

A natural element updated Lagrangian strategy for free-surface fluid dynamics [☆]

D. González ^a, E. Cueto ^{a,*}, F. Chinesta ^b, M. Doblaré ^a

^a Aragón Institute of Engineering Research., University of Zaragoza, Edificio Betancourt,
Campus Rio Ebro. María de Luna, s.n. E-50018 Zaragoza, Spain

^b Laboratoire de Mécanique des Systèmes et des Procédés, LMSP UMR 8106, CNRS-ENSAM-ESEM,
105, Bvd. de l'Hôpital, F-75013 Paris, France

Received 24 January 2006; received in revised form 28 July 2006; accepted 5 September 2006
Available online 18 October 2006

Abstract

We present a novel algorithm to simulate free-surface fluid dynamics phenomena at low Reynolds numbers in an updated Lagrangian framework. It is based on the use of one of the most recent meshless methods, the so-called natural element method. Free-surface tracking is performed by employing a particular instance of “shape constructors” called α -shapes. This means that at each time step the geometry of the domain is extracted by employing a particular member of the finite set of shapes described by the nodal cloud. The Lagrangian framework allows us to integrate the inertial terms of the Navier–Stokes equations by employing the method of characteristics which are, precisely, the nodal pathlines. A theoretical description of the method is included together with some examples showing its performance.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Meshless; Natural element method; α -Shapes; Free-surface flows; Fluid Dynamics

1. Introduction

The Eulerian approach has been the most extended framework to represent the equations governing the dynamics of a Newtonian fluid. In this approach, the computational mesh is fixed and the fluid moves with respect to the grid. This formulation has the obvious advantage of an easy treatment of large distortions in the fluid motion and is indispensable in the treatment of turbulent flows, for instance [20]. However, the presence of free boundaries extremely complicates the formulation of flow problems in Eulerian frameworks. This is so since the free boundary must be located somewhere within an element. Volume of fluid (VoF) techniques rely on the employ of an implicit function called the *presence of fluid* function, that takes unity value in the region filled with fluid and vanishing in the empty domain. This function is advected with the velocity of

[☆] Research supported by Spanish Ministry of Education and Science through Project CICYT-DPI2005-08727-C02-01.

* Corresponding author. Tel.: +34 976761000; fax: +34 976762578.

E-mail address: ecueto@unizar.es (E. Cueto).

the fluid throughout the computation. Other techniques, known as *tracking* methods rely on the use of markers, whose position is updated with the just computed fluid velocity field [43].

Arbitrary Lagrangian–Eulerian formulations [19], in which an artificial velocity of the computational mesh is introduced, somewhat alleviates this problem. However, finding the most adequate velocity of the mesh for a particular problem is far from being straightforward. Also, the positioning of the nodes on the free surface deserves some analysis [9].

In the last decade, the irruption of meshless or meshfree methods (see [35] or [7] just to cite some of the first works on the topic) in the field of computational mechanics renewed the interest on (updated) Lagrangian approaches to fluid flows problems involving free-surfaces. Recently, a number of works have been published in this area. See, for instance, [32] for an application of the Meshless–Local Petrov–Galerkin method [1] to non-linear water-wave problems or [28] for an application of the particle finite element method to free-surface problems.

In essence, meshless methods are less dependent on the regularity of the mesh than finite elements [3]. In general, they are based on the employ of scattered data interpolation techniques in a Galerkin framework, although collocation approaches exist. Moving least squares interpolation, for instance, is the basis of some of the most popular meshless methods [35]. Other meshless methods have demonstrated to possess an equivalent structure, although not initially based on MLS approximations [31]. This lower dependency on mesh distortion allows us to employ updated Lagrangian strategies for fluid mechanics problems, in which the nodes move with their respective material velocity, regardless of the regularity of their resulting position at each time step. Nodal connectivity is then found by a search algorithm transparent to the user.

However, most meshless methods present some problems that, despite the vast effort of research made during the last years, is still an open issue. For instance, those methods based on MLS approximants (or, in general, those based on circular, square or elliptically supported shape functions) lack of appropriate interpolation along essential boundaries.

The vast majority of meshless methods—at least those based on Galerkin strategies—also present some problems in the numerical integration of the weak form of the problem. Since they employ non-polynomial shape functions, numerical integration of the resulting stiffness or mass matrices presents some deficiencies if traditional Gauss quadratures are used. Other important factor in the numerical integration error is the non-conformity of integration cells and shape functions' supports, as reported in [18].

Among the newest meshless methods, the natural element method (NEM) presents some interesting features [40,16]. For instance, it has been demonstrated that exact interpolation of essential (Dirichlet) boundary conditions is possible under very weak conditions [15,14]. Also, it has been recently demonstrated [23] that the NEM possesses a particularly well-suited structure for the application of stabilised conforming nodal integration schemes [12], thus leading to a very accurate nodal method with great accuracy in numerical integration.

In this paper, we present a scheme based on the use of the method of characteristics [36] for the integration of the inertia terms of the Navier–Stokes equations. A salient feature of the method is the use of α -shapes in order to track the free surface of the domain as it evolves. After a theoretical description of the NEM in Sections 2 and 3, the proposed algorithm is described in Section 4. Some numerical examples demonstrating the capabilities of the proposed method are shown in Section 5. Finally, we conclude with some discussions and some concluding remarks.

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 [39,26] to construct trial and test functions. Prior to the introduction of these interpolation techniques, it is necessary to define some basic concepts.

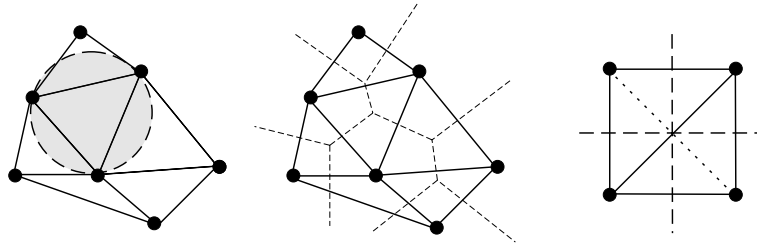


Fig. 1. Delaunay triangulation and Voronoi diagram of a cloud of points.

The model will be constructed upon a cloud of points with no connectivity on it. We will call this cloud of points $N = \{n_1, n_2, \dots, n_M\} \subset \mathbb{R}^d$, 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 Fig. 1):

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

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

The dual structure of the Voronoi diagram is the Delaunay triangulation¹, obtained by connecting nodes that share a common $(d - 1)$ -dimensional facet. While 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, as depicted in Fig. 1 (right)). 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

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

The simplest of the natural neighbour-based interpolants is the so-called Thiessen’s interpolant [42]. Its interpolating functions are defined as

$$\psi_I(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in T_I \\ 0 & \text{elsewhere.} \end{cases} \tag{3}$$

The Thiessen interpolant is a piece-wise constant function, defined over each Voronoi cell. It defines a method of interpolation often referred to as *nearest neighbour* interpolation, since a point is given a value defined by its nearest neighbour. Although it is obviously not valid for the solution of second-order partial differential equations, it can be used to interpolate the pressure in formulations arising from Hellinger–Reissner-like mixed variational principles.

The most extended natural neighbour interpolation method, however, is the Sibson interpolant [38,39]. Consider the introduction of the point \mathbf{x} in the cloud of nodes. Due to this introduction, the Voronoi diagram will be altered, affecting the Voronoi cells of the natural neighbours of \mathbf{x} . Sibson [38] defined the natural neighbour coordinates of a point \mathbf{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 \mathbf{x} to the initial cloud of points to the total volume of T_x . In other words, if $\kappa(\mathbf{x})$ and $\kappa_I(\mathbf{x})$ are the Lebesgue measures of T_x and T_{xI} respectively, the natural neighbour coordinates of \mathbf{x} with respect to the node I is defined as

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.

$$\phi_I(\mathbf{x}) = \frac{\kappa_I(\mathbf{x})}{\kappa(\mathbf{x})}. \tag{4}$$

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

$$\phi_1(\mathbf{x}) = \frac{A_{abfe}}{A_{abcd}}. \tag{5}$$

It is straightforward to prove that NE shape functions (see Fig. 3) form a partition of unity [4], as well as some other properties like positivity (i.e., $0 \leq \phi_I(\mathbf{x}) \leq 1 \forall I, \forall \mathbf{x}$) and strict interpolation:

$$\phi_I(\mathbf{x}_J) = \delta_{IJ}. \tag{6}$$

A third type of natural neighbour interpolation was independently established by Belikov [6] and Hiyoshi [26]. It is referred to as *non-Sibsonian* or *Laplace* interpolation. It has not been used in this work.

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 [40,26] 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 possess linear completeness (i.e., exact reproduction of a linear field).

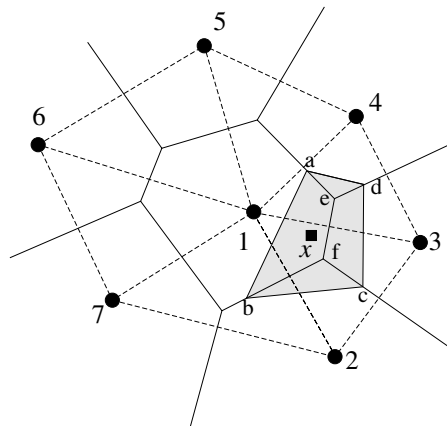


Fig. 2. Definition of the natural neighbour coordinates of a point x .

