The natural volume method (NVM): Presentation and application to shallow water inviscid flows

Riadh Ata, Azzeddine Soulaïmani, Francisco Chinesta

The natural volume method (NVM): Presentation and application to shallow water inviscid flows

R. Ata[1], A. Soulaïmani[1] and F. Chinesta[2,†]

[1]*Department of Mechanical Engineering, École de Technologie Supérieure, 1100 Notre-Dame West, Montreal, Que., Canada H3C 1K3*

[2]*LMSP, École Nationale Supérieure d'Arts et Métiers de Paris, 151 boulevard de l'Hôpital, F-75013 Paris, France*

In this paper a fully Lagrangian formulation is used to simulate 2D shallow water inviscid flows. The natural element method (NEM), which has been used successfully with several solid and fluid mechanics applications, is used to approximate the fluxes over Voronoi cells. This particle-based method has shown huge potential in terms of handling problems involving large deformations. Its main advantage lies in the interpolant character of its shape function and consequently the ease it allows with respect to the imposition of Dirichlet boundary conditions.

In this paper, we use the NEM collocationally, and in a Lagrangian kinematic description, in order to simulate shallow water flows that are boundary moving problems. This formulation is ultimately shown to constitute a finite-volume methodology requiring a flux computation on Voronoi cells rather than the standard elements, in a triangular or quadrilateral mesh. St Venant equations are used as the mathematical model. These equations have discontinuous solutions that physically represent the existence of shock waves, meaning that stabilization issues have thus been considered. An artificial viscosity deduced from an analogy with Riemann solvers is introduced to upwind the scheme and therefore stabilize the method. Some inviscid bidimensional flows were used as preliminary benchmark tests, which produced decent results, leading to well-founded hopes for the future of this method in real applications.

KEY WORDS: NEM; shallow water equations; stabilization; Lagrangian description; collocation; inviscid flows

1. INTRODUCTION

Free surface flows are often simulated in shallow domains using Eulerian descriptions. Most software applications use the finite-element or the finite-volume method in Eulerian formulations. However, several difficulties arise in simulating some phenomena related to the Lagrangian behavior of the flow, such as in moving boundary problems or moving shocks. The Lagrangian description is thus preferred for overcoming most such difficulties. Fully Lagrangian applications are few in the literature and we can mostly find applications that use the so-called *moving mesh methods*. The improvement in accuracy is achieved by adapting a fixed mesh so that nodes are concentrated in the areas where the solution presents high gradients (such as shocks or boundary layer areas). The mesh is thus moved using equations that involve node velocities. Normally, the partial differential equation (PDE) and the mesh equations are solved simultaneously. Examples of moving mesh methods include the *moving finite-difference methods* (for instance, see Dorfi and Drury [1]) and the *moving finite-element method* (FEM) [2, 3]. More recently, the *meshless FEM* was introduced by Idelsohn *et al.* in [4] and applied in fluid mechanics with free surface in [5]. In the same family of methods, Huang and Russell [6] used a cell averaging cubic Hermite collocation discretization for the physical PDEs combined with a finite-difference discretization for the PDE to solve problems with discontinuous solutions.

Most works covering these methods employ conventional numerical approaches. Actually, in the last decades, meshless- or mesh-free methods have surfaced as a serious mean of handling problems that could not be handled conventionally; these include problems presenting large distortion and fragmentation, and problems with unconventional behaviors. These methods are used mostly in the Galerkin formulations and some of them in a collocation manner. The family of mesh-free methods are used on a wide variety of mechanical and electrical applications. In hydraulics, Monaghan has explored the smooth particle hydrodynamics (SPH) method for free surface flows using an isentropic state law [7]. Many others have used the same law to simulate several problems in fluid mechanics, including Bonet *et al.* [8] who covered the use of SPH in fluids. The volume of fluid technique applied to the Navier–Stokes equations gave spectacular results when simulating free surface flows with breaking waves and for sloshing problems [9]. Ata and Soulaïmani presented a stabilized SPH method for inviscid shallow water flows using the St Venant equations [10], where they showed that the stabilization problem can be solved using an artificial viscosity arising from the upwinding of the flux in Riemann solvers. The treatment of boundary conditions, nonetheless, remains a problem that is yet to be solved. The non-interpolant nature of the shape function of almost all the mesh-free methods, including SPH, represents the main handicap when imposing Dirichlet boundary conditions. These methods are therefore only well adapted to simulating unbounded-domain problems in fluid mechanics. The natural element method (NEM) represents a serious mean of overcoming such problems; the NEM has the advantage of having an interpolant shape function.

