Towards a natural element Lagrangian strategy in fluid-structure interaction

David González, Andres Galavís, Elías Cueto, Francisco Chinesta, Manuel Doblare

TOWARDS A NATURAL ELEMENT LAGRANGIAN STRATEGY IN FLUID-STRUCTURE INTERACTION

D. González*, A. Galavís*, E. Cueto*, F. Chinesta†, M. Doblaré*

*Aragón Institute of Engineering Research. University of Zaragoza.
Betancourt Building. María de Luna, s.n. E-50018 Zaragoza, Spain.
ecueto@unizar.es
†LMSP UMR 8106 CNRS-ENSAM-ESEM.
151 Bvd. de l'Hôpital, F-75013 Paris, France.
francisco.chinesta@paris.ensam.fr

1 Introduction

Many efforts of research have been made in the Fluid-Structure Interaction (FSI) field in order to couple eulerian (or ALE) formulations in the fluid domain with lagrangian formulations in the solid domain. See, for instance, [8][16], just to cite a few. In some cases, however, the employ of lagrangian formulations for both domains seems to be interesting. These cases include —but they are not restricted to— the presence of free surfaces in the fluid, for instance, whose accurate description requires an additional technique (ALE, VoF, level set, among them) in order to capture the position of the free surface.

The development of meshless techniques in the nineties opened the possibility of employing lagrangian formulations for both the solid and fluid domains. This is so since meshless methods are less sensible to "mesh" distortion (i.e., relative nodal displacement) than finite element methods are. Thus, it is possible to employ an updated lagrangian strategy for the fluid domain, while employing a total or updated lagrangian strategy for the solid. This approach is very convenient for some classes of problems, especially those involving drastic changes in the fluid domain geometry.

Little has been done, however, in coupling meshless formulations for solids and fluids. In [12], for instance, the so-called Meshless Finite element Method was employed in order to simulate FSI problems from an updated Lagrangian perspective.

However, in this case the solid domain was modelled by using an hipoelastic formulation, and thus one can speak of two different fluids interacting rather than an actual FSI.

In this paper we describe mainly the formulation developed for the fluid from an updated lagrangian strategy. We employ the α-shape-based Natural Element Method (α-NEM) to this end. This formulation posses some advantages, that include an exact interpolation along the boundary [5], that allows for a standard, FE-like, treatment of the fluid-solid interface conditions. We first describe the basics of the α-NEM and then introduce the proposed numerical scheme for the integration of the Navier-Stokes equations in lagrangian form. Finally, we include some examples that demonstrate the accuracy of the proposed scheme and show the potential of the technique.

2 The Natural Element Method

2.1 Natural Neighbour interpolation

As mentioned before, the vast majority of meshless methods are based on the employ of scattered data approximation techniques to construct the approximating spaces of the Galerkin method. These techniques must have, of course, low sensitivity to mesh distortion, as opposed to FE methods. Among these techniques, the Natural Element Method employs any instance of Natural Neighbour interpolation [18] [10] to construct trial and test functions. Prior to the introduction of these interpolation techniques, it is necessary to define some basic concepts.

The model will be constructed upon a cloud of points with no connectivity on it. We will call this cloud of points $\boldsymbol{N} = \{n_1, n_2, \ldots, n_M\} \subset \mathbb{R}^d$, and there is a unique decomposition of the space into regions such that each point within these regions is closer to the node to which the region is associated than to any other in the cloud. This kind of space decomposition is called a Voronoi diagram of the cloud of points and each Voronoi cell is formally defined as (see figure 1):

$$T_I = \{\boldsymbol{x} \in \mathbb{R}^d : d(\boldsymbol{x}, \boldsymbol{x}_I) < d(\boldsymbol{x}, \boldsymbol{x}_J) \; \forall \; J \neq I\}, \tag{1}$$

where $d(\cdot, \cdot)$ is the Euclidean distance function.

The dual structure of the Voronoi diagram is the Delaunay triangulation[1], obtained by connecting nodes that share a common $(d-1)$-dimensional facet. While

[1] Even in three-dimensional spaces, it is common to refer to the Delaunay tetrahedralisation with the word *triangulation* in the vast majority of the literature

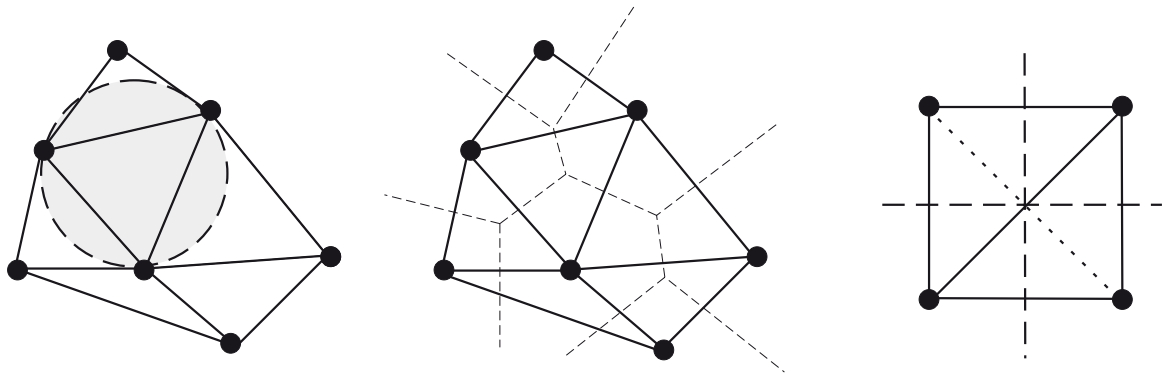


FIG. 1: *Delaunay triangulation and Voronoi diagram of a cloud of points.*

the Voronoi structure is unique, the Delaunay triangulation is not, there being some so-called *degenerate* cases in which there are two or more possible Delaunay triangulations (consider, for example, the case of triangulating a square in 2D). Another way to define the Delaunay triangulation of a set of nodes is by invoking the *empty circumcircle* property, which means that no node of the cloud lies within the circle covering a Delaunay triangle. Two nodes sharing a facet of their Voronoi cell are called *natural neighbours* and hence the name of the technique.

In order to define the natural neighbour co-ordinates it is necessary to introduce some additional concepts. The second-order Voronoi diagram of the cloud is defined as

$$T_{IJ} = \{\boldsymbol{x} \in \mathbb{R}^d : d(\boldsymbol{x}, \boldsymbol{x}_I) < d(\boldsymbol{x}, \boldsymbol{x}_J) < d(\boldsymbol{x}, \boldsymbol{x}_K) \; \forall \; J \neq I \neq K\}. \tag{2}$$

The most extended natural neighbour interpolation method, however, is the Sibson interpolant [17] [18]. Consider the introduction of the point $\boldsymbol{x}$ as one of the nodal points of the clouds. Due to this introduction, the Voronoi diagram will be altered, affecting the Voronoi cells of the natural neighbours of $\boldsymbol{x}$. Sibson [17] defined the natural neighbour coordinates of a point $\boldsymbol{x}$ with respect to one of its neighbours I as the ratio of the cell T_I that is transferred to T_x when adding $\boldsymbol{x}$ to the initial cloud of points to the total volume of T_x. In other words, if $\kappa(\boldsymbol{x})$ and $\kappa_I(\boldsymbol{x})$ are the Lebesgue measures of T_x and T_{xI} respectively, the natural neighbour coordinates of $\boldsymbol{x}$ with respect to the node I is defined as

$$\phi_I(\boldsymbol{x}) = \frac{\kappa_I(\boldsymbol{x})}{\kappa(\boldsymbol{x})}. \tag{3}$$