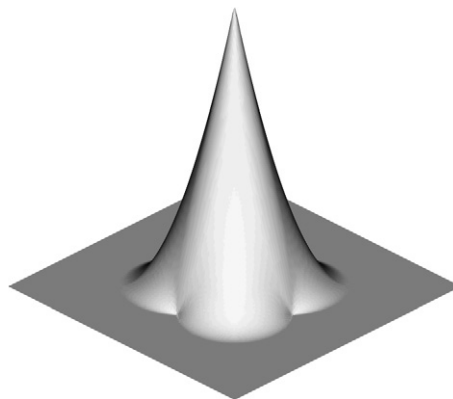


Fig. 3. Typical function $\phi(x)$. Courtesy N. Sukumar.

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 [15,14,47] 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 [15,13] it was demonstrated that the construction of the Sibson interpolant over an α -shape [22] 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 [14].

As mentioned before, Laplace interpolants were initially supposed to reproduce linear essential boundary conditions exactly [41], although it was later demonstrated that some criteria must be met in order to ensure it [14]. The α -shape approach mentioned before was later adopted in [27] in the so-called *meshless Finite Element 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.

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 [30], 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 [22]. Other shape constructors giving homotopy-equivalent shapes have been recently proposed [17]. α -Shapes define a one-parameter family of shapes \mathcal{S}_α (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.

Details about the formal definition of the family of α -shapes can be found in [22]. In brief, the use of α -shapes to define the boundary of the domain relies in the choice of the level of detail needed to represent the domain, which is always an analyst’s decision. It is obvious then that the minimum nodal spacing parameter, say h , should be chosen so as to reproduce at least that level of detail α .

α -Shapes provide a means so as to eliminate from the triangulation those triangles or tetrahedra whose size is bigger than the before-mentioned level of detail. This criterion is very simple: just eliminate those triangles (tetrahedra) whose circum-radius is bigger than the level of detail, α .

In Fig. 4 an example of the previously presented theory is presented. It represents some instances of the finite set of shapes for a cloud in a intermediate step of the simulation of a wave breaking in a beach.

Note that the key question in using α -shapes is not to find the precise value of α for a given configuration of the nodal cloud. Instead, we must set the problems in terms of *what level of detail are we interested in taking into account* for a particular geometry.

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 [27], although it was not initially pointed out by Idelsohn and co-workers). As demonstrated in [15], the construction of natural neighbour interpolation (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.

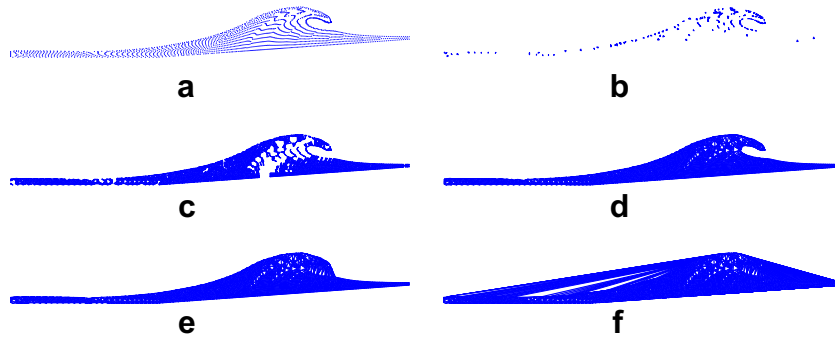


Fig. 4. Evolution of the family of α -shapes of a cloud of points representing a wave breaking on a beach. Shapes \mathcal{S}_0 (a), $\mathcal{S}_{0.5}$ (b), $\mathcal{S}_{1.0}$ (c), $\mathcal{S}_{2.0}$ (d), $\mathcal{S}_{3.0}$ (e) and \mathcal{S}_∞ (f) are depicted.

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

We consider here the problem of fluid dynamics at moderate Reynolds number. By moderate we mean that we are far from the limit of infinite Re (in the examples presented here, Re is among 10 and 2000). 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:

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

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega \times (0, T), \quad (8)$$

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

The constitutive equation for a Newtonian fluid is given by:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} = -p\mathbf{I} + 2\mu\mathbf{D} + \lambda(\nabla \cdot \mathbf{v})\mathbf{I}, \quad (9)$$

where \mathbf{D} 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 $\nabla \cdot \mathbf{v} = 0$ and consequently the before-mentioned Eq. (9), is reduced to the so-called Stokes law

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D}. \quad (10)$$

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

$$\rho(\mathbf{v}_{,t} + (\mathbf{v} \cdot \nabla)\mathbf{v}) - 2\mu\nabla \cdot \mathbf{D} + \nabla p = \rho \mathbf{b}. \quad (11)$$

It is usual to rewrite this last equation as

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

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

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

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

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{v}_D(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_D, \quad t \in (0, T), \quad (14)$$

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

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

4.2. Time discretization

The motion equations can be grouped to

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

$$\nabla \cdot \mathbf{v} = 0, \quad (17)$$

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D}. \quad (18)$$

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

$$\int_{\Omega} 2\mu\mathbf{D} : \mathbf{D}^* d\Omega - \int_{\Omega} p\mathbf{I} : \mathbf{D}^* d\Omega = - \int_{\Omega} \rho \mathbf{b} \cdot \mathbf{v}^* d\Omega + \int_{\Omega} \rho \frac{d\mathbf{v}}{dt} \cdot \mathbf{v}^* d\Omega, \quad (19)$$

and

$$\int_{\Omega} \nabla \cdot \mathbf{v} p^* d\Omega = 0, \quad (20)$$

where “ \cdot ” denotes the tensor product twice contracted and \mathbf{b} the vector of volumetric forces applied to the fluid. \mathbf{D}^* represents an admissible variation of the strain rate tensor, whereas \mathbf{v}^* represents equivalently an admissible variation of the velocity.

The second term in the right-hand side of Eq. (19) represents the inertia effects. 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:

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

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

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

So we arrive to

$$\int_{\Omega} 2\mu\mathbf{D} : \mathbf{D}^* d\Omega - \int_{\Omega} p\mathbf{I} : \mathbf{D}^* d\Omega - \int_{\Omega} \frac{\mathbf{v} \cdot \mathbf{v}^*}{\Delta t} d\Omega = - \int_{\Omega} \rho \mathbf{b} \cdot \mathbf{v}^* d\Omega - \int_{\Omega} \rho \frac{\mathbf{v}^{n-1} \cdot \mathbf{v}^*}{\Delta t} d\Omega, \quad (23)$$

and

$$\int_{\Omega} \nabla \cdot \mathbf{v} p^* d\Omega = 0, \quad (24)$$

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 ξ_k (see Fig. 5):

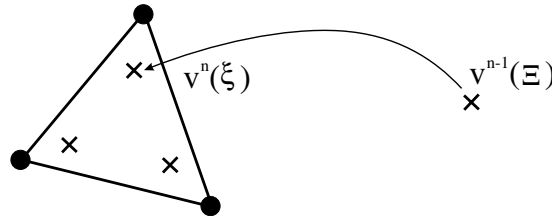


Fig. 5. Determination of the position of quadrature points at time step t^{n-1} .

$$\int_{\Omega} \rho \frac{\mathbf{v}^{n-1} \cdot \mathbf{v}^*}{\Delta t} d\Omega = \sum_k \rho \frac{\mathbf{v}^{n-1}(\Xi_k) \cdot \mathbf{v}^*(\xi_k)}{\Delta t} \omega_k, \tag{25}$$

where ω_k represent the weights associated to integration points ξ_k , and Ξ_k corresponds to the position occupied at time t^{n-1} by the quadrature point ξ_k , see Fig. 5.

If we employ some type of nodal integration, as in [23], this procedure becomes straightforward, with the only need to store nodal velocities at time step t^{n-1} . We 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

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

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

Since we are using an updated Lagrangian strategy, the computation of the term $\mathbf{v}^{n-1}(\mathbf{X}_k^{i-1})$ requires a projection from the stored nodal velocities at time t^{n-1} . One problem related to this projection is that the Delaunay triangulation is highly sensible to small nodal movements. However, the resulting interpolation is not sensible to these changes (the associated Voronoi diagram is also non-sensible, see [44]) 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^n). 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^{n-1} only. It can occur that some of the nodes neighbouring the integration point at time t^n were not actually its neighbours at time t^{n-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.

Remark 1. The opposite assumption, i.e., to recompute the neighbourhood of a given evaluation point from the stored velocities at time t^{n-1} is a more time-demanding scheme and has proven to render very similar results. No appreciable gain in accuracy has been found.

Remark 2. We have found that the use of direct nodal integration in the computation of Eq. (25)—i.e., by assigning to each node the area of its related Voronoi cell as integration weight—leads to instabilities. Higher-order nodal integration schemes should be used.