The NEM was originally a Galerkin-based method. It is considered to fall under the meshless method family even though it uses the Delaunay triangulation or the Voronoi diagram to build its shape function. The NEM uses the notion of natural neighbor interpolation, an interpolation that has very striking properties such as a strictly interpolating character, the ability to exactly interpolate piece-wise linear boundary conditions on convex domain, and a well-defined and robust approximation with no user-defined parameters on non-uniform grids [11]. For a non-convex domain, the NEM does not reproduce linear boundary conditions. A counterexample is shown in [12]. This problem could be overcome by the use of α-shape technique [11] or a constrained Voronoi diagram [13].

This method was first used for the interpolation of data from a set of scattered points [14]. Recently, it has been used for some applications in solid mechanics [15], such as those involving multiscale problems [16] and crack simulation [13]. Liang *et al.* [17] used the Sibson NEM for the interpolation of data on new inserted nodes when applying the finite-volume method (FVM) on a dynamically adaptive quadtree grid. Del Pin [18] used the NEM shape function as a kernel to implement the moving-particle semi-implicit method in a Lagrangian formulation; the latter is a moving finite-difference method.

In this paper, the NEM is investigated in order to explore its capacity to simulate free surface flows in a fully Lagrangian formulation. The NEM's shape function is defined on Voronoi cells, giving rise to a non-standard Lagrangian finite-volume scheme called *the natural volume method* (*NVM*).

We begin by giving a brief description of the NEM in Section 2 and we recall the C-NEM method in Section 3. In Section 4, we present the shallow water equations (SWE) and their discretization in the NVM framework. Finally, results on 2D applications are shown.

2. THE NATURAL ELEMENT METHOD

In the NEM, interpolation functions (the trial and test functions for Galerkin formulation) are constructed using the notion of natural neighbors. Natural neighbor co-ordinates are based on well-known geometric concepts such as the Voronoi diagram and the Delaunay tessellation [19]. It was Sibson [14] who first presented the notion of natural neighbor interpolation for data approximation. Further works on their construction and properties were carried out later by Farin [20] and others. Recently, a new type of interpolation (non-Sibsonian), which is similar to but less costly than Sibson's, has been introduced by Belikov *et al.* [21]. In fact, the non-Sibsonian interpolant requires the computation of Lebesgue measures of order $D-1$ in $\mathbb{R}^D$ to compute the shape functions, whereas for the Sibson interpolant, a computation of areas (in 2D) or volumes (in 3D) is necessary.

2.1. Voronoi diagram and Delaunay tessellation

The Voronoi (or Dirichlet) diagram is a unique geometric construct obtained from a set of distinct nodes in a Euclidian space of dimension D.

Let $\Im$ be a set of M distinct nodes of a bounded domain Ω: $\Im=\{n_1,n_2,\ldots,n_M\}$. The Voronoi diagram associated with the set $\Im$ is the subdivision of the domain into regions $V(n_I)$ such that each point in $V(n_I)$ is the closest to node n_I than to any other node $n_J\in\Im$ $(I\neq J)$. The region $V(n_I)$ (first-order Voronoi cell) of node n_I is the convex polygon (polyhedron):

$$V(n_I)=\{x\in\mathbb{R}^D:d(x,x_I)<d(x,x_J)\ \forall J\neq I\} \tag{1}$$

where d is a Lebesgue measure defined on Ω.

The dual of the Voronoi diagram is the Delaunay tessellation. The Delaunay triangles are constructed by connecting nodes whose Voronoi cells have common boundaries. It is important to note that the Voronoi diagram is unique, even if the Delaunay tessellation is not. Figure 1 shows the triangulation and its dual Voronoi diagram for the case of 9 nodes scattered in a square in 2D.

For the sake of simplicity, from now on we will consider the space dimension $D=2$.

