0.001$ s. The technique presented here, but without X-IVS prediction could converge up to $Dt = 0.01$ s.

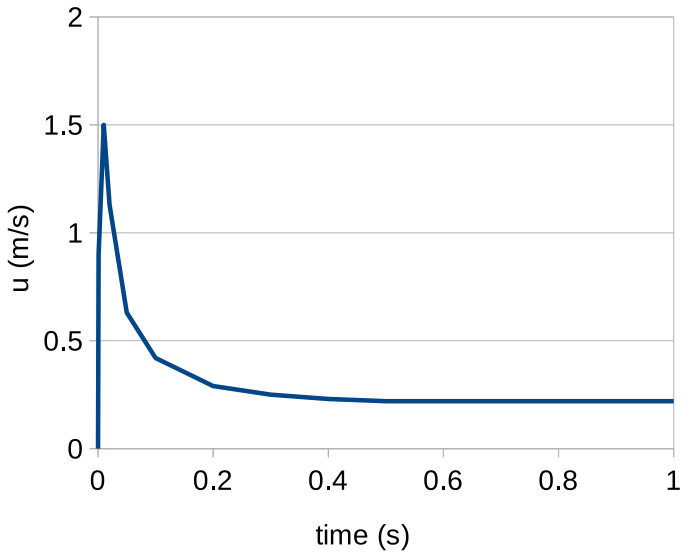


Figure 6: Horizontal velocity evolution at the middle of the inlet

This appears surprising at the first glance, as the steady state the maximum velocity is of order $u \approx 0.2$ m/s, meaning that a critical time step (corresponding to $C \approx 1$) must be $Dt_{crit} = 0.25$ s. However in the transient stage of the simulation large velocity develops in the vicinity of the inlet ($u \approx 2$ m/s, see Fig. 6), which provides the upper bound of $Dt \leq 0.025$ s in the transient stage for the standard Lagrangian formulation. One can appreciate that in the method equipped with the X-IVS prediction one could accommodate the time step up to $Dt = 0.3$ s, which is 10 times larger than the theoretical critical size for the standard method, and 30 times larger than the actual time step size the standard method can accommodate.

The number of non-linear iterations necessary for reaching the convergence for different time step sizes is summarized in Table 3. We also provide the data obtained by using the standard method (i.e., without X-IVS prediction). One can see that for small time steps both the method proposed here and the standard one show nearly equivalent convergence characteristics. For a large time step size the standard method diverges due to the element inversion. The proposed method provides convergent results up to $Dt = 0.3$ s. However, for such time step the number of iterations per time step becomes excessive. Best results in terms of convergence speed were observed for 0.1 s $\leq Dt \leq 0.2$ s.

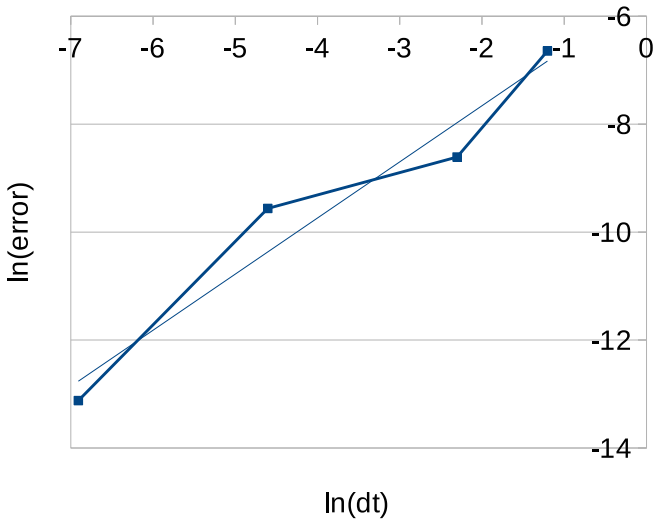


Figure 7: Error in the horizontal velocity along the cross-section at $x = 5$ m vs. time step size (logarithmic scale)

Data	Present method		Standard method	
Dt	N-l it.tot	N-l it./time step	N-l it.tot	N-l it./time step
0.001	1002	1	1003	1
0.01	140	1.4	154	1.5
0.1	81	8	-	-
0.2	83	16	-	-
0.3	105	35	-	-

Table 3: Example 3.1. Convergence characteristics of displacement-based formulations with and without X-IVS prediction for different time step sizes