Remark 3. In [28] a Lagrangian method that employs natural neighbour interpolation to construct the discrete form of the problem was presented. In that case, however, an implicit three-step fractional method was employed to perform the time integration. This approach needs for a stabilization if small time increments are chosen. See [28] for more details. In addition, in that approach, the nodes are real material points, with velocity and mass, as required in the fractional algorithm used for solving the motion and mass conservation equations. In the method proposed in the present paper the nodes are used to define the functional interpolation of trial and test functions in a characteristics-Galerkin weighted residual formulation that can be applied to any partial differential equation.

Remark 4. The resulting system of equations is solved by employing an in-house conjugate gradient method. No special difficulties have been found in the resolution of the equations.

4.4. Mixed approximation

It is well known that not all of the displacement–pressure approximations for the weak form of the problem, Eqs. (23) and (24), lead to stable and convergent results [5]. The conditions to be fulfilled by the chosen approximation are determined by the *inf-sup* or Ladyzhenskaya–Babuška–Brezzi (LBB) condition [2,10],

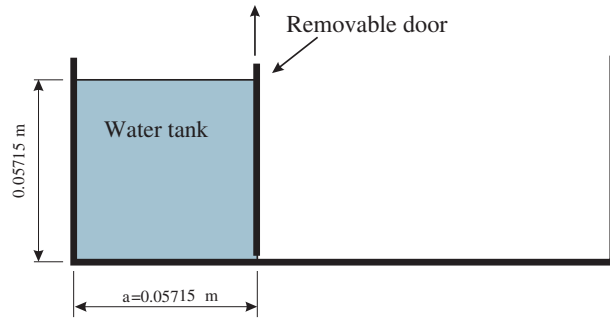


Fig. 6. Experimental configuration of the broken dam problem.

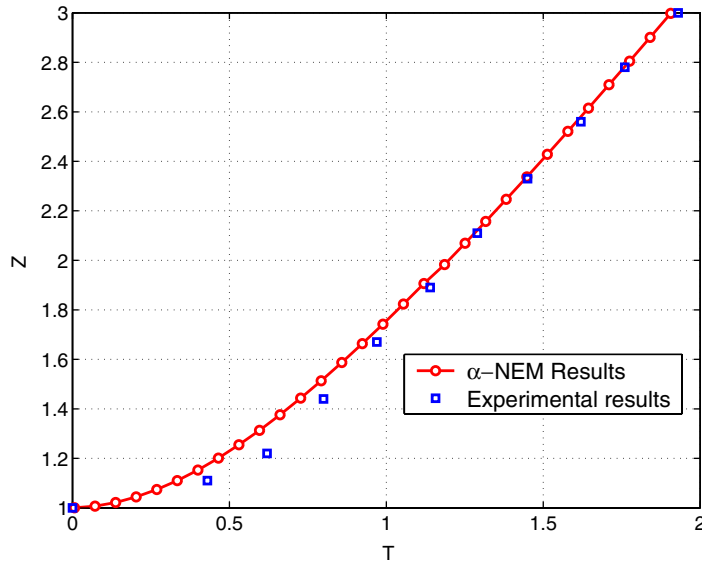


Fig. 7. Front position for the broken dam problem in non-dimensional form. $Z = z/a$, $T = t\sqrt{g/a}$.

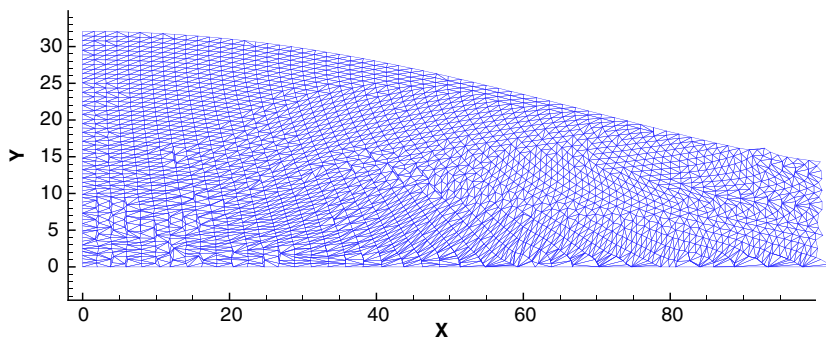


Fig. 8. Detail of the deformed triangulation for the 100th time step. Note the highly distorted triangles on the lower region of the geometry.

together with the ellipticity condition of the resultant formulation. This last condition is trivially met if no reduced integration is used in this class of problems. The LBB condition may be written as

$$\inf_{p^h \in \mathcal{P}_h} \sup_{\mathbf{u}^h \in \mathcal{U}_h} \frac{\int_{\Omega} p^h \nabla \cdot \mathbf{u}^h \, d\Omega}{\|p^h\|_0 \|\mathbf{u}^h\|_1} = \gamma_h \geq \gamma > 0, \tag{26}$$

where γ is a positive constant independent of the mesh size, h . \mathcal{U}_h and \mathcal{P}_h represent, respectively, the velocity and pressure approximation spaces whose norms are defined as

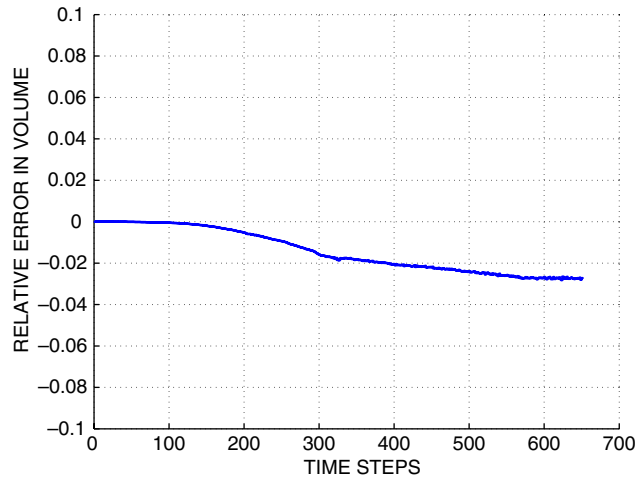


Fig. 9. Evolution of the fluid volume relative error for the broken dam problem.