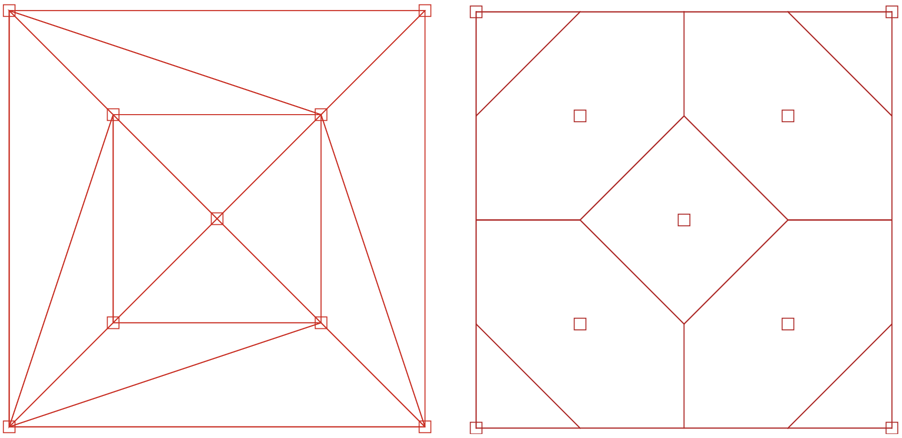

Figure 1. Delaunay triangulation (left) and its relative Voronoi diagram (right) for a 9-node domain.

2.2. *Sibsonian interpolants* [*14*]

The definition of the first-order Voronoi diagram was introduced by Equation (1). Similarly, we can extend this definition to a higher order $k>1$. Then, we define the Voronoi diagram of second order as

$$V_{IJ}=\{x\in\mathbb{R}^2 : d(x,x_I)<d(x,x_J)<d(x,x_K)\ \forall K\neq I,J\} \tag{2}$$

The second-order Voronoi diagram then becomes the subdivision of the plane (or the space) into domains V_{IJ} such that V_{IJ} is the locus of all points that have I as the closest node and J as the second closest node.

Sibson has used the second-order Voronoi diagram notion to define the natural neighbor-type interpolation function. This function can be used as a trial or test function in Galerkin-type formulations.

If a point P with coordinates $\mathbf{x}$ is to be inserted into the Delaunay tessellation, natural neighbor-type shape function associated with P is defined as the ratio of the area of the second-order Voronoi cell $A_I(P)$ to the total area of the first-order Voronoi cell $A(P)$ associated with P (see Figure 2):

$$\phi_I(\mathbf{x})=\frac{A_I(P)}{A(P)}, \qquad A(P)=\sum_{J=1}^{n} A_J(P) \tag{3}$$

where n is the number of natural neighbors of P. The derivatives of the Sibson-type shape function could be obtained by deriving Equation (3) as follows:

$$\phi_{I,j}(\mathbf{x})=\frac{A_{I,j}(P)-\phi_I(\mathbf{x})A_{,j}(P)}{A(P)}, \qquad j=1,2 \tag{4}$$

For more details and to consult exhaustive studies on all characteristics of the Sibson interpolation, we refer to References [11, 22].

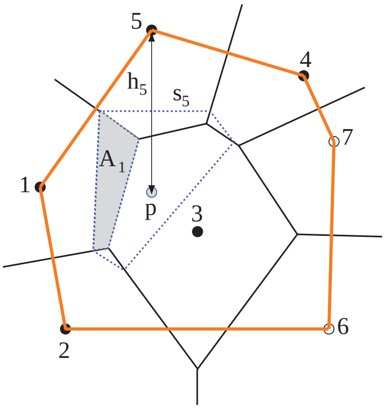

Figure 2. Sibson *vs* Laplace interpolations [19].

2.3. *Non-Sibsonian or Laplace interpolation*

Let $\Im$ be a set of nodes to which are associated their Voronoi cells defined by Equation (1). Let t_{IJ} be the facet associated with cells V_I and V_J (of dimension $D-1$: segment in 2D or polygon in 3D) and $d(t_{IJ})$ the measure of Lebesgue of t_{IJ}. If any point P with coordinates $\mathbf{x} \in \mathbb{R}^2$ is introduced into the tessellation and if this point has n natural neighbors, then the Laplace shape function associated with the node I, in the 2D case, is [19, 21, 23]

$$\phi_I(\mathbf{x}) = \frac{\alpha_I(\mathbf{x})}{\sum_{J=1}^{n} \alpha_J(\mathbf{x})}, \quad \alpha_J(\mathbf{x}) = \frac{s_J(\mathbf{x})}{h_J(\mathbf{x})} \tag{5}$$

where $\alpha_J(\mathbf{x})$ is the Laplace weight function, $s_I(\mathbf{x})$ is the length of the Voronoi edge associated with P and node I, and $h_I(\mathbf{x})$ is the Euclidian distance between points P and I (see Figure 2). The derivative of the Laplace interpolation function is

$$\phi_{I,j}(\mathbf{x}) = \frac{\alpha_{I,j}(\mathbf{x}) - \phi_I(\mathbf{x})\alpha_{,j}(\mathbf{x})}{\alpha(\mathbf{x})} \tag{6}$$

with $\alpha(\mathbf{x}) = \sum_J \alpha_J(\mathbf{x})$. In 2D, the Laplace shape function uses a fraction of length whereas Sibson's function uses a fraction of areas, and as a result the computational cost is lower for the Laplace interpolation. This advantage is more remarkable in the 3D case.