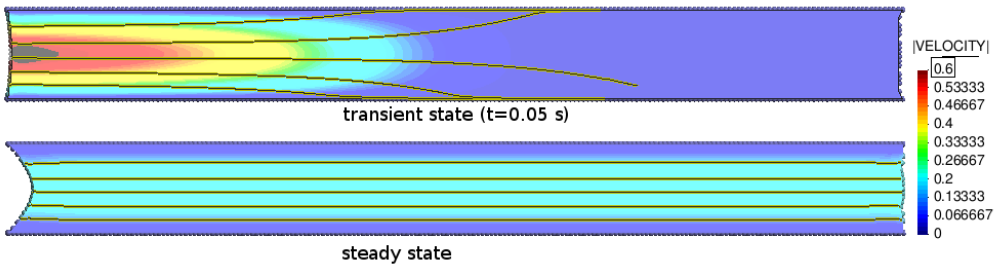


Figure 8: Streamlines during the transient stage and at steady state.

Fig. 8 shows the streamlines in the fluid domain during the transient stage ($t = 0.05$ s) and at steady-state. One can see that at the steady-state the streamlines are nearly parallel to the walls.

3.2. Sloshing in an elastic container

This example analyzes the fluid sloshing in an elastic container. The bulk modulus and the density of the fluid are $\kappa = 10^7$ Pa and $\rho_f = 1000$ kg/m^3, the dynamic viscosity $\mu = 10^{-2}\ Pa \cdot s$. As shown in Fig. 9 the width L and the height H of the internal part of the tank are 1.4 m and 2.6 m, respectively. The **thickness** of the tank walls t is 0.5 m. The properties of the solid are: Young's modulus $E = 10^6$ Pa, Poisson's ratio $\nu = 0.1$, density $\rho_s = 2500$ kg/m^3. The test is adapted from [6] with modifications of the constitutive model (here linear elastic law is used, while elasto-plastic model is applied in the mentioned work). The walls are fixed in the left and right lower corners as indicated by a the solid diagonal line in Fig. 9.

The problem is discretized by a uniform unstructured computational mesh with element size $h = 0.035$ m, leading to ca. 5600 nodes and 10000 triangular elements. **The simulation was performed for a time span of 2.5 s.**

Fig. 10 displays the evolution of the fluid-structure domain in time as well as the pressure distribution.

Fig. 11 displays time evolution of displacements at different location of the container: the middle of the bottom wall and the left and right upper corners of the vertical walls. Fig. 11(a) shows the vertical displacement histories at the middle of the bottom structure obtained using small time step size ($Dt = 0.001$ s). The results of the standard (no streamline prediction) and the proposed (with streamline prediction) Lagrangian methods are com-

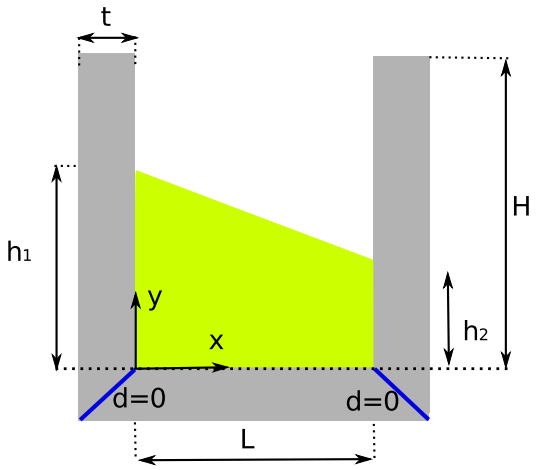
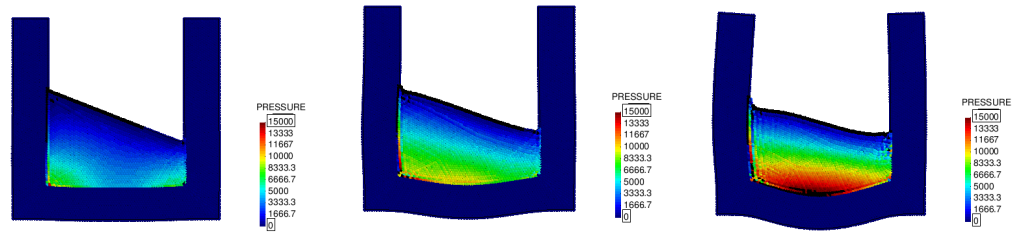
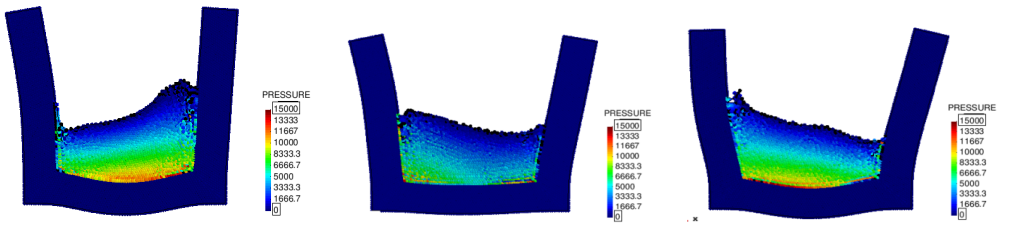