In Fig. 2 the shape function associated to node 1 may be expressed as

$$\phi_1(\boldsymbol{x}) = \frac{A_{abfe}}{A_{abcd}}. \tag{4}$$

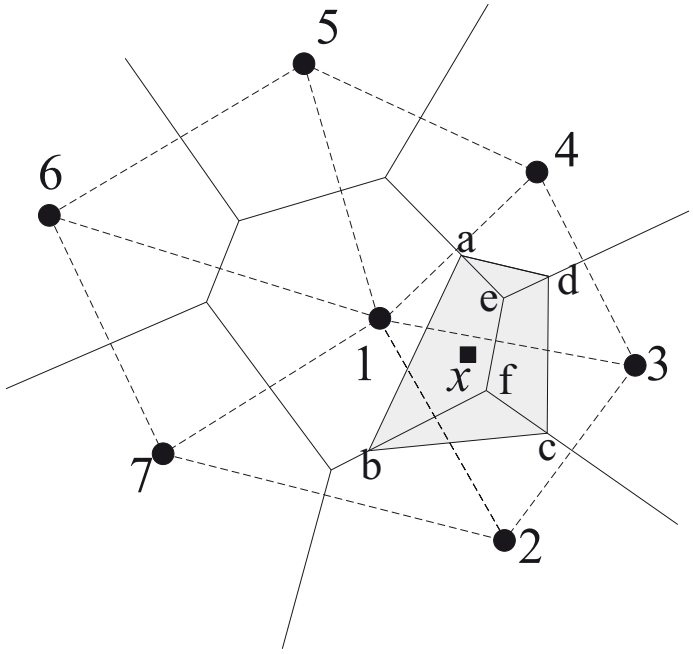


FIG. 2: *Definition of the Natural Neighbour coordinates of a point* $\boldsymbol{x}$.

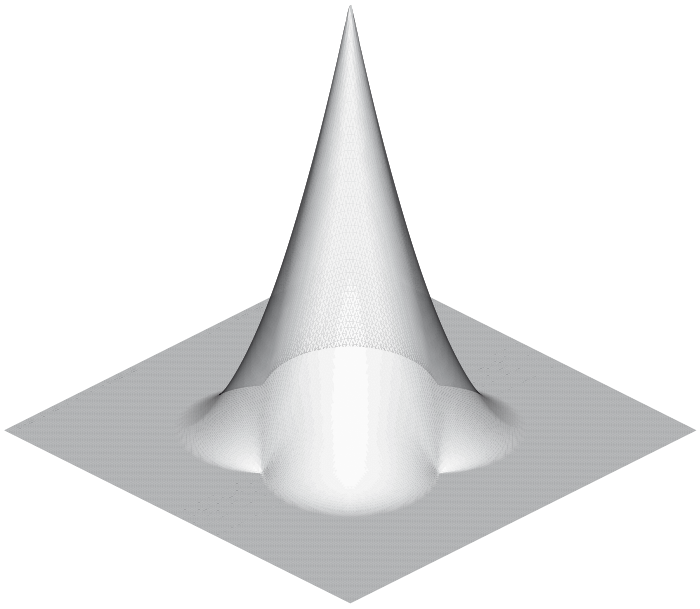


FIG. 3: *Typical function* $\phi(\boldsymbol{x})$. *Courtesy N. Sukumar.*

It is straightforward to prove that NE shape functions form a partition of unity [1], as well as some other properties like positivity (i.e., $0 \leq \phi_I(\boldsymbol{x}) \leq 1 \quad \forall I, \ \forall \boldsymbol{x}$) and interpolation:

$$\phi_I(\boldsymbol{x}_J) = \delta_{IJ}. \tag{5}$$

A second type of natural neighbour interpolation was independently established by Belikov [2] and Hiyoshi [10]. It is referred to as *non-Sibsonian* or *Laplace* interpolation. This kind of interpolation shares many properties with Sibson's coordinates (like positivity, interpolation and linear completeness), and is less computationally expensive, as noted in [20]. However, it is not as smooth as the previous one since the derivatives of the non-Sibsonian shape function are not defined along the edges of the Delaunay triangles that lie within its support. In this work, only Sibson and Thiessen interpolation have been considered.

It has been argued that Laplace interpolation provides "exact" —i.e., up to linear— interpolation along the boundary [20], although it was later demonstrated that this is not true in general [4]. Some of the most salient features of natural neighbour interpolation are studied in the following section.

2.2 Properties of Natural Neighbour interpolation

Sibson interpolants have some remarkable properties that help to construct the trial and test functional spaces of the Galerkin method (see [19], [10] for proofs of the following properties).

Besides properties like continuity and smoothness (everywhere except at the nodes for Sibson interpolants and at some other lines of zero measure for the Laplace interpolant), Sibson and Laplace interpolants posses linear completeness (i.e., exact reproduction of a linear field).

Sibson and Laplace interpolants can also reproduce linear functions exactly along convex boundaries. This is in sharp contrast to the vast majority of meshless methods. In addition, in [5] [4] [24] distinct methods of imposing linear displacement fields along non-convex boundaries were developed. These are based on the use of α-shapes, ε-samplings or visibility criteria, respectively. So, essential boundary conditions can be imposed directly, as in traditional Finite Element methods. In [5] and [3] it was demonstrated that the construction of the Sibson interpolant over an α-shape [7] of the domain allows us to accurately extract the shape of the domain, defined in terms of nodes only, while ensuring linear interpolation along any kind of boundaries (convex or not). This property was later generalised for arbitrary clouds of points and a explicit definition of the domain through CAD techniques in [4].

As mentioned before, Laplace interpolants were initially supposed to reproduce linear essential boundary conditions exactly [20], although it was later demonstrated that some criteria must be met in order to ensure it [4]. The α-shape approach mentioned before was later adopted in [11] in the so-called *meshless Finite Ele-*

ment method, which consists, essentially, in adopting FE approximation for well-shaped triangles or tetrahedra, and Laplace interpolation for badly-shaped tetrahedra grouped forming a polyhedron.

More recently, an approach based on a visibility criterion to ensure an appropriate interpolation along the boundary has been presented [24], proving successful results for instance, in the simulation of cracks, where an α-shape-based approach would need for a tremendous increase in the nodal sampling.

In the next section we study the implication of α-shapes in the development of the method here proposed.

3 The α-shapes-based Natural Element Method

The identification of the free surface in an updated Lagrangian flow simulation deserves some comments. In many prior works, location of boundary nodes is performed by flagging coincident element faces [13], for instance. Once the updating of nodal positions has been performed, a recursive check must be done in order to find overlapping boundary segments, thus generating "air" bubbles, holes or cavities in the domain, splashing drops, etc. In three dimensions this technique is obviously much more expensive. Splashing and similar phenomena is usually not considered with this approach.

With the irruption of meshless methods, in which models are constructed by a set of nodes only, boundary tracking can be performed by employing different strategies. In particular, we have employed *shape constructors* to perform this task. Shape constructors are geometrical entities that transform finite point sets into a multiply connected shape in general. Due to their importance in many areas, they have attracted much attention in Computational Geometry in the last years. In particular, we employ α-shapes [7]. α-shapes define a one-parameter family of shapes (being α the parameter), ranging from the "coarsest" to the "finest" level of detail. α can be seen, precisely, as a measure of this level of detail.

An α-shape is a polytope that is not necessarily convex nor connected, being triangulated by a subset of the Delaunay triangulation of the points. Thus, the empty circumcircle criterion holds. Let $\boldsymbol{N}$ be our finite set of points in $\mathbb{R}^3$ and α a real number, with $0 \leq \alpha < \infty$. A k-simplex σ_T with $0 \leq k \leq 3$ is defined as the convex hull of a subset $T \subseteq N$ of size $\mid T \mid = k+1$. Let b be an α-ball, that is, an open ball of radius α. A k-simplex σ_T is said to be α-exposed if there exist an empty α-ball b with $T = \partial b \bigcap N$ where ∂ means the boundary of the ball. In other words, a k-simplex is said to be α-exposed if an α-ball that passes through its defining points

contains no other point of the set N.