2.4. *Properties*

In addition to positivity, the most important properties of the natural element-type interpolation given by Equation (3) or (5) are listed below

- The interpolation is defined from the nodal values according to

$$u^h(\mathbf{x}) = \sum_{I}^{n} \phi_I(\mathbf{x}) u_I \tag{7}$$

- Nodal interpolation: This can be expressed as

$$\phi_I(\mathbf{x}_J) = \delta_{IJ} \tag{8}$$

  with δ_{IJ} being the delta Kronecker function. Thus, as for the cases of FEM or FVM, the NEM is an interpolant method. This property is very important as it simplifies the imposition of Dirichlet boundary conditions. This property is lost for most of the meshless method family, which therefore represents a major drawback for them.
- The partition of unity: Sibson and Laplace interpolations possess zero-order consistency or the partition of unity property

$$\sum_I^N \phi_I(\mathbf{x}) = 1 \tag{9}$$

  Consequently, all NEM interpolations reproduce exactly constant functions.
- Linear completeness: The NEM-type interpolations reproduce exactly linear functions, which yield their linear completeness

$$x = \sum_I^N \phi_I(\mathbf{x}) x_I \tag{10}$$

  See the demonstration in [11].
- Strict linearity on the border: At the boundary of a convex domain, the shape function is strictly linear between two successive nodes; a proof can be found in [15].
- The support nature: The support of NEM-type interpolation functions is the union of Delaunay circumcircles of node I. Figure 3 presents a comparison between the NEM support and the moving least-squares (MLS) support [19]. For Galerkin-type formulations, this shape of the support makes the numerical integration a relatively tough task when compared with finite elements, for instance. Besides, another difficulty lies on the fact that the shape function is rational, which complicates the integration task more. For our work, the collocation formulation avoids all these drawbacks; for further details, see [24].
- In 2D, the implementation of Bowyer–Watson algorithm [25, 26] can be adopted to compute the Sibson- and Laplace-type shape functions. The Lasserre algorithm has been used also in [27]. The Watson algorithm is not extensible in 3D. Besides, it fails on the nodes located on the Delaunay edges. Thus, the Bowyer–Watson algorithm is efficient to evaluate the Laplace shape function in 2D [11]. For a full description and for more details on this algorithm, we refer, for instance, to Cueto *et al.* [11].

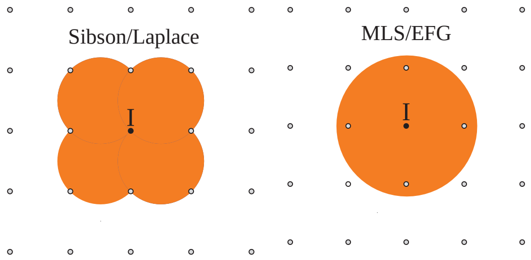

Figure 3. Support of the NEM-type interpolation function *vs* those of MLS and EFG—figure from [19].

3. THE CONSTRAINED NEM OR C-NEM

To overcome the problems seen in the NEM in its standard formulation when handling the non-convex character of a domain, Cueto *et al.* [28] first proposed a new technique called ‘alpha shape,’ which is a generalization of the concept of the convex hull of a cloud of points. Yvonnet *et al.* [22] suggested an original approach based on the use of the constrained Voronoi diagram, which was introduced by Klein and Lingas [29]. Giving a constrained Voronoi diagram, the C-NEM-type interpolation is given as: An unknown function u is approximated by $u^h(\mathbf{x})$ given by

$$u^h(\mathbf{x}) = \sum_{I=1}^{N} \phi_I^{\mathrm{C}}(\mathbf{x}) u_I \tag{11}$$

where N is the number of natural neighbors visible from x and ϕ_I^{C} are the constrained natural neighbor shape functions. u_I are the nodal unknown values of natural neighbors visible from the point $\mathbf{x}$ [22].