Figure 9: The model of water sloshing in a elastic container



(a) t=0.1 s (b) t=0.2 s (c) t=0.3 s

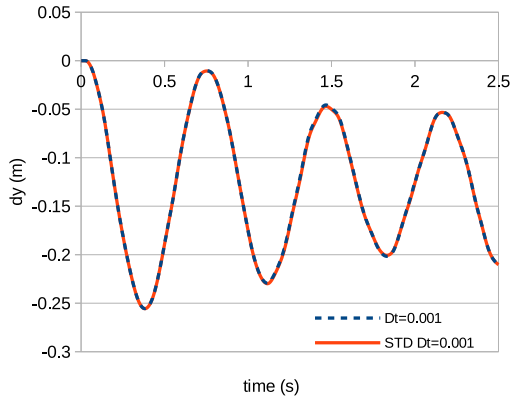


(d) t=1.0 s (e) t=1.5 s (f) t=2.0 s

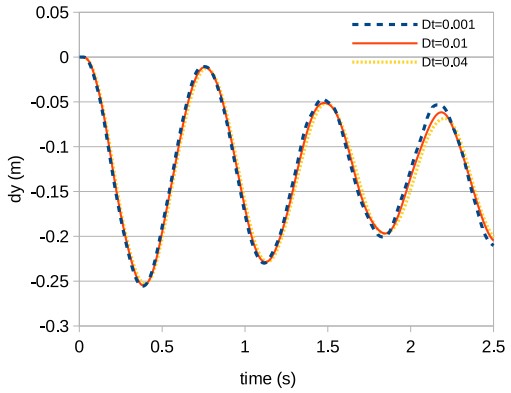
Figure 10: Snapshots of water sloshing in the elastic tank. Time step used: Dt=0.04 s

pared. One can see that the results are identical. Taking into account that for the considered problem the maximum velocity observed is of the order of 2 m/s, $Dt = 0.001$ s corresponds to Courant number $C \approx 0.06$, which is sufficiently small to ensure that the standard methodology works. The solution obtained using $Dt = 0.001$ s will be considered a reference solution.

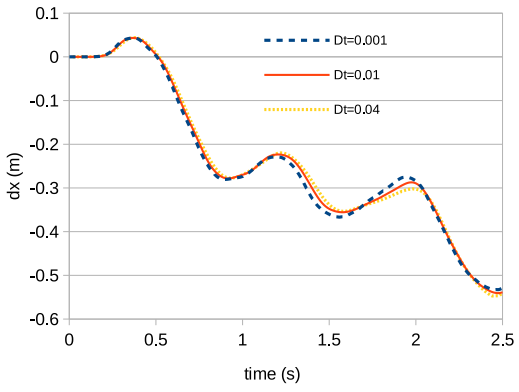
Fig. 11(b) shows comparison of the vertical displacement evolution obtained using much larger time steps: $Dt = 0.02$ s ($C \approx 1.2$) and $Dt = 0.04$ s ($C \approx 2.4$), where standard methodology fails. One can see that a very good agreement with the reference solution is observed. For larger time steps convergence could not be obtained. Fig. 11(c) and Fig. 11(d) present displacement histories of the upper left and upper right corners of the vertical columns, respectively. The results obtained using different time steps are displayed.