Thus, we can define the family of sets $F_{k,\alpha}$ as the sets of α-exposed k-simplices for the given set N. This allows us to define an α-shape of the set N as the polytope whose boundary consists on the triangles in $F_{2,\alpha}$, the edges in $F_{1,\alpha}$ and the vertices or nodes in $F_{0,\alpha}$.

A three-dimensional simplicial complex is a collection, $\mathcal{C}$, of closed k-simplices ($0 \leq k \leq 3$) that satisfies:

(i) If $\sigma_T \in \mathcal{C}$ then $\sigma_{T'} \in \mathcal{C}$ for every $T' \subseteq T$.

(ii) The intersection of two simplexes in $\mathcal{C}$ is empty or is a face of both.

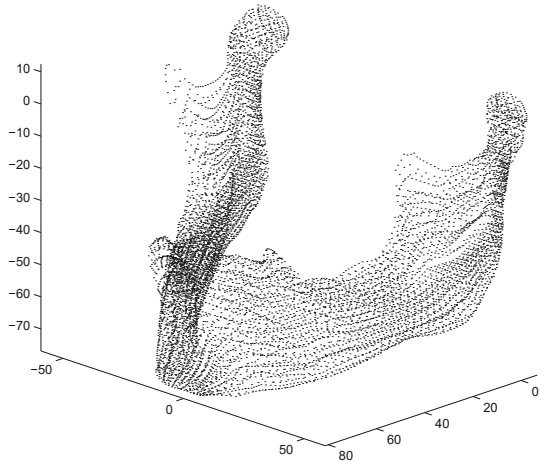
Each k-simplex σ_T included in the Delaunay triangulation, $\mathcal{D}$, defines an open ball b_T whose bounding spherical surface (in the general case) ∂b_T passes through the $k+1$ points of the simplex. Let ϱ_T be the radius of that bounding sphere, then, the family $G_{k,\alpha}$, is formed by all the k-simplexes $\sigma_T \in \mathcal{D}$ whose ball b_T is empty and $\varrho_T < \alpha$. The family $G_{k,\alpha}$ does not necessarily form simplicial complexes, so Edelsbrunner and Mücke [7] defined the α-complex, $\mathcal{C}_\alpha$, as the simplicial complex whose k-simplexes are either in $G_{k,\alpha}$, or else they bound $(k+1)$-simplexes of C_α . If we define the underlying space of $\mathcal{C}_\alpha$, $|\mathcal{C}_\alpha|$, as the union of all simplexes in $\mathcal{C}_\alpha$, the following relationship between α-shapes and α-complexes is found:

$$\mathcal{S}_\alpha = |\mathcal{C}_\alpha| \quad \forall 0 \leq \alpha < \infty \tag{6}$$

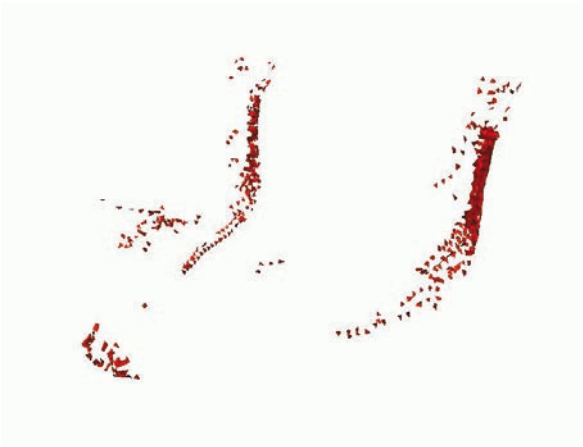
Other shape constructors giving homotopy-equivalent shapes have been recently proposed [6]. In Fig. 4 an example of the previously presented theory is presented. It represents some instances of the family of shapes for a cloud of points obtained in the three-dimensional scanning of a human mandible.

Note that the key question in using α-shapes is not to find the precise value of α. Instead, we must set the problems in terms of *what level of detail are we interested in taking into account* for a particular geometry. If this level is set to, say, α, then the nodal spacing parameter (h in the vast majority of the Finite Element literature) must be accordingly chosen so as to verify $h < \alpha$. Actually, it is clear that we are not going to be able to represent levels of detail of smaller scale than the nodal spacing.

But the use of shape constructors, and particularly, the use of α-shapes has another relevant influence in the Natural Element Method (also in the Meshless Finite Element Method [11], although it was not initially pointed out by Idelsohn and coworkers). As demonstrated in [5], the construction of natural neighbour interpolation



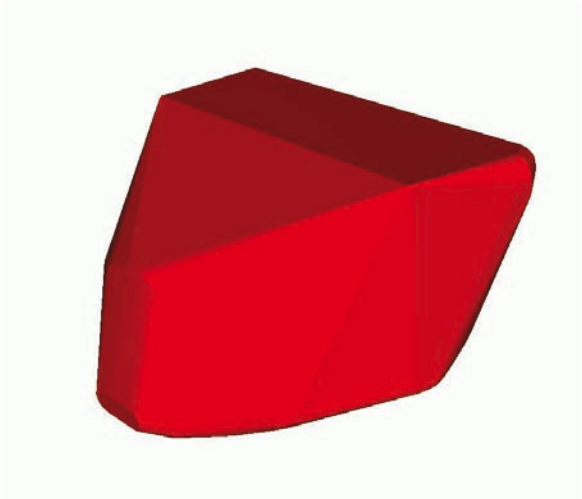
(a)



(b)

(c)

(d)



(e)

FIG 4: *Evolution of the family of α shapes of a cloud of points representing a*

(Sibson or Laplace) on an α-shape of the domain alters the distance measure. Natural neighbour interpolation is performed on the basis of Voronoi diagrams, which employ euclidean distance measure in their most general form. This leads to some lack of interpolation along non-convex boundaries. This interpolation is recovered if we construct the natural neighbour interpolants over an α-shape of the domain.

Thus, the use of α-shapes in the construction of updated Lagrangian simulations of fluid flow provides an appealing way to track the boundary of domain while ensuring appropriate interpolation of essential boundary conditions, that can be imposed directly in the discrete system of equations, as in the Finite Element Method.

4 Governing equations and discretisation

4.1 Governing equations within the fluid domain

We consider here the problem of Fluid Dynamics at moderate Reynolds number. Thus, the governing equations can be set as follows. Consider a fluid in a region Ω of the space $\mathbb{R}^2$ or $\mathbb{R}^3$. The fluid flow is governed by the following mass and momentum conservation equations:

$$\rho(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}) = \nabla\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} \qquad \text{in } \Omega\times(0,T), \tag{7}$$

$$\nabla\cdot\boldsymbol{v} = 0 \qquad \text{in } \Omega\times(0,T) \tag{8}$$

where $\boldsymbol{v}$ represents the fluid velocity, $\boldsymbol{\sigma}$ the stress tensor, ρ represents fluid density and $\boldsymbol{b}$ the volumetric forces acting on the fluid.