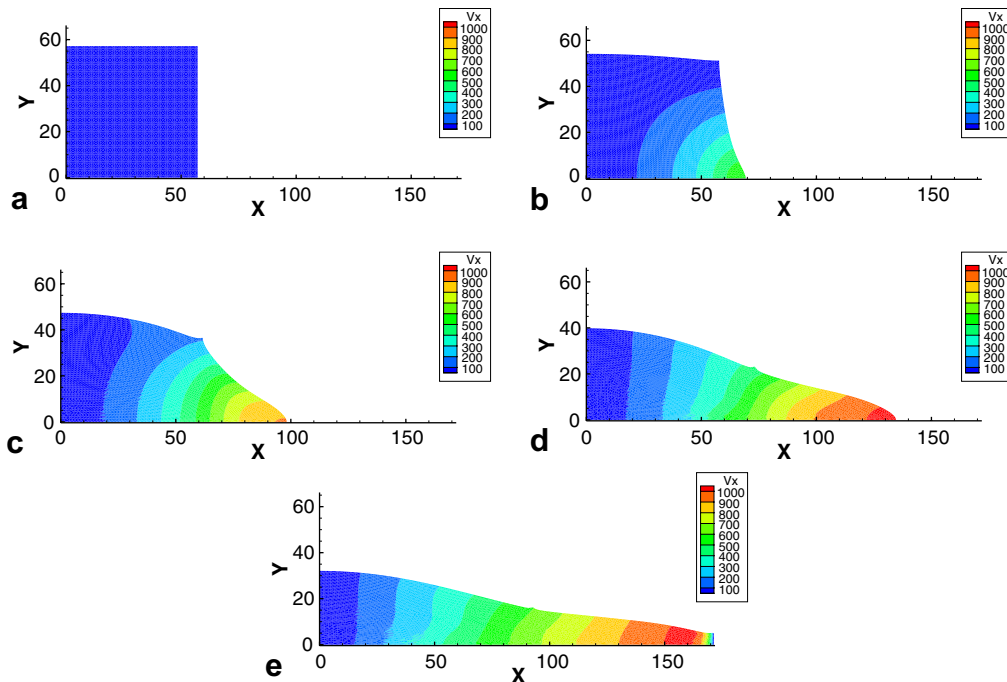
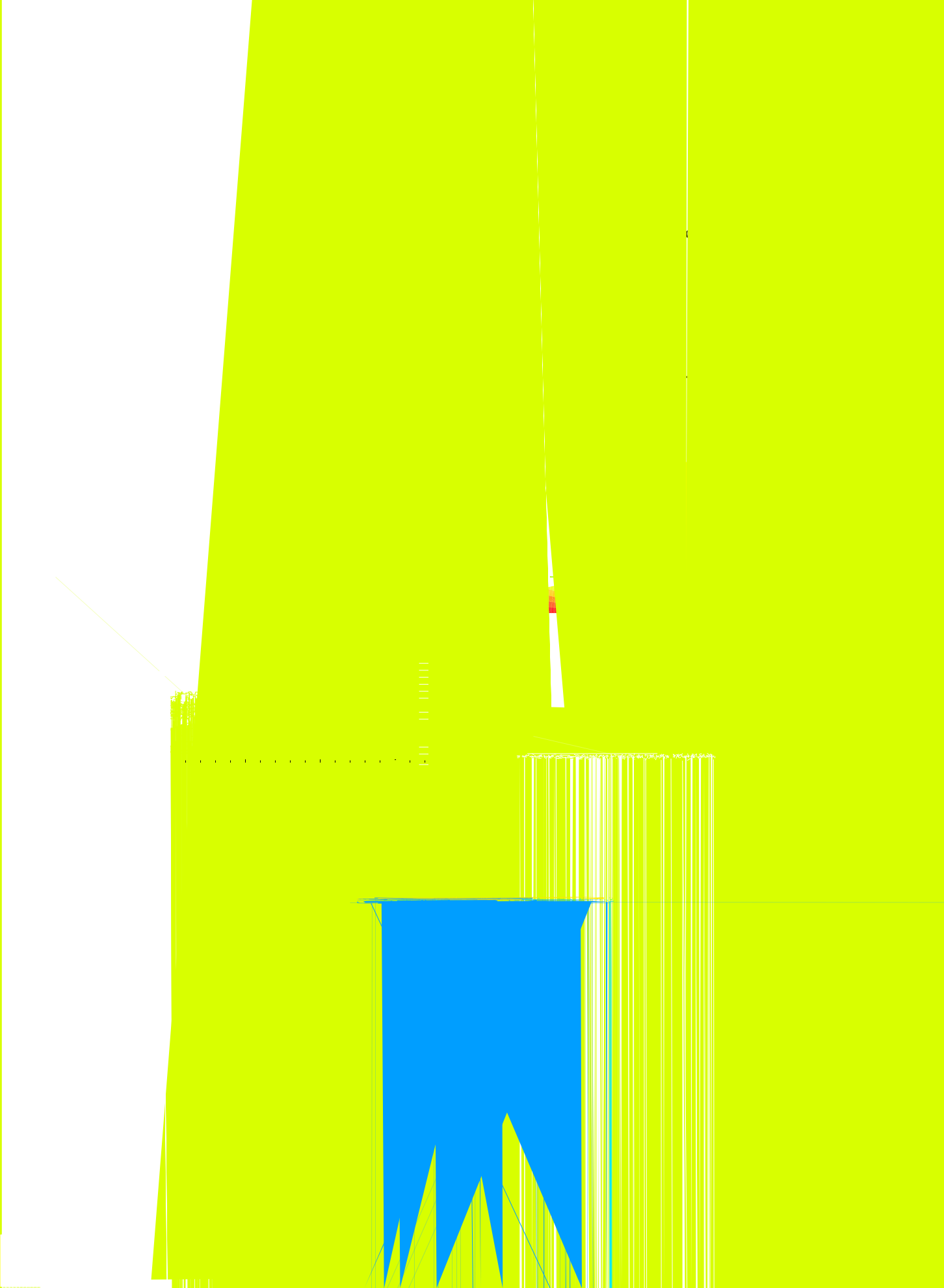
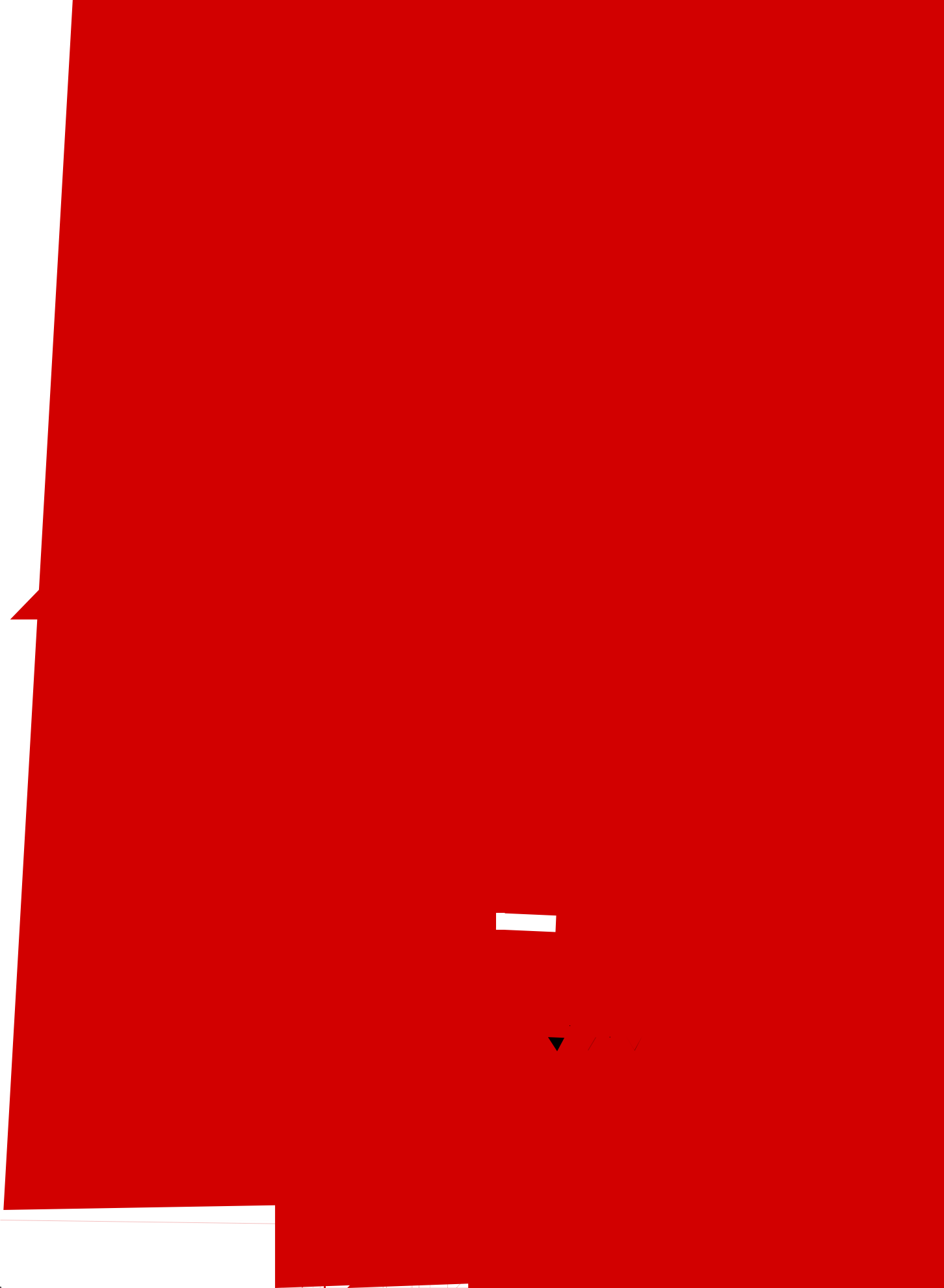


Fig. 10. Evolution of the horizontal velocity (mm/s) for the broken dam problem. Time steps 1, 25, 50, 75 and 100.





Although the results for a single problem can not be extrapolated, no spurious oscillations were found throughout the development of this work, nor in many other applications performed by the authors.

A very special case exists nevertheless. If we place the nodes on a regular lattice and few Gauss integration points are used, bilinear approximation for the velocity can be obtained (see [40] for more details, in that case bilinear approximation would be obtained at the Gauss points only). In this case, the resulting approximation will be equivalent to that of the bilinear-velocity, constant pressure quadrilateral FE, that does not verify the LBB conditions under certain boundary conditions (see [5]) This case is, however, very rare in a real application.