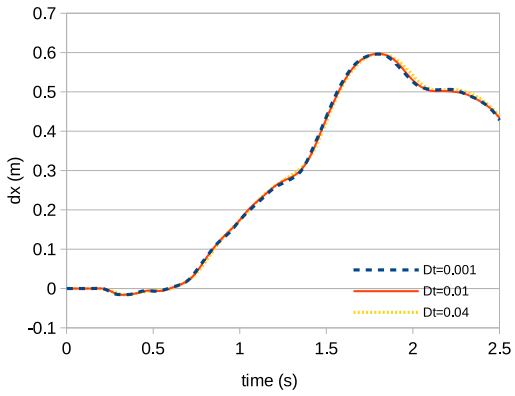
(a) Displacement evolution of the bottom wall: comparison with the standard formulation



(b) Displacement evolution of the bottom wall for different time step sizes



(c) Displacement evolution of the left upper corner for different time step sizes



(d) Displacement evolution of the right upper corner for different time step sizes

Figure 11: The deflection of various container parts: vertical deflection of the mid-bottom, and horizontal deflections of the upper left and upper right corners.

Data		Present method		Standard method	
dt	C	N-l it.tot	N-l it./time step	N-l it.tot	N-l it./time step
0.001	0.085	**2560**	2	**2562**	2
0.01	0.85	**801**	3.2	**1175**	3.3
0.02	1.7	**728**	6	-	-
0.04	3.4	**1490**	24	-	-
0.1	-	-	-	-	-

Table 4: Example 3.2. Convergence characteristics of displacement-based formulations with and without X-IVS prediction for different time step sizes

The number of non-linear iterations necessary for obtaining convergence for different time step sizes is summarized in Table 4. We also provide the data obtained by using the standard method (the one without streamline integration prediction). One can see that for small time steps both the method proposed here and the standard one show similar convergence features. **We note that the first column of the table provides the maximum time step size. In case of the present method with X-IVS prediction, the maximum step size is equal to the actual time step. In case of standard technique, the time step was estimated using the criterion based on the determinant of the elemental Jacobian. Whenever for a given element the element degradation or inversion was expected the actual time step was reduced. While for $Dt = 0.001$ s standard method did not require to reduce the actual time step, for $Dt = 0.01$ s it was the case on multiple occasions. This led to an overall of 356 time steps instead of 250 for simulating the time span of 2.5 s.**

For large time step size the standard method diverged once element inversion took place. The proposed method provides convergent results up to Dt=0.04 s, however for such time step the number of iterations per time step becomes excessive. Best results in terms of convergence speed are exhibited for 0.01 s $\leq dt \leq$0.02 s

In order to assess the gain due to using X-IVS prediction, computational time corresponding to different solution steps is shown next. Table 5 summarizes the data obtained when solving Example 3.2 using a mesh containing ca. 5000 nodes. The data provides average cost per time step. Once can see that when using small time step (Dt=0.001) that ensures convergence in a single iteration, the relative cost of X-IVS step is around 14.5 %. For the time step size identified as optimal, the relative cost of the prediction step

Dt	System solve	Re-mesh.	X-IVS pred.	Total	X-IVS rel. cost
0.001 s	0.16 s	0.07 s	0.04 s	0.27 s	14.5 %
0.01 s	0.45 s	0.07 s	0.04 s	0.56 s	7.14 %

Table 5: Example 3.2. Time consumption of different solution steps. Mesh1: 5600 nodes, 10000 elements.

Dt	System solve	Re-mesh.	X-IVS pred.	Total	X-IVS rel. cost
0.001 s	0.6 s	0.26 s	0.14 s	1 s	14 %
0.01 s	2.7 s	0.26 s	0.14 s	3.1 s	4.5 %

Table 6: Example 3.2. Time consumption of different solution steps. Mesh2: 17000 nodes, 33000 elements.

decreases to less than 10 %. Overall, X-IVS prediction cost is approximately twice smaller than the cost of re-meshing. Table 6 corresponds to a solution obtained on a finer mesh (ca 17000 nodes). This simulation confirms previous observations.