The constitutive equation for a newtonian fluid is given by:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} = -p\boldsymbol{I} + 2\mu\nabla^s\boldsymbol{v} + \lambda(\nabla\cdot\boldsymbol{v})\boldsymbol{I}, \tag{9}$$

where $\nabla^s(\boldsymbol{v})$ is the strain rate tensor, p the pressure, μ is the dynamic viscosity of the fluid and λ the second coefficient of viscosity. For incompressible fluids $\nabla\cdot\boldsymbol{v} = 0$ and consequently the before-mentioned Eq. (9), is reduced to the so-called Stokes law

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\nabla^s\boldsymbol{v}. \tag{10}$$

Substituting into Eqs. (7)-(8) we arrive to

$$\rho\Big(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\Big) - 2\mu\nabla\cdot\nabla^s\boldsymbol{v} + \nabla p = \rho\boldsymbol{b}. \tag{11}$$

It is usual to rewrite this last equation as:

$$\rho\Big(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\Big) - \mu\nabla^2\boldsymbol{v} - \mu\nabla(\nabla\cdot\boldsymbol{v}) + \nabla p = \rho\boldsymbol{b}. \tag{12}$$

Under the incompressibility assumption (8), this last Eq. (12) is transformed into

$$\rho\Big(\boldsymbol{v}_{,t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\Big) - \mu\nabla^2\boldsymbol{v} + \nabla p = \rho\boldsymbol{b}, \qquad \text{in } \Omega\times(0,T). \tag{13}$$

To solve the problem we must prescribe an initial state as well as boundary conditions given by

$$\boldsymbol{v}(\boldsymbol{x},t) = \boldsymbol{v}_D(\boldsymbol{x},t), \qquad \boldsymbol{x}\in\Gamma_D, \qquad t\in(0,T), \tag{14}$$

where Γ_D stands for the Dirichlet (essential) portion of the boundary and Γ_N represents the Neumann or natural portion of the boundary:

$$\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{x},t) = \boldsymbol{t}(\boldsymbol{x},t), \qquad \boldsymbol{x}\in\Gamma_N, \qquad t\in(0,T). \tag{15}$$

4.2 Time discretization

The motion equations can be grouped to

$$\nabla\cdot\boldsymbol{\sigma} + \rho\boldsymbol{b} = \rho\frac{d\boldsymbol{v}}{dt} = \rho\left(\frac{\partial\boldsymbol{v}}{\partial t} + \boldsymbol{v}\nabla\cdot\boldsymbol{v}\right), \tag{16}$$

$$\nabla\cdot\boldsymbol{v} = 0, \tag{17}$$

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\nabla^s\boldsymbol{v}. \tag{18}$$

The weak form of the problem associated to Eqs. (16), (17) and (18) is:

$$\int_\Omega 2\mu\boldsymbol{D}:\boldsymbol{D}^*\,d\Omega - \int_\Omega p\boldsymbol{I}:\boldsymbol{D}^*\,d\Omega = -\int_\Omega \rho\boldsymbol{b}\boldsymbol{v}^*\,d\Omega + \int_\Omega \rho\frac{d\boldsymbol{v}}{dt}\boldsymbol{v}^*\,d\Omega, \tag{19}$$

and

$$\int_\Omega \nabla\cdot\boldsymbol{v}\;p^*\,d\Omega = 0, \tag{20}$$

where the tensor $\boldsymbol{D} = \nabla^s\boldsymbol{v}$ represents the strain rate tensor and $\boldsymbol{b}$ the vector of volumetric forces applied to the fluid.

The second term in the right-hand side of Eq. (19) represents the inertia effects during the flow. Time discretization of this term represents the discretization of the material derivative along the nodal trajectories, which are precisely the characteristic

lines related to the advection operator. Thus, assuming known the flow kinematics at time $t^{n-1} = (n-1)\Delta t$, we proceed as follows:

$$\int_\Omega \rho \frac{d\boldsymbol{v}}{dt} \boldsymbol{v}^* \, d\Omega = \int_\Omega \rho \frac{\boldsymbol{v}^n(\boldsymbol{x}) - \boldsymbol{v}^{n-1}(\boldsymbol{X})}{\Delta t} \boldsymbol{v}^* \, d\Omega, \tag{21}$$

where $\boldsymbol{X}$ represents the position at time t^{n-1} occupied by the particle located at position $\boldsymbol{x}$ at present time t^n, i.e.:

$$\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{v}^{n-1}(\boldsymbol{X})\Delta t. \tag{22}$$

So we arrive to

$$\int_\Omega 2\mu \boldsymbol{D} : \boldsymbol{D}^* \, d\Omega - \int_\Omega p\boldsymbol{I} : \boldsymbol{D}^* \, d\Omega - \int_\Omega \rho \frac{\boldsymbol{v}\boldsymbol{v}^*}{\Delta t} \, d\Omega =$$

$$= - \int_\Omega \rho \boldsymbol{b}\boldsymbol{v}^* \, d\Omega - \int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\boldsymbol{v}^*}{\Delta t} \, d\Omega, \tag{23}$$

and

$$\int_\Omega \nabla \cdot \boldsymbol{v} \; p^* \, d\Omega = 0. \tag{24}$$

where we have dropped the superindex in all the variables corresponding to the current time step.

4.3 Algorithmical issues

The most difficult term in Eq. (23) is the second term of the right-hand side. The numerical integration of this term depends on the quadrature scheme employed.

If we employ traditional Gauss-based quadratures on the Delaunay triangles, it will be necessary to find the position at time t^{n-1} of the point occupying at time t^n the position of the integration point $\boldsymbol{\xi}_k$ (see Fig. 5):

$$\int_\Omega \rho \frac{\boldsymbol{v}^{n-1}\boldsymbol{v}^*}{\Delta t} \, d\Omega = \sum_k \rho \frac{\boldsymbol{v}^{n-1}(\boldsymbol{\Xi}_k)\boldsymbol{v}^*(\boldsymbol{\xi}_k)}{\Delta t} \omega_k, \tag{25}$$

where ω_k represent the weights associated to integration points $\boldsymbol{\xi}_k$, and $\boldsymbol{\Xi}_k$ corresponds to the position occupied at time t^{n-1} by the quadrature points $\boldsymbol{\xi}_k$.

If we employ some type of nodal integration, as in [9], this procedure becomes straightforward, with the only need to store nodal velocities at time step $t-1$. We

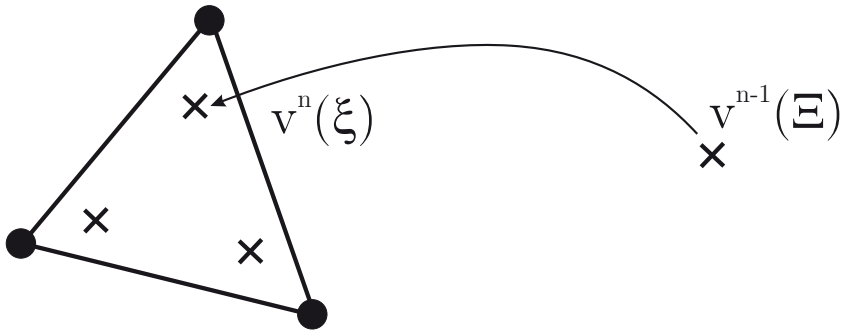


FIG. 5: *Determination of the position of quadrature points at time step* $t-1$.

discuss here the procedure to follow when employing Gauss quadratures on the Delaunay triangles. We proceed iteratively. Denoting by i the current iteration, we apply

$$\boldsymbol{x}_k = \boldsymbol{X}_k^i + \boldsymbol{v}^{n-1}(\boldsymbol{X}_k^i)\Delta t, \;\; \text{with } \boldsymbol{x}_k = \boldsymbol{X}_k^0$$

until $\boldsymbol{X}_k^i \approx \boldsymbol{X}_k^{i-1}$.

Since we are using an updated Lagrangian strategy, the computation of the term $\boldsymbol{v}^{n-1}(\boldsymbol{X}_k^{i-1})$ requires a projection from the stored nodal velocities at time $t-1$. One problem related to this projection is that the Delaunay triangulation is highly sensible to slight nodal movements. However, the resulting interpolation is not sensible to these changes (the associated Voronoi diagram is also non-sensible, see [22]) so it is a reasonable assumption to consider the neighbourhood of a given integration point as fixed (and therefore equal to that of the time step t). Many FE or meshless codes do not consider the possibility of storing a previous nodal connectivity. We have assumed that the number of natural neighbours of a given integration point does not change during a time step, thus needing the storage of nodal velocities at time $t-1$ only. It can occur that some of the nodes neigbouring the integration point at time t were not actually its neighbours at time $t-1$, but this does not constitute a problem, since the number of natural neighbours of a point is usually high (much bigger than three). The quality of the interpolation is thus guaranteed. In fact, this procedure has shown to converge at a high speed, with no more than 3 iterations, at least for reasonable time steps.