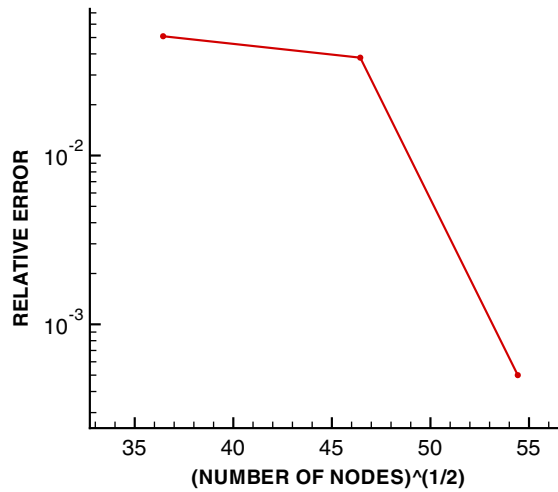
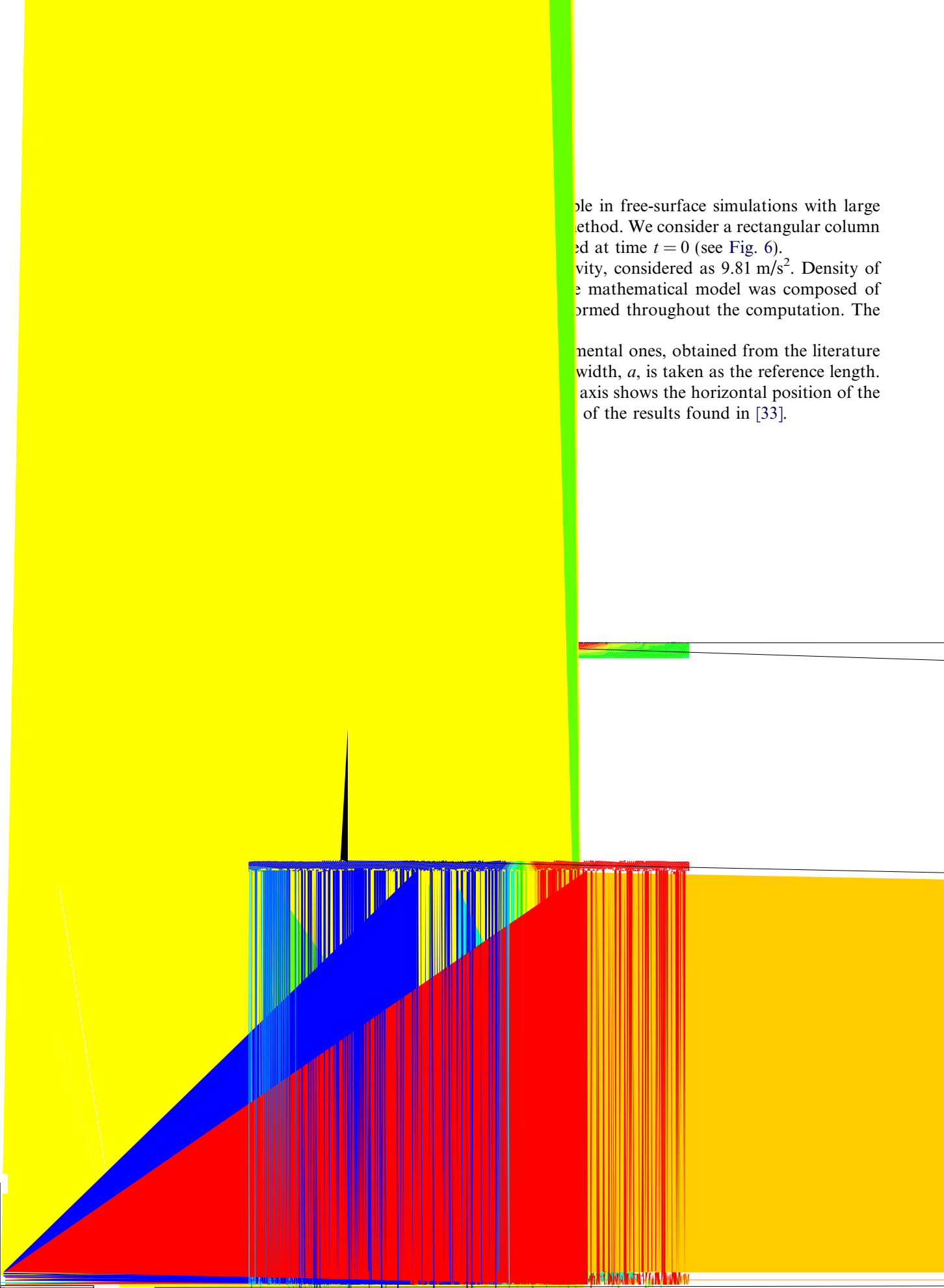


Fig. 14. Convergence of the error in the maximum height reached by the wave at the right wall.





ple in free-surface simulations with large method. We consider a rectangular column ed at time $t = 0$ (see Fig. 6). vity, considered as 9.81 m/s^2 . Density of e mathematical model was composed of rned throughout the computation. The

mental ones, obtained from the literature width, a , is taken as the reference length. axis shows the horizontal position of the of the results found in [33].

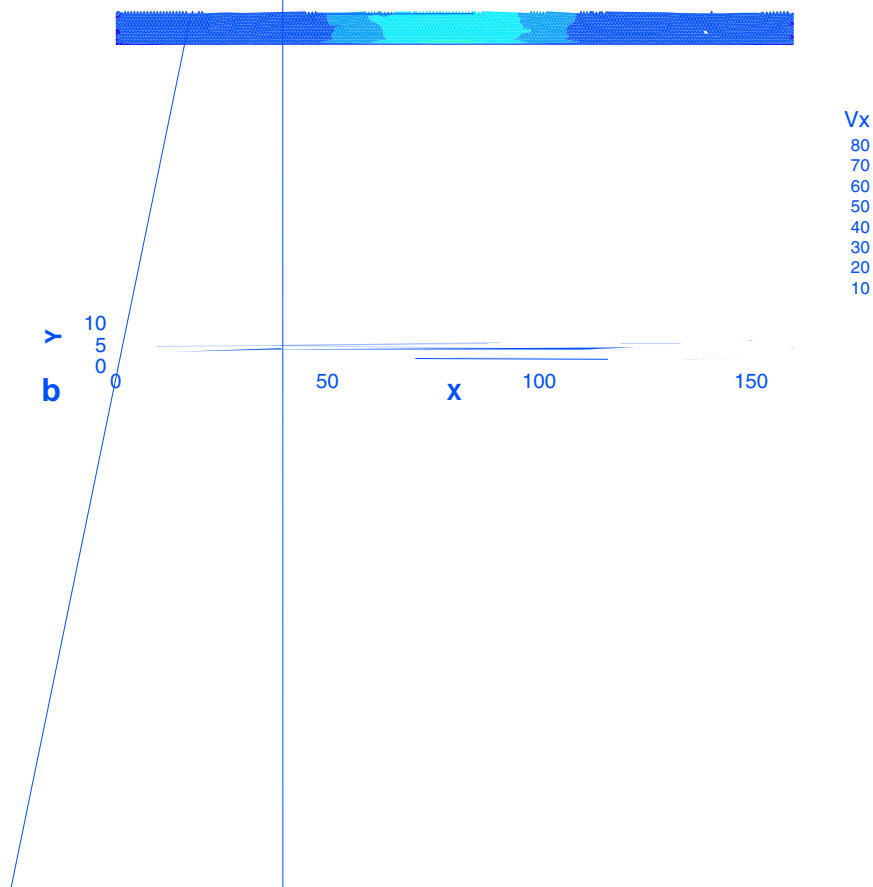
As can be noticed, excellent agreement is found between experimental and numerical results, despite the distortion of the triangulation. A detail of the triangulation is shown in Fig. 8.

In Fig. 9 the error in mass conservation is depicted, which remains always below 3%. Note that this example is especially exigent to the behaviour of the α -shape technique, since the circumradius of the triangles is continuously growing during the simulation, due to the particular form of fluid motion, that moves from a compacted geometry to a very flat and wide one. This makes the model to loose level of detail. This constitutes maybe the worst-case scenario for this technique. The subsequent simulations in this paper present much better conservation properties in this aspect, as will be seen. The influence of the relationship between the parameter α and the nodal parameter h on this error is deeply analyzed in [34]. If the error is judged to be too big, by simply adding new nodes we can reproduce again finer levels of detail. In addition, note that there is no restrictions in the positions of the new nodes in the cloud. In Figs. 10 and 11 the evolution of the horizontal and vertical components of the velocity, respectively, are depicted.

Despite the before-mentioned aspects, the resulting simulations show excellent accuracy, as can be seen in Fig. 7.

5.2. Analysis of the propagation of a solitary wave

In many branches of science and engineering, such the design of harbours, for instance, the analysis of the propagation of a solitary wave has utmost importance. In 1834, Russell studied experimentally the problem of



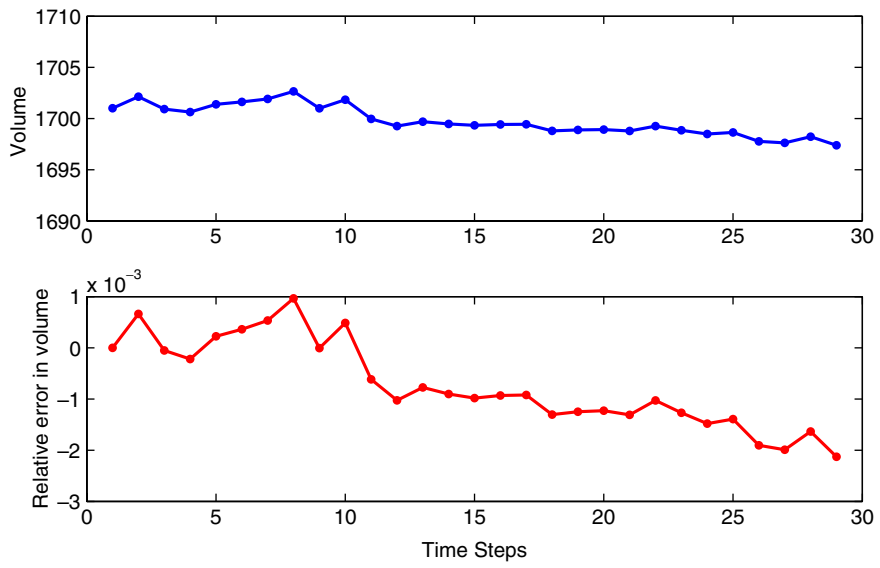


Fig. 19. Volume (top) and relative error (down) in the simulation of the propagation of a solitary wave.