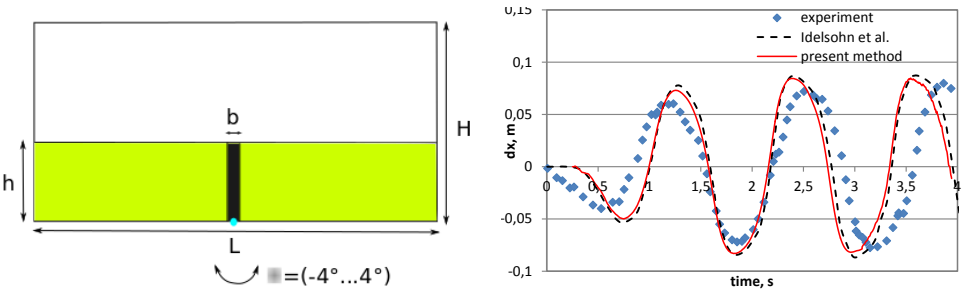
3.3. Shallow oil sloshing in a rigid container with a vertical elastic beam

This test case was analyzed both experimentally and numerically in [35] and further studied in [14]. The benchmark models the rotational motion of a rectangular container filled with liquid. A vertical elastic beam is clamped at the bottom of the container. The geometry of the model is shown in Fig. 12(a). The tank has a length $L = 0.609$ m and a height $H = 0.3445$ m. The container moves around a fixed point located in the mid-point of the bottom wall ($x = 0.3045$ m, $y = 0$ m). The motion with an amplitude of $\phi = 4^\circ$ and a period $T = 1.21$ s is prescribed to the container walls. The beam is made of polyurethane resin with the following properties: density is $\rho_s = 1100$ kg/m3 and Young modulus $E = 6$ MPa. The beam thickness is $b = 0.004$ m.

The tank is filled with sunflower oil, with the density of $\rho_f = 917\ kg/m^3$ and the kinematic viscosity of $\mu = 5e^{-5}\ m^2/s$. The bulk modulus of the fluid is set $\kappa = 10^7$ Pa. The original free surface level of the liquid coincided with the beam height ($h = 0.1148$ m). Note that in the experiment, when the motor is started there is a transition from the rest state to the harmonic motion due to inertia. To account for this a delay of 0.25 s in the onset of the tank motion was introduced in the numerical simulation. Uniform unstructured mesh with size of 0.003 m was used. In [35] maximum time step size was set to 0.0025 s, which corresponds to Courant number $C \approx 1$

for the given mesh size, taking into account that the maximum fluid velocity observed is of order of 1 m/s.

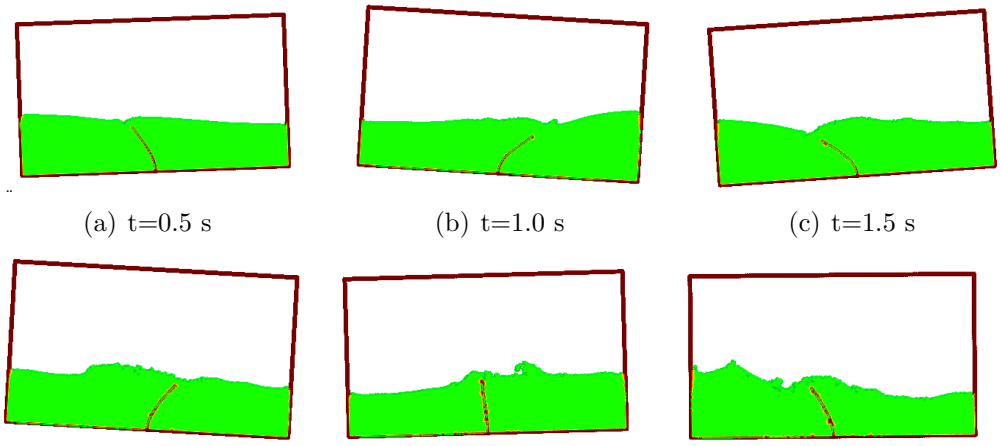
Here the example is solved with and without the X-IVS prediction, the latter corresponding the standard technique similar to the one employed in [35] and [14] .