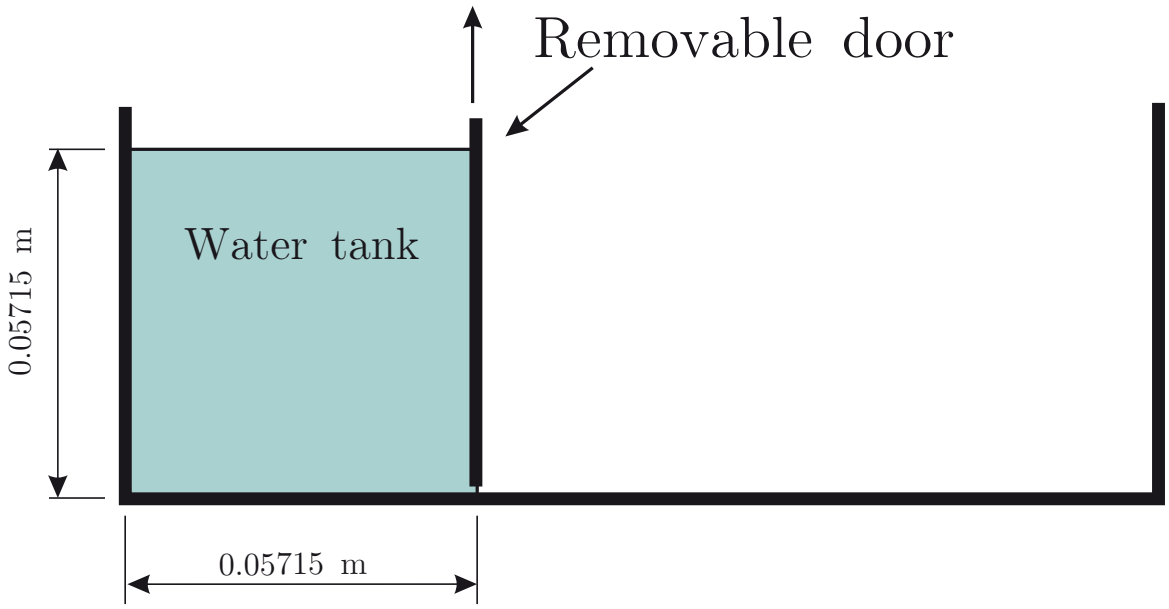


FIG. 6: *Experimental configuration of the broken dam problem.*

5 Numerical examples

5.1 Broken dam problem

The simulation of the broken dam problem is a classical example in free-surface simulations. We consider a rectangular column of water, initially retained by a door that is instantaneously removed at time $t = 0$ (see Fig. 6).

When the door is removed, water flows under the action of gravity, considered as $9.81\ m/s^2$. Density of water is $10^3\ kg/m^3$, and a viscosity of $0.1\ Pa \cdot s$ was assumed. The mathematical model was composed of 3364 nodes. No remeshing, addition or deletion of nodes was performed throughout the computation.

Fig. 7 shows a comparison between numerical results and experimental ones, obtained from the literature [14]. As can be noticed, excellent agreement is found between experimental and numerical results, despite the distortion of the triangulation, shown in Fig. 8. A detail of the triangulation is shown in Fig. 9.

In Figs. 10 and 11 the error in mass conservation is depicted, which remains always below 3%. The influence of the relationship between the parameter α and the nodal parameter h on this error is deeply analyzed in [15]. In Fig. 12 the evolution of the vertical component of the velocity is depicted.

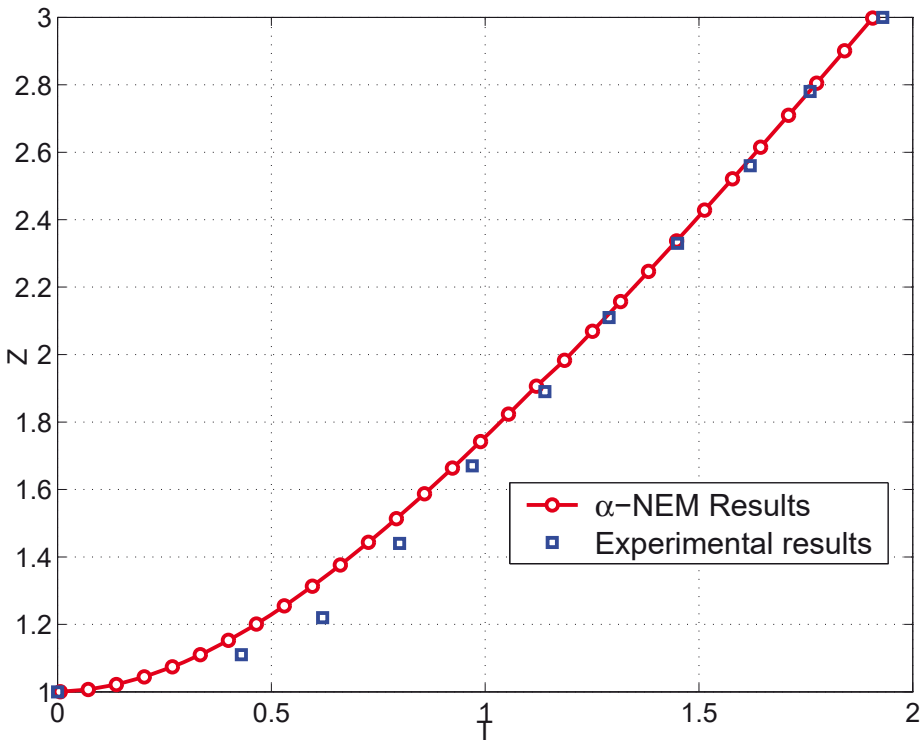


FIG. 7: *Front position for the broken dam problem.*

5.2 Rotating water mill

This example can be found in [12]. It represents a water mill, which is considered rigid, moving with a prescribed velocity. The geometry of the domain is represented in Fig. 13.

The model consisted of 2367 nodes, that remained the same throughout the simulation. Viscosity was set to 0.001 Pa·s. The mill moves with angular velocity of 0.5 rad·s^{-1}. Some snapshots of the velocity field during the simulation are shown in Fig. 14.

The simulation covered 90 degrees of rotation in the mill. The generated wave is clearly seen.

6 Conclusion

In this paper we have presented our initial developments towards a lagrangian scheme for FSI problems by employing the Natural Element Method. The Navier-Stokes equations are integrated in a lagrangian setting by employing the method of characteristics. By employing the NEM a standard coupling (in the FE sense) between