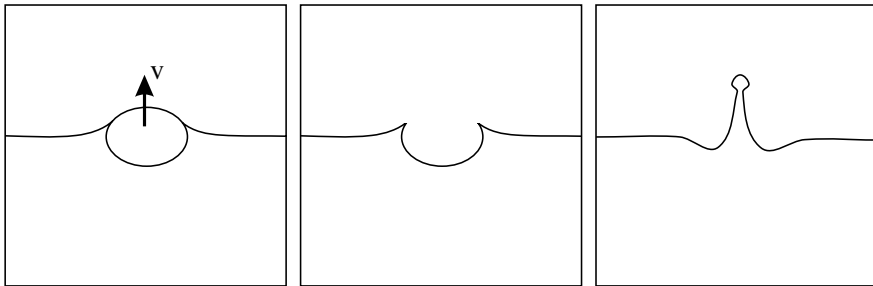
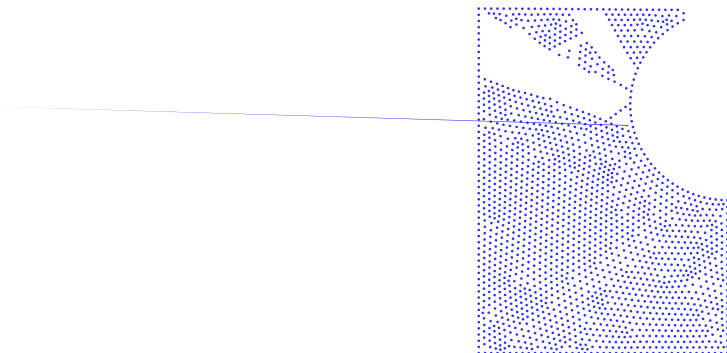


Fig. 20. Schematic representation of the formation of a jet after the rupture of a bubble.



a propagating wave in a constant-deep channel. Some analytical results exist on the shape of the wave, propagating along an ideally infinitely long channel [29].

Equations governing the behaviour of such a wave, due to Laitone, are the following:

$$u_1 = \sqrt{gd} \frac{H}{d} \operatorname{sech}^2 \left[\sqrt{\frac{3}{4}} \frac{H}{d^3} (x - ct) \right] \tag{35}$$

$$u_2 = \sqrt{3gd} \left(\frac{H}{d} \right)^{3/2} \left(\frac{y}{d} \right) \operatorname{sech}^2 \left[\sqrt{\frac{3}{4}} \frac{H}{d^3} (x - ct) \right] \tanh \left[\sqrt{\frac{3}{4}} \frac{H}{d^3} (x - ct) \right] \tag{36}$$

$$\eta = d + H \operatorname{sech}^2 \left[\sqrt{\frac{3}{4}} \frac{H}{d^3} (x - ct) \right] \tag{37}$$

$$c = \sqrt{gd \left(1 + \frac{H}{d} \right)}, \tag{38}$$

where H and d represent the initial height of the wave and the deep of the channel at a position where the water possesses null velocity, respectively. u_1 and u_2 represent the horizontal and vertical components of the velocity and, finally, η represents the free surface elevation. c is a parameter. See Fig. 12 for a schematic representation of the problem.

Laitone’s formulae are valid only for an infinitely long channel and ideally for an inviscid fluid. In our case, the simulations we performed include some (very low) viscosity. So, if we perform the simulation with

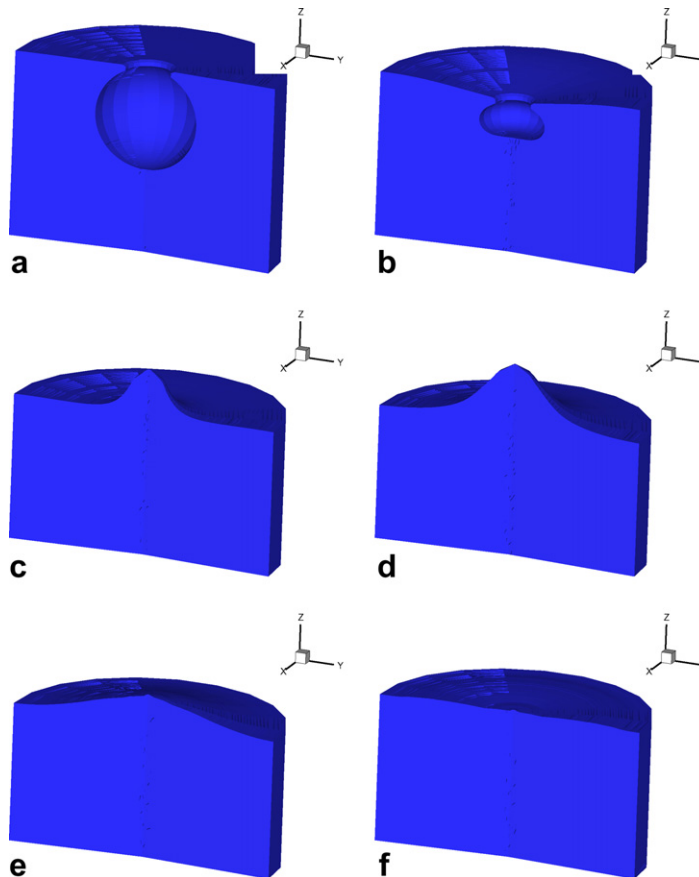


Fig. 22. Evolution of the free surface in the formation of the jet. Post-processing after axisymmetric results.

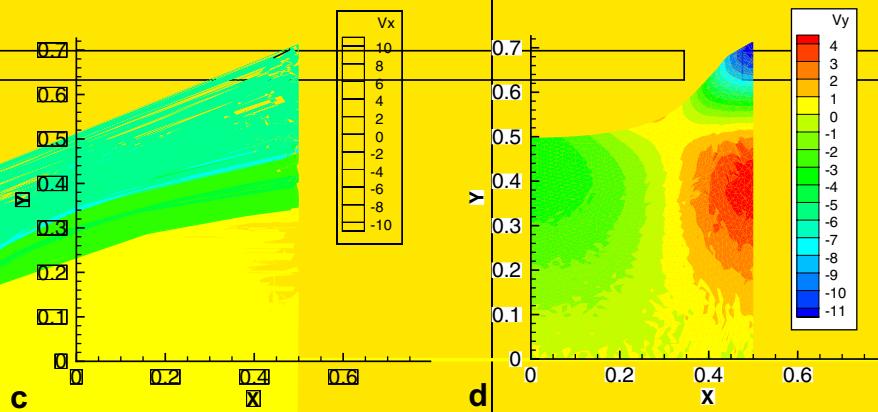


Fig. 23. Horizontal and vertical velocity plots (cm/s) for the initial configuration and for time $t = 0.015$ s.

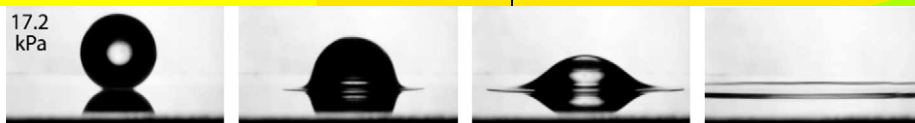


Fig. 24. Sequence of the drop deformation under very low ambient pressure. Photos courtesy of Lei Xu and Sidney Nagel.

Three different clouds of points were used to test the convergence of the proposed method. The clouds contained 1328, 2157 and 3815 nodes, respectively, and are shown in Fig. 13. Convergence of the relative error in the height reached by the wave at the right wall is shown in Fig. 14. The numerical result obtained in our simulation for 3815 nodes was $R/d = 0.42025$, see Figs. 15 and 16.

In Figs. 17(a)–(c) and 18(a)–(c) the vertical and horizontal component, respectively, of the velocity field is depicted for some intermediate time steps.

The error in volume for the simulation is shown in Fig. 19. In this case the error is much less than in the broken dam example. The level of detail is conserved constant throughout the simulation.