(a) Scheme of the test

(b) Horizontal displacement of the beam's tip

Figure 12: The model of oil sloshing in the container with a vertical elastic beam



(a) t=0.5 s (b) t=1.0 s (c) t=1.5 s

(d) t=2.0 s (e) t=2.5 s (f) t=3.0 s

Figure 13: Snapshots of water sloshing in a rigid tank with an elastic beam. Time step used: dt=0.01 s (C=4)

Fig. 13 shows the snapshots of the simulation at 6 time instances. Fig. 14 shows the velocity streamlines at $t = 1.5$ s. One can see that the particle

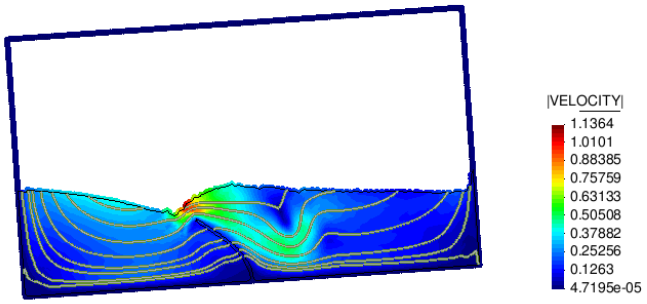


Figure 14: Streamlines at $t = 1.5$ s.

position prediction based upon the integration along the streamlines is particularly advantageous in the vicinity of the corners and next to the structure. There, simple predictions, such as the ones based on previous-step solution $\mathbf{v}_n \Delta t$ would lead to erroneous results moving the particles across the solid for large time steps.

Fig. 12(b) displays the evolution of the horizontal displacement d_x of the beam's upper left corner. The results obtained with the present method are compared with the experimental data and the numerical simulation reported in [35], [36]. One can see a good agreement with the experimental data and an almost exact match with the numerical results.

Next the time step size necessary for obtaining convergent results when using the standard technique (no X-IVS prediction) is recorded. In this simulation, when using no X-IVS prediction, a variable time step based on critical time step estimation was used. Fig. 15 shows the actual time step sizes used in the simulation without X-IVS prediction when maximum time step size was set to $Dt_{max} = 0.0025$ s. In order to ensure that no element becomes inverted, the actual time step size had to be reduced (the average actual time step was $Dt \approx 0.0015$). In case of using X-IVS prediction the constant time step was maintained.

Conclusions and outlook. In this paper a **Lagrangian** displacement-based fluid model has been proposed. The main novelty of the model consisted in combining the explicit integration for the motion of the nodes along the streamlines with a fully implicit correction **by solving the Navier-Stokes equations for displacement**. The streamline prediction for the nodal motion allowed to alleviate the severe time step size restrictions encountered

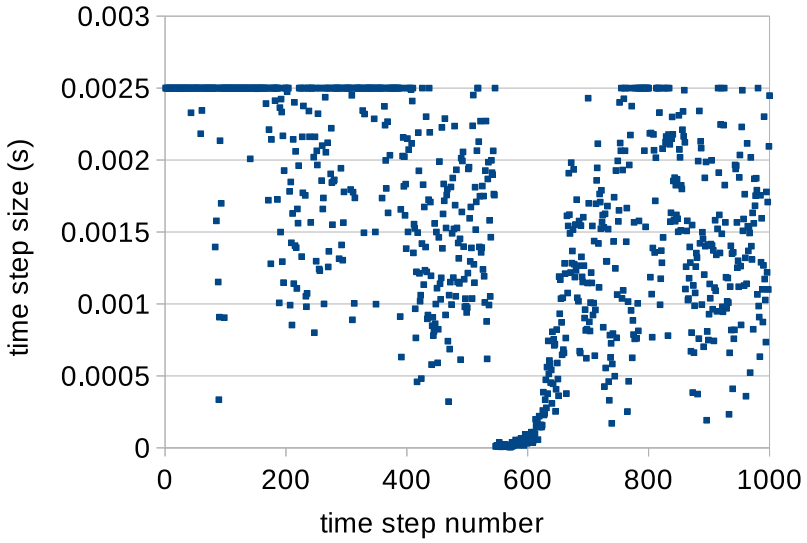


Figure 15: Actual time step used in the simulation without X-IVS prediction. Maximum time step size: 0.0025 s

when using former approaches **that did not include the X-IVS prediction**. On the other hand, using a displacement-based formulation instead of a velocity-based one has ensured that the mesh position at the end of each time step respects the governing equations.