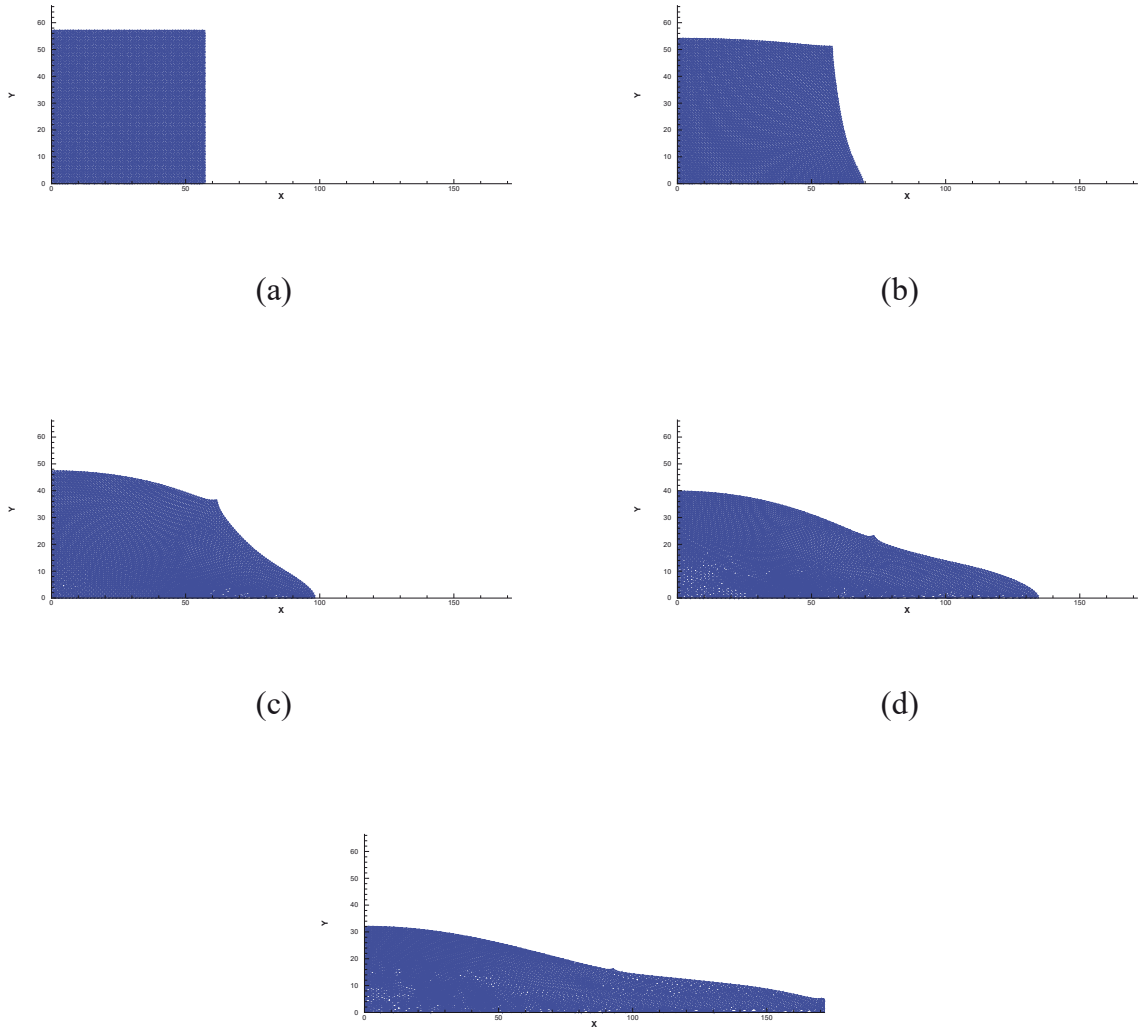


FIG. 8: *Evolution of the α-shapes in time for the broken dam problem.*

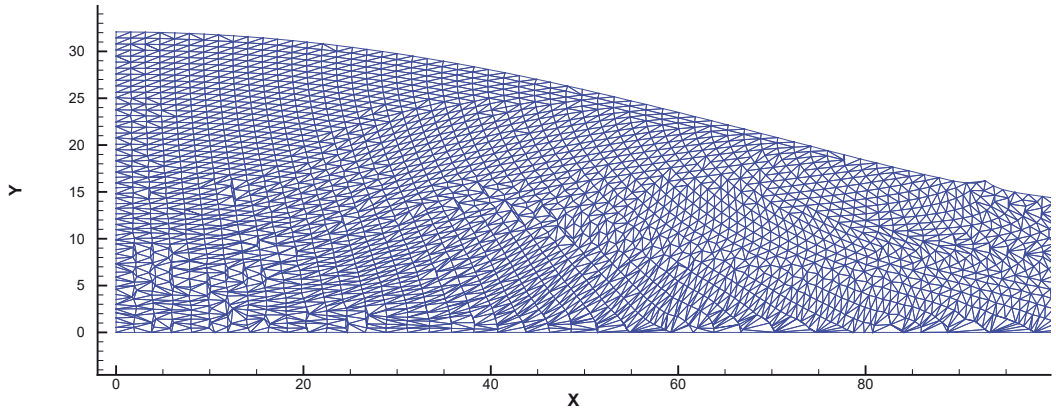


FIG. 9: *Detail of the deformed triangulation for the* 100^{th} *time step. Note the highly distorted triangles on the lower region of the geometry.*

the fluid and solid domains can be done and thus the potential capabilities of the proposed method.

This method can be now integrated in a block-iterative scheme in which the interaction with the solid domain can be solved by standard techniques, such as Newton-Raphson or Picard methods.

Acknowledgements

The work here presented has been funded by the Spanish Ministry of Education and Science, through grant number DPI2005-08727-C02-01. This funding is gratefully acknowledged.

References

[1] BABUŠKA, I., AND MELENK, J. M. The partition of Unity Method. *International Journal for Numerical Methods in Engineering* **40** (1997), 727–758.

[2] BELIKOV, V. V., IVANOV, V. D., KONTOROVICH, V. K., KORYTNIK, S. A., AND SEMENOV, A. Y. The Non-Sibsonian Interpolation: A New Method of Interpolation of the Values of a Function on an Arbitrary Set of Points. *Computational Mathematics and Mathematical Physics* **37**, 1 (1997), 9–15.

[3] CUETO, E., CALVO, B., AND DOBLARÉ, M. Modeling three-dimensional piece-wise homogeneous domains using the α-shape based Natural Element Method. *International Journal for Numerical Methods in Engineering* **54** (2002), 871–897.

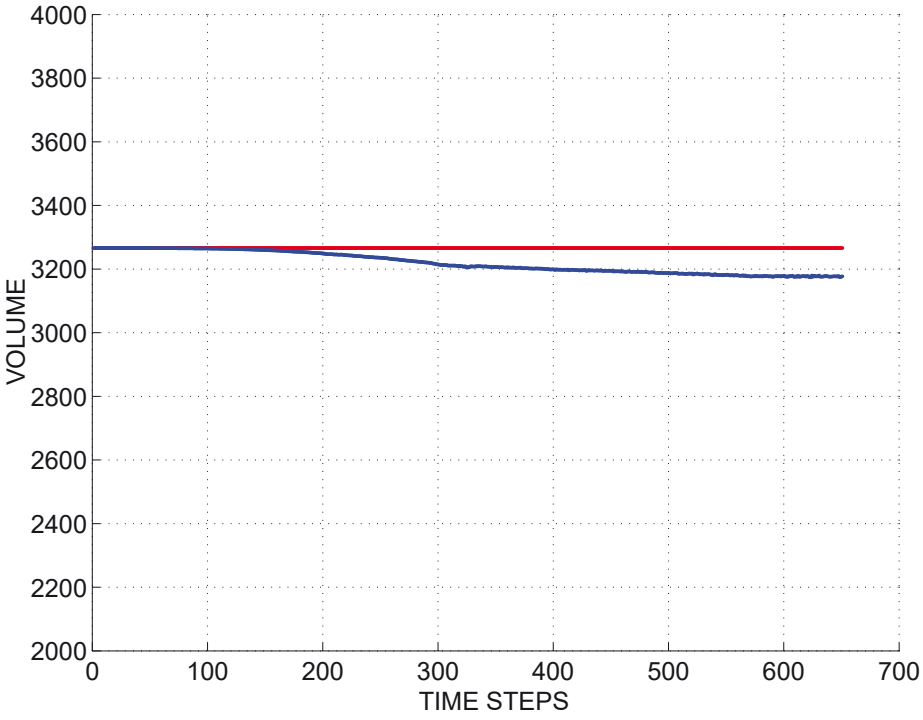


FIG. 10: *Evolution of the fluid volume for the broken dam problem. In red, the theoretical one and, in blue, the computed volume.*