5.3. Jet formation in bubbles bursting at a free surface

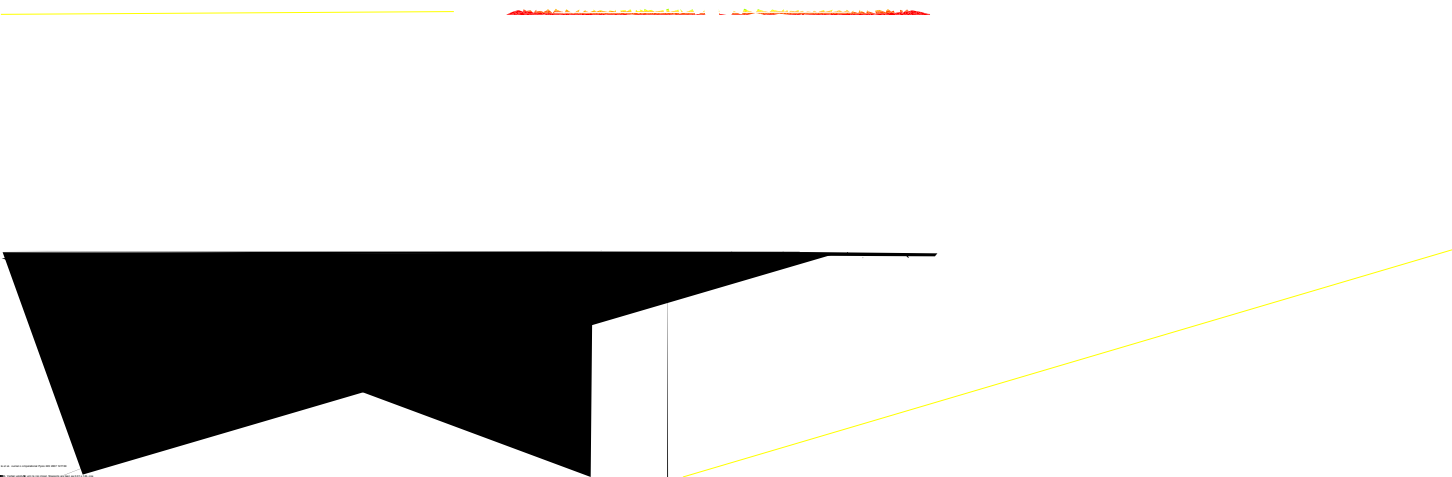
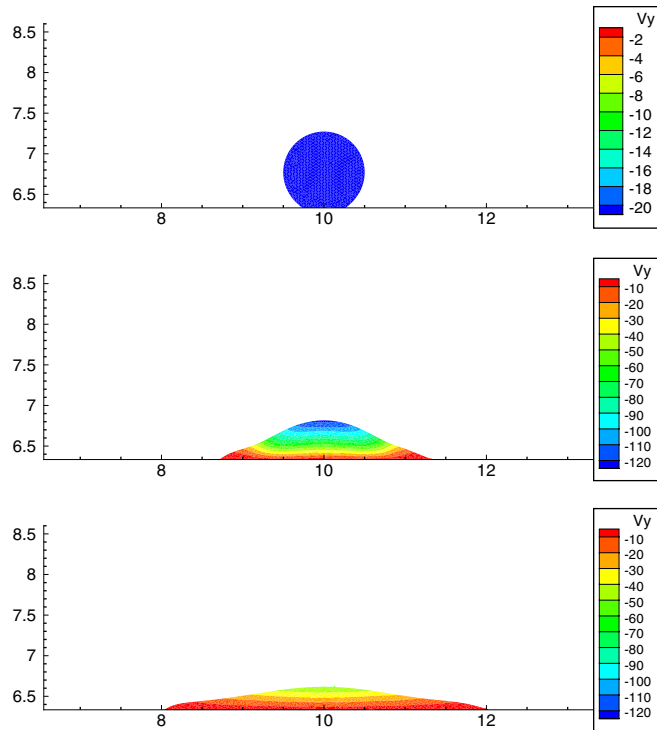
When a bubble gets onto the surface of a liquid a variety of physical phenomena occurs. These phenomena have been deeply studied from a numerical point of view in [21] by employing a method of marker and cells



Fig. 25. Evolution of the free surface in the drop impact.

and an Eulerian approach. The schematic sequence of the process is shown in Fig. 20. When the bubble reaches the free surface the thin film breaks, usually spreading out small drops of liquid. Then, the small cavity produced collapses under the action of surface tension and buoyancy, thus generating a jet. In this example, we will only consider the effect of buoyancy in the development of the jet.

For this purpose an axisymmetric formulation of the Navier–Stokes equations has been employed. The initial cloud of points is depicted in Fig. 21. The model is composed by 2146 nodes. Again, neither addition nor deletion of nodes was done during the simulation. Viscosity of the liquid was assumed to be 10^{-3} Pa s, with density 1000 kg/m^3 . Time steps of 0.001 s were employed. Some snapshots of the geometry of the free surface are depicted in Fig. 22. In Fig. 23 contour plots of the horizontal and vertical velocity are shown for the initial configuration and for time $t = 0.015 \text{ s}$.



It can be noticed the great similitude between these results and those in [21]. Although no analytical solution exists for this problem, the overall behaviour of drops approaching the free surface and consequent jet formation are captured accurately.

5.4. Drop impacting on a rigid surface

Impact of drops into rigid dry surfaces has been an active topic of research since the early stages of high velocity photography. Worthington [45] can be cited as the first in investigating the shape of the crown-shaped jet produced by the impact of a drop. It is now well known that this shape strongly depends on the surrounding pressure. This phenomenon has been recently studied in [46], although other references can be cited (see [25], for instance).

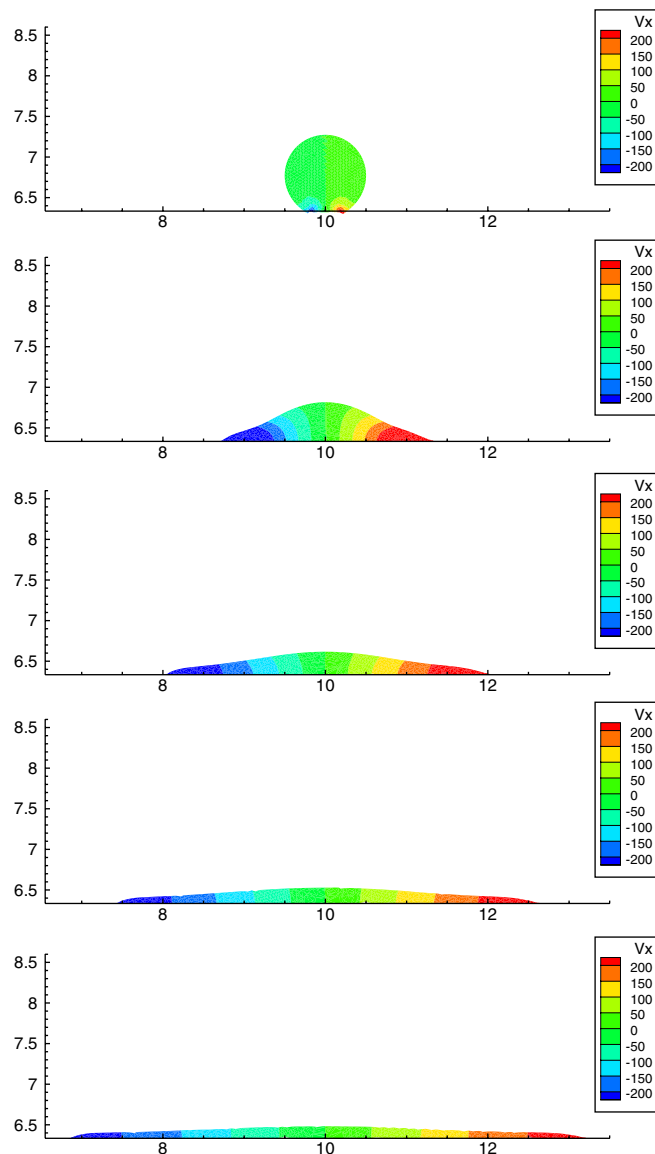


Fig. 27. Horizontal velocity field during the drop impact. Snapshots are taken each 0.01 s.

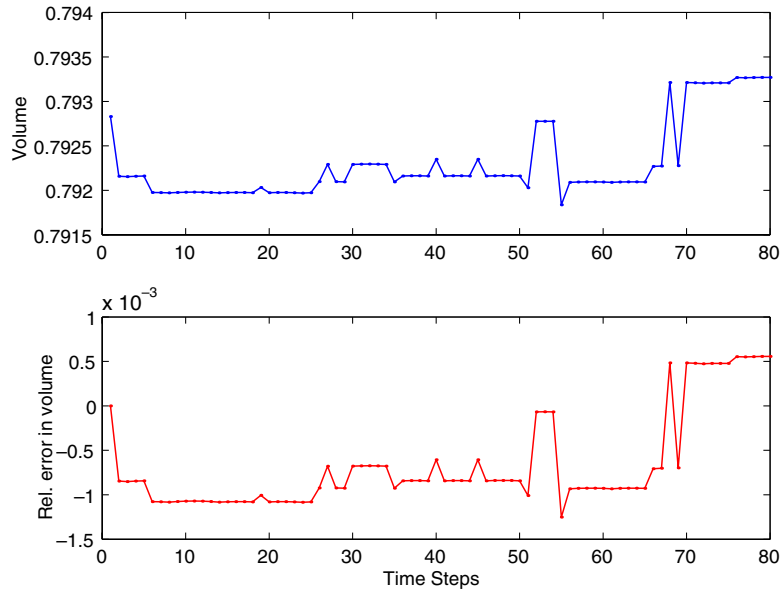


Fig. 28. Volume conservation in the splashing drop simulation.

It was argued in [46] that below a threshold in the ambient pressure, no drop is ejected from the main drop, nor jet formation is observed (see [46, pp. 184505–184502, Fig. 1], reproduced in Fig. 24). Instead, the formation of a “shoulder” in the drop is observed as it spreads on the surface [25].

By applying the just introduced technique, we have simulated the splashing of a drop on an infinitely rigid surface. The model is composed by 595 nodes. A viscosity of 10^{-3} Pa s was assumed and the usual density of 1000 kg/m^3 . A time increment of 0.002 s was considered in this simulation. The drop has an initial radius of 1 mm and is initially located at a position 10 mm above the surface. Then, the drop is left to the action of gravity.