It is very important to note that when a point $\mathbf{x}$ comes close to an edge or a face of Γ, the contour of the concerned Voronoi cells is extended to infinity to the outside of Ω such that the contribution in area (in 2D) or in volume (in 3D) is infinite. These infinite contributions allow the retrieval of the linearity of the interpolation on the boundary of the domain. Figure 4 shows the C-NEM-type shape functions (Sibson and Laplace).

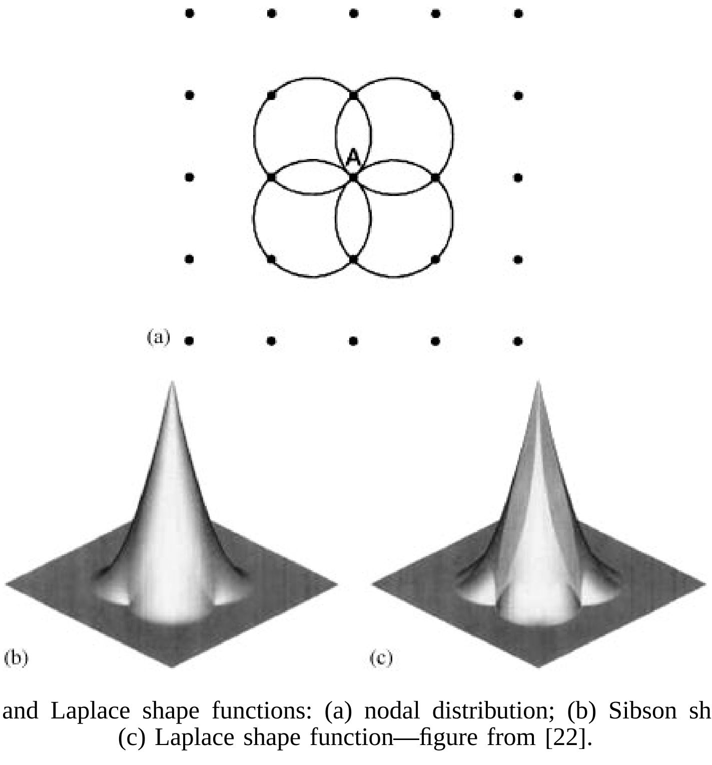

Figure 4. Sibson and Laplace shape functions: (a) nodal distribution; (b) Sibson shape function; and (c) Laplace shape function—figure from [22].

4. PROBLEM SETTING

4.1. The Shallow water equations (SWE)

We attempt to find a numerical solution to the SWE using the NEM. The NEM was used to simulate some fluid mechanics problems especially relating to metal forming. The updated Lagrangian formulation was mostly used [30]. That is the case of the FEM when applied to treat SWE [31]. González *et al.* [32] recently presented an updated Lagrangian application to some free surface problems in fluid dynamics using Navier–stokes equations with moderate Reynolds numbers. No fully Lagrangian application of the NEM seems to be available in the literature. The rapid distortion of used mesh, and thus the frequent re-meshing, represents the biggest challenge to overcome for scientists. The paper by Birknes and Pedersen [33] could be taken as a typical work. It introduces the use of the 'particle FEM (PFEM)' to the SWE to simulate the long wave run-up. The originality of this work lies in its use of a mixed Eulerian–Lagrangian formulation of the PFEM.

The inviscid SWE in the non-conservative form when neglecting the bed slope and friction terms are given by

$$\frac{\mathrm{D}h}{\mathrm{D}t}+h\nabla\cdot\mathbf{u}=0 \tag{12}$$

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}+g\nabla h=0 \tag{13}$$

where $h, \mathbf{u}$, and g are, respectively, water height, depth-average velocity, and gravity. $\mathrm{D}/\mathrm{D}t$ refers to the total derivative. Cases that introduce bed slope and friction terms will be covered in future works.

4.2. Mass conservation

Usually, continuity equation (12) is used for mass conservation. However, although we use a Lagrangian formulation of the NEM, the mass is automatically conserved. In fact, the total mass of the fluid is divided into N particles, where N is the total number of nodes. Each particle is assigned to a node such that, in any given time step, the mass (the volume for an incompressible medium) of each particle is conserved. The volume of a particle I is given by $V_I = A_I h_I$, where A_I is the Voronoi cell area and h_I is the averaged water depth over the area. The new value of the depth h_I^{n+1} at the time