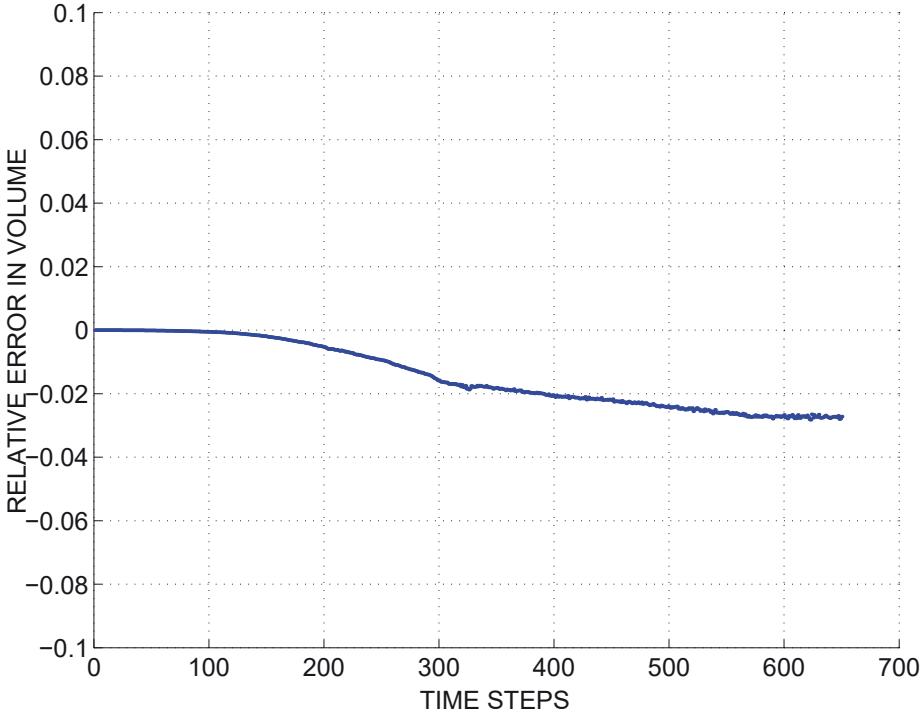


FIG. 11: *Evolution of the relative error fluid volume error in volume for the broken dam problem.*

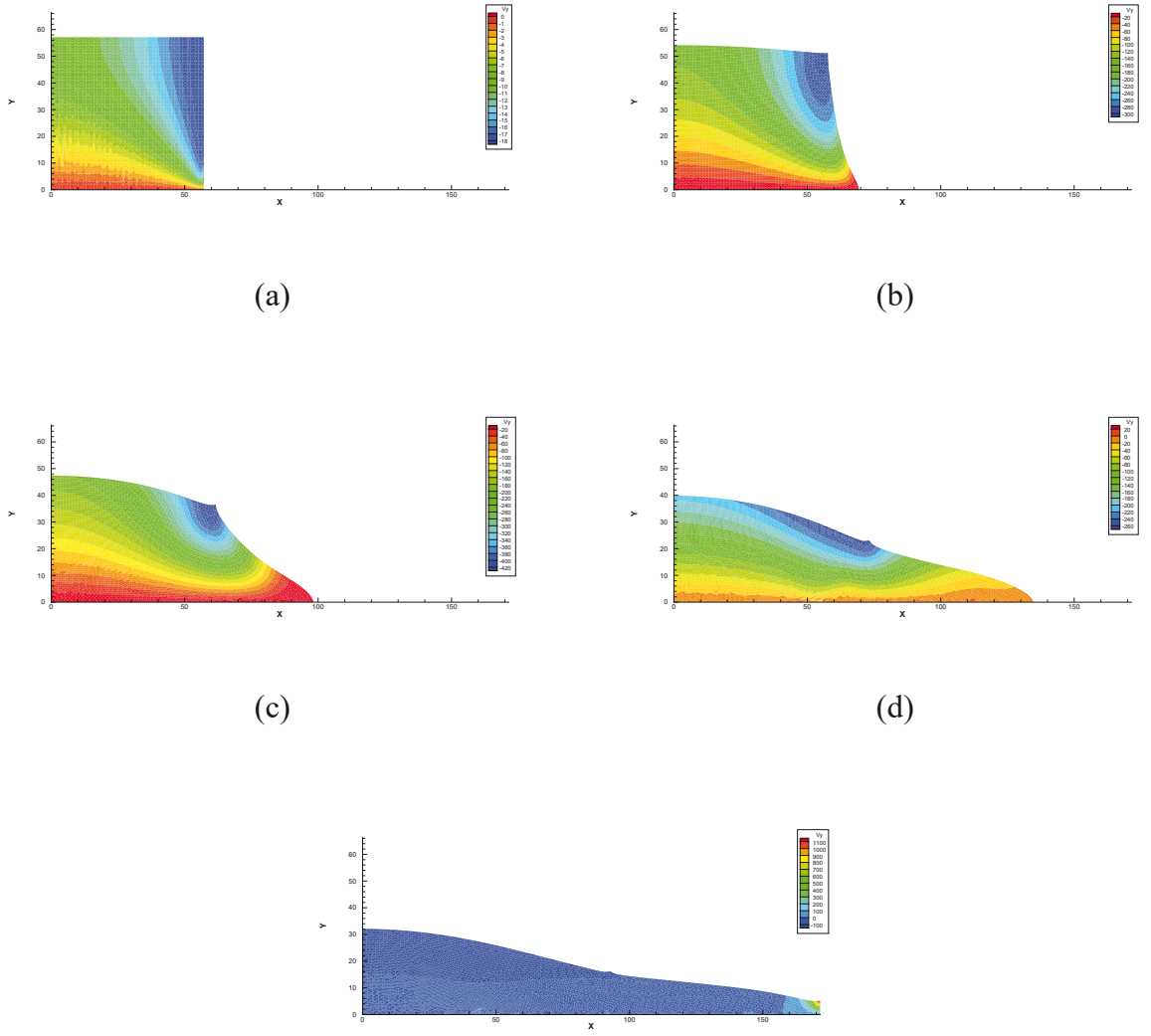


FIG. 12: *Evolution of the vertical velocity for the broken dam problem.*

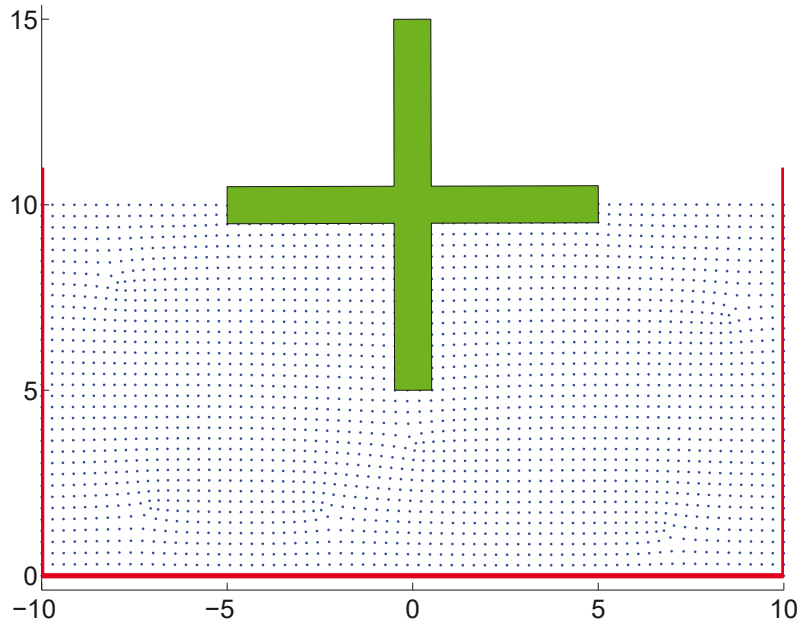


FIG. 13: *Initial geometry of the rotating water mill problem.*

[4] CUETO, E., CEGOÑINO, J., CALVO, B., AND DOBLARÉ, M. On the imposition of essential boundary conditions in Natural Neighbour Galerkin methods. *Communications in Numerical Methods in Engineering* **19(5)** (2003), 361–376.

[5] CUETO, E., DOBLARÉ, M., AND GRACIA, L. Imposing essential boundary conditions in the Natural Element Method by means of density-scaled α-shapes. *International Journal for Numerical Methods in Engineering* **49-4** (2000), 519–546.