Moreover, it was shown that **thanks to choosing displacement as the primary kinematic variable in the fluid domain** the proposed fluid formulation could been naturally coupled to displacement-based **elastic** solid formulations leading to a monolithic FSI scheme. The FSI scheme has proven to be efficient, leading to convergent solutions even when time steps **larger than those permissible in the formulations that do not include X-IVS prediction** have been used. It was discovered, however, that time steps may not be arbitrary large. In the considered examples for very large time steps the number of non-linear iterations per time step necessary to obtain convergence became prohibitively large. Nevertheless, the time steps that led to minimum overall number of iterations/per simulation in the problems considered were much larger than those of the previously proposed tightly coupled FSI approaches. It has been also shown that the proposed scheme allows employing fixed time step size, rather than adjusting it as it was done on the previous formulations even when the Courant number of the flow is

larger than 1.

The computational cost associated with the X-IVS prediction resulted to be small compared to that of the other solution steps (system solve and the re-meshing).

In the future, one must investigate a) how an optimal time step may be estimated apriori for a given case b) possibility of applying the proposed method to the problems involving non-elastic solids c) possible modifications of the method so as to account for truly incompressible behavior.

Acknowledgments. This work has been supported under the auspices of the FPDI-2013-18471 grant of the Spanish *Ministerio de Economia y Competitividad* as well as partially funded by the COMETAD project of the National RTD Plan (ref. MAT2014-60435-C2-1-R) of the mentioned ministry.

[1] Idelsohn S.R., Marti J., Limache A., and Oñate E. Unified Lagrangian formulation for elastic solids and incompressible fluids. application to fluid-structure interaction problems via the PFEM. Computer Methods in Applied Mechanics and Engineering, 197:17621776, 2008.

[2] Cremonesi M., Frangi A., and Perego U. A Lagrangian finite element approach for the analysis of fluid-structure interaction problems. International Journal Numerical Methods in Engineering, 84:610–630, 2010.

[3] Ryzhakov P., Rossi R., Idelsohn S., and Oñate E. A monolithic Lagrangian approach for fluid-structure interaction problems. Journal of Computational Mechanics, 46/6:883–399, 2010.

[4] Zhu M. and Scott M. Improved fractional step method for simulating fluid-structure interaction using the PFEM. International Journal for Numerical Methods in Engineering, 99(12):925–944, 2014.

[5] Zhu M. and Scott M. Direct differentiation of the quasi-incompressible fluid formulation of fluid–structure interaction using the PFEM. Computational Particle Mechanics, pages 1–13, 2016.

[6] Zhu M. and Scott M. Unified fractional step method for Lagrangian analysis of quasi-incompressible fluid and nonlinear structure interaction using the PFEM. International Journal for Numerical Methods in Engineering, 2016.

549 [7] Oñate E., Franci A., and Carbonell J. M. Lagrangian formulation
550 for finite element analysis of quasiincompressible fluids with reduced
551 mass losses. International Journal for Numerical Methods in Fluids,
552 74(10):699–731, 4 2014.

553 [8] Antoci C., Gallati M., and Sibilla S. Numerical simulation of fluid-
554 structure interaction by sph. Journal of Computers and Structures,
555 85:879–890, 2007.

556 [9] C Antoci. Simulazione numerica dellinterazione fluido-struttura con la
557 tecnica SPH. PhD thesis, Ph. D. thesis, Universita di Pavia. 100, 2006.

558 [10] Yang Q., Jones V., and McCue L. Free-surface flow interactions with
559 deformable structures using an SPH–FEM model. Ocean Engineering,
560 55:136–147, 2012.

561 [11] Potapov S., Maurel B., Combescure A., and Fabis J. Modeling
562 accidental-type fluid–structure interaction problems with the SPH
563 method. Computers & Structures, 87(11):721–734, 2009.

564 [12] Ryzhakov P., Oñate E., Rossi R., and Idelsohn. Lagrangian FE methods
565 for coupled problems in fluid mechanics. CIMNE edition, 2010.

566 [13] Ryzhakov P., Oñate E., Rossi R., and Idelsohn S. Improving mass
567 conservation in simulation of incompressible flows. Int. Jour. for
568 Num.Methds. in Eng., 2012. early view, published online 29/03/2012.

569 [14] P Ryzhakov. A modified fractional step method for fluid–structure in-
570 teraction problems. Revista Internacional de Métodos Numéricos para
571 Cálculo y Diseño en Ingeniería, 2016.

572 [15] Idelsohn S.R., Nigro N., Limache A., and Oñate E. Large time-step
573 explicit integration method for solving problems with dominant con-
574 vection. Computer Methods in Applied Mechanics and Engineering,
575 217220:168 – 185, 2012.