In Fig. 25 the sequence of deformation of the drop is depicted. Note how, in spite of the large deformation of the drop, which becomes a thin film at the end of the simulation, the proposed method is able to reproduce the physical phenomena occurring during the impact. The velocity field is depicted in Figs. 26 and 27.

The volume conservation through the simulation is analyzed in Fig. 28. Note how the predicted volume is very close to the initial one, with a relative error of less than 0.2%.

6. Conclusions

In this paper, we have presented a method for the numerical simulation of free-surface dynamics of flows within an updated Lagrangian treatment. The method is based on the use of a natural neighbour characteristics-Galerkin method in conjunction with α -shapes. Time integration of the advection term in the Navier–Stokes equations is made by employing the method of characteristics, thus taking advantage of the lagrangian character of the method.

This technique has some remarkable properties within meshless methods, such as strictly interpolant character and exact imposition of essential boundary conditions. The method has been applied to the numerical simulation of some well-known physical phenomena, such as the so-called broken dam problem, a bubble bursting at a free surface or the impact of a drop on a rigid surface.

In all these examples the proposed method has behaved with reasonable accuracy, even in situations with extremely large deformations. We believe that its extension to three-dimensional problems, where methods based on Eulerian approaches such as markers or level set techniques, can become much more burdensome, will definitely show its potential of use.

Acknowledgements

The authors gratefully acknowledge Prof. Sidney Nagel and Lei Xu for their snapshots of the impacting drop.

References

- [1] S.N. Atluri, H.G. Kim, J.Y. Cho, A critical assesment of the truly meshless local Petrov–Galerkin and local boundary integral equation methods, *Computational Mechanics* 24 (1999) 348–372.
- [2] I. Babuška, the finite element method with Lagrange multipliers, *Numerische Mathematik* 20 (1973) 179–192.
- [3] I. Babuška, A. Aziz, On the angle condition in the finite element method, *SIAM Journal of Numerical Analysis* 13 (1976) 214–227.
- [4] I. Babuška, J.M. Melenk, The partition of unity method, *International Journal for Numerical Methods in Engineering* 40 (1997) 727–758.
- [5] K.J. Bathe, *Finite Element Procedures*, Prentice-Hall, 1996.
- [6] V.V. Belikov, V.D. Ivanov, V.K. Kontorovich, S.A. Korytnik, A.Yu. Semenov, 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.
- [7] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, *International Journal for Numerical Methods in Engineering* 37 (1994) 229–256.
- [9] H. Braess, P. Wriggers, Arbitrary Lagrangian Eulerian finite element analysis of free surface flow, *Computer Methods in Applied Mechanics and Engineering* 190 (2000) 95–109.
- [10] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from lagrange multipliers, *Revue Française d'Automatique Informatique Recherche Operationelle, Analyse Numérique* 8 (1974) 129–151.
- [11] D. Chapelle, K.J. Bathe, The inf-sup test, *Computers and Structures* 47 (4–5) (1993) 537–545.
- [12] Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, Yang You, A stabilized conforming nodal integration for galerkin mesh-free methods, *International Journal for Numerical Methods in Engineering* 50 (2001) 435–466.
- [13] E. Cueto, B. Calvo, M. Doblaré, 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.
- [14] E. Cueto, J. Cegoñino, B. Calvo, M. Doblaré, On the imposition of essential boundary conditions in natural neighbour Galerkin methods, *Communications in Numerical Methods in Engineering* 19 (5) (2003) 361–376.
- [15] E. Cueto, M. Doblaré, L. Gracia, 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.
- [16] E. Cueto, N. Sukumar, B. Calvo, M.A. Martínez, J. Cegoñino, M. Doblaré, Overview and recent advances in natural neighbour Galerkin methods, *Archives of Computational Methods in Engineering* 10 (4) (2003) 307–384.
- [17] T.K. Dey, J. Giesen, M. John, Alpha-shapes and flow shapes are homotopy equivalent. In: *Proceeding Symposium on Theory Computing (STOC 2003)*, 2003, pp. 493–502.
- [18] J. Dolbow, T. Belytschko, Numerical integration of the Galerkin weak form in meshfree methods, *Computational Mechanics* 23 (1999) 219–230.
- [19] J. Donea, Arbitrary Lagrangian–Eulerian finite element methods, in: T. Belytschko, T.J.R. Hughes (Eds.), *Computer Methods for Transient Analyses*, Elsevier, Amsterdam, 1983, pp. 474–516.
- [20] J. Donea, A. Huerta, *Finite Element Methods for Flow Problems*, Wiley, 2003.
- [21] L. Duchemin, S. Popinet, C. Josserand, S. Zaleski, Jet formation in bubbles bursting at a free surface, *Physics of Fluids* 14 (2002) 3000–3008.
- [22] H. Edelsbrunner, E. Mücke, Three dimensional alpha shapes, *ACM Transactions on Graphics* 13 (1994) 43–72.
- [23] D. Gonzalez, E. Cueto, M.A. Martinez, M. Doblaré, Numerical integration in natural neighbour Galerkin methods, *International Journal for Numerical Methods in Engineering* 60 (12) (2004) 2077–2104.
- [24] D. González, E. Cueto, M. Doblaré, Volumetric locking in natural neighbour Galerkin methods, *International Journal for Numerical Methods in Engineering* 61 (4) (2004) 611–632.
- [25] F.H. Harlow, J.P. Shannon, The splash of a liquid drop, *Journal of Applied Physics* 38 (10) (1967) 3855–3866.
- [26] H. Hiyoshi, K. Sugihara, Two generalizations of an interpolant based on Voronoi diagrams, *International Journal of Shape Modeling* 5 (2) (1999) 219–231.
- [27] S.R. Idelsohn, E. Oñate, N. Calvo, F. del Pin, The meshless finite element method, *International Journal for Numerical Methods in Engineering* 58 (2003) 893–912.
- [28] S.R. Idelsohn, E. Oñate, F. del Pin, The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves, *International Journal for Numerical Methods in Engineering* 61 (7) (2004) 964–989.
- [29] E.V. Laitone, The second approximation to cnoidal and solitary waves, *Journal of Fluid Mechanics* 9 (1960) 430–444.
- [30] R.W. Lewis, S.E. Navti, C. Taylor, A mixed Lagrangian–Eulerian approach to modelling fluid flow during mould filling, *International Journal for Numerical Methods in Engineering* 25 (1997) 931–952.
- [31] W.K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods, *International Journal for Numerical Methods in Engineering* 38 (1995) 1655–1679.

- [32] Q.W. Ma, Meshless local Petrov–Galerkin method for two-dimensional nonlinear water wave problems, *Journal of Computational Physics* 205 (2) (2004) 611–625.
- [33] J.C. Martin, W.J. Moyce, Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane, *Philosophical Transactions of the Royal Society of London* 244 (1952) 312.
- [34] M.A. Martínez, E. Cueto, I. Alfaro, M. Doblaré, F. Chinesta, Updated Lagrangian free surface flow simulations with natural neighbour Galerkin methods, *International Journal for Numerical Methods in Engineering* 60 (13) (2004) 2105–2129.
- [35] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements, *Computational Mechanics* 10 (1992) 307–318.
- [36] O. Pironneau, *Finite Element Methods for Fluids*, Wiley, 1989.
- [37] B. Ramaswamy, Numerical simulation of unsteady viscous free surface flow, *Journal of Computational Physics* 90 (1990) 396–430.
- [38] R. Sibson, A vector identity for the Dirichlet tessellation, *Mathematical Proceedings of the Cambridge Philosophical Society* 87 (1980) 151–155.
- [39] R. Sibson, A brief description of natural neighbour interpolation, in: V. Barnett (Ed.), *Interpreting Multivariate Data*, Wiley, 1981, pp. 21–36.
- [40] N. Sukumar, B. Moran, T. Belytschko, The natural element method in solid mechanics, *International Journal for Numerical Methods in Engineering* 43 (5) (1998) 839–887.
- [41] N. Sukumar, B. Moran, A. Yu Semenov, V.V. Belikov, Natural neighbor Galerkin methods, *International Journal for Numerical Methods in Engineering* 50 (1) (2001) 1–27.
- [42] A.H. Thiessen, Precipitation averages for large areas, *Monthly Weather Report* 39 (1911) 1082–1084.
- [43] S.O. Unverdi, G. Tryggvason, A front tracking method for viscous incompressible flows, *Journal of Computational Physics* 100 (1992) 25–37.
- [44] D. Watson, Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes, *The Computer Journal* 24 (2) (1981) 162–172.
- [45] A.M. Worthington, On the forms assumed by drops of liquid falling on a horizontal plate, *Philosophical Transactions of the Royal Society of London* 25 (1877) 261.
- [46] L. Xu, W.W. Zhang, S.R. Nagel, Drop splashing on a dry smooth surface, *Physical Review Letters* 94 (2005) 184505.
- [47] J. Yvonnet, D. Ryckelynck, P. Lorong, F. Chinesta, A new extension of the natural element method for non-convex and discontinuous problems: the constrained natural element method, *International Journal for Numerical Methods in Engineering* 60 (8) (2004) 1452–1474.