[6] DEY, T. K., GIESEN, J., AND JOHN, M. Alpha-shapes and flow shapes are homotopy equivalent. In *Proc. Sympos. Theory Computing (STOC 2003)* (2003), pp. 493–502.

[7] EDELSBRUNNER, H., AND MÜCKE, E. Three Dimensional Alpha Shapes. *ACM Transactions on Graphics* **13** (1994), 43–72.

[8] FERNANDEZ, M. A., AND LETALLEC, P. Linear stability analysis in fluid-structure interaction with transpiration. Part I: Formulation and mathematical analysis. *Computer Methods in Applied Mechanics and Engineering* **192 (43)** (2003), 4805–4835.

[9] GONZALEZ, D., CUETO, E., MARTINEZ, M. A., AND DOBLARE, M. Numerical integration in Natural Neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* **60(12)** (2004), 2077–2104.

[10] HIYOSHI, H., AND SUGIHARA, K. Two Generalizations of an Interpolant Based on Voronoi Diagrams. *International Journal of Shape Modeling* **5**, 2 (1999), 219–231.

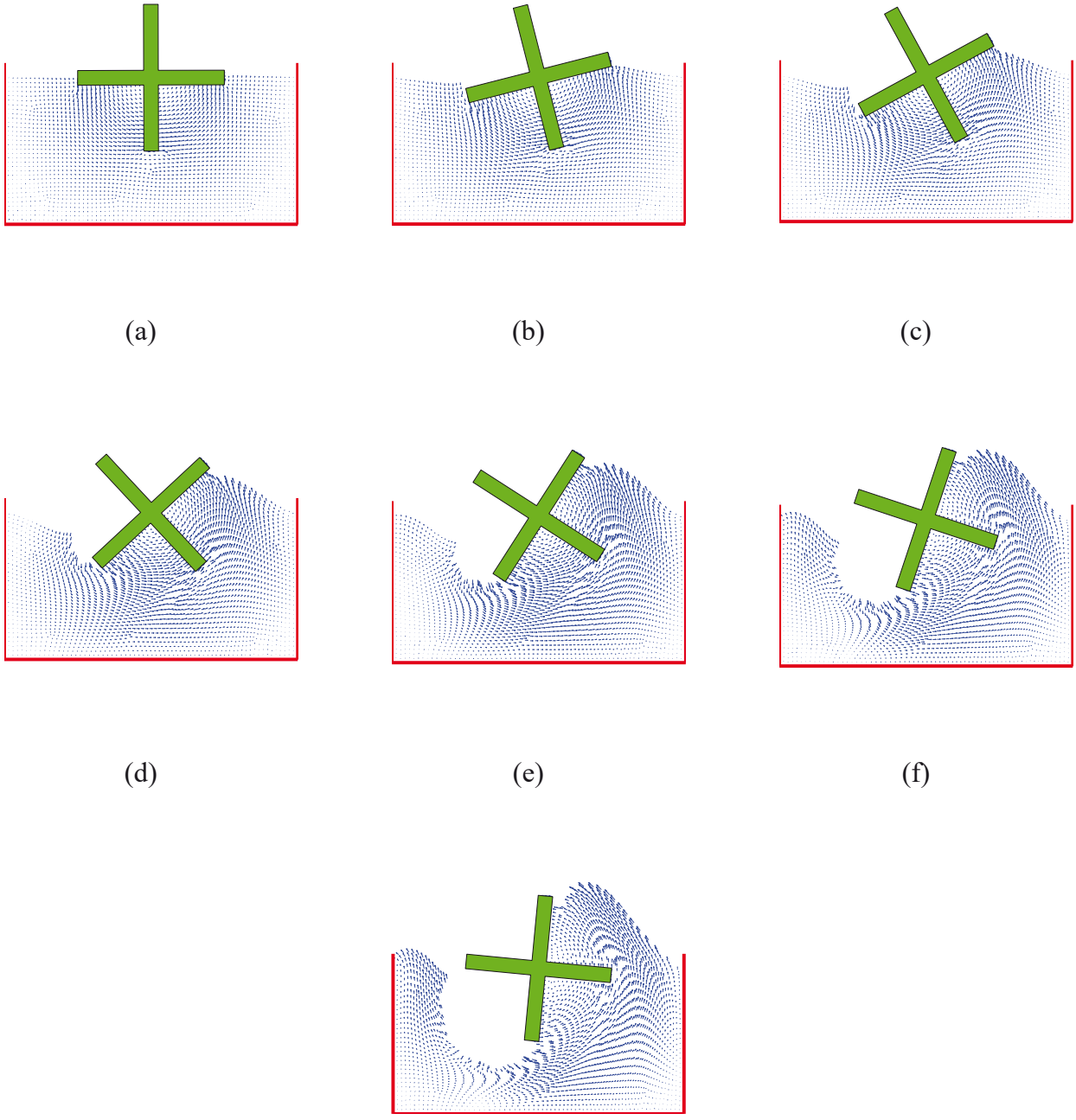


FIG. 14: *Some snapshots of the velocity field in the rotating mill problem.*

[11] IDELSOHN, S. R., OÑATE, E., CALVO, N., AND DEL PIN, F. The meshless finite element method. *International Journal for Numerical Methods in Engineering* **58** (2003), 893–912.

[12] IDELSOHN, S. R., ONATE, E., AND PIN, F. D. A Lagrangian meshless finite element method applied to fluid-structure interaction problems. *Computers and Structures* **81 (8-11)** (2003), 655–671.

[13] LEWIS, R. W., NAVTI, S. E., AND TAYLOR, C. A mixed Lagrangian-Eulerian approach to modelling fluid flow during mould filling. *International Journal for Numerical Methods in Engineering* **25** (1997), 931–952.

[14] MARTIN, J., AND MOYCE, W. Part IV. an experimental study of the collapse of liquid columns on a rigid horizontal plane. *Phil. Tran. R. Soc. London* **244** (1952), 312.

[15] MARTINEZ, M. A., CUETO, E., ALFARO, I., DOBLARE, M., AND CHINESTA, F. Updated Lagrangian free surface flow simulations with Natural Neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* **60(13)** (2004), 2105–2129.

[16] RUGONYI, S., AND BATHE, K. J. On finite element analysis of fluid flows fully coupled with structural interactions. *Computer Modeling in Engineering and Sciences* **2 (2)** (2001), 195–212.

[17] SIBSON, R. A Vector Identity for the Dirichlet Tesselation. *Mathematical Proceedings of the Cambridge Philosophical Society* **87** (1980), 151–155.

[18] SIBSON, R. A Brief Description of Natural Neighbour Interpolation. In *Interpreting Multivariate Data. V. Barnett (Editor)*. John Wiley, 1981, pp. 21–36.

[19] SUKUMAR, N., MORAN, B., AND BELYTSCHKO, T. The Natural Element Method in Solid Mechanics. *International Journal for Numerical Methods in Engineering* **43(5)** (1998), 839–887.

[20] SUKUMAR, N., MORAN, B., SEMENOV, A. Y., AND BELIKOV, V. V. Natural Neighbor Galerkin Methods. *International Journal for Numerical Methods in Engineering* **50(1)** (2001), 1–27.

[21] THIESSEN, A. H. Precipitation averages for large areas. *Monthly Weather Report* **39** (1911), 1082–1084.

[22] WATSON, D. Computing the n-dimensional Delaunay Tessellation with Application to Voronoi Polytopes. *The Computer Journal* **24(2)** (1981), 162–172.

[23] WATSON, D. *Nngridr. An Implementation of Natural Neighbor Interpolation*. Published by the author, 1994.

[24] YVONNET, J., RYCKELYNCK, D., LORONG, P., AND CHINESTA, F. A new extension of the Natural Element method for non-convex and discontnuous problems: the Constrained Natural Element method. *International Journal for Numerical Methods in Enginering* **60(8)** (2004), 1452–1474.