576 [16] Idelsohn S. R., Nigro N., Gimenez J., Rossi R., and Marti J. A fast and
577 accurate method to solve the incompressible navier-stokes equations.
578 Engineering Computations, 30(2):197–222, 2013.

[17] Idelsohn S. R., Marti J., Becker P., and Oñate E. Analysis of multifluid flows with large time steps using the particle finite element method. International Journal for Numerical Methods in Fluids, 75(9):621–644, 2014. FLD-13-0380.R1.

[18] Hron J. and Turek S. A monolithic fem/multigrid solver for an ale formulation of fluid-structure interaction with applications in biomechanics. In Fluid-structure interaction, pages 146–170. Springer, 2006.

[19] Barker A. and Cai X.-C. Scalable parallel methods for monolithic coupling in fluid–structure interaction with application to blood flow modeling. Journal of computational physics, 229(3):642–659, 2010.

[20] Walhorn E., Kolke A., Hubner B., and Dinkler D. Fluid-structure coupling within monolithic model involving free surface flow. Computers and Structures, 83:2100–2111, 2005.

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

[22] Bazilevs Y., Calo V.and Hughes T., and Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. Computational mechanics, 43(1):3–37, 2008.

[23] Heil M. and Hazel A.and Boyle J. Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches. Computational Mechanics, 43(1):91–101, 2008.

[24] Hou G., Wang J., and Layton A. Numerical methods for fluid-structure interaction a review. Commun. Comput. Phys, 12(2):337–377, 2012.

[25] Antoci C., M. Gallati, and Sibilla S. Numerical simulation of fluid–structure interaction by SPH. Computers & Structures, 85(11):879–890, 2007.

[26] Ryzhakov P., Cotela J., Rossi R., and E. Oñate. A two-step monolithic method for the efficient simulation of incompressible flows. International Journal for Numerical Methods in Fluids, 74(12):919–934, 2014.

609 [27] Pavel B Bochev, Clark R Dohrmann, and Max D Gunzburger. Stabiliza-
610 tion of low-order mixed finite elements for the stokes equations. SIAM
611 Journal on Numerical Analysis, 44(1):82–101, 2006.

612 [28] Delaunay B. Sur la sphre vide. Izvestia Akademii Nauk SSSR, Otdelenie
613 Matematicheskikh i Estestvennykh Nauk, 7:793–800, 1934.

614 [29] Calvo N., Idelsohn S.R., and Oñate E. The extended Delaunay tessel-
615 lation. Engineering Computations, 20:583–600, 2003.

616 [30] Oñate E., Idelsohn S., Del Pin F., and Aubry R. The Particle Finite
617 Element Method: an overview. International Journal of Computational
618 Methods, 1:267–307, 2004.

619 [31] Becker P., Idelsohn S. R., and Oñate E. A unified monolithic approach
620 for multi-fluid flows and fluid–structure interaction using the particle
621 finite element method with fixed mesh. Computational Mechanics,
622 55(6):1091–1104, 2015.

623 [32] Cremonesi M., Frangi A., and Perego U. A Lagrangian finite ele-
624 ment approach for the analysis of fluid–structure interaction problems.
625 International journal for numerical methods in engineering, 84(5):610–
626 630, 2010.

627 [33] Oñate E. and Zárate F. Rotation-free triangular plate and shell ele-
628 ments. International Journal for Numerical Methods in Engineering,
629 47(1-3):557–603, 2000.

630 [34] Dadvand P., Rossi R., and Oñate E. An object-oriented environment
631 for developing finite element codes for multi-disciplinary applications.
632 Archieves of Computational Methods in Engineering, 17/3:253–297,
633 2010.

634 [35] Idelsohn S.R., Marti J., Souto-Iglesias A., and Oñate E. Interaction
635 between an elastic structure and free-surface flows: experimental versus
636 numerical comparisons using the PFEM. Computational Mechanics,
637 43(1):125–132, 2008.

638 [36] Degroote J., Souto-Iglesias A., Van Paepegem W., Annerel S., Brugge-
639 man P., and Vierendeels J. Partitioned simulation of the interaction

between an elastic structure and free surface flow. Computer methods in applied mechanics and engineering, 199(33):2085–2098, 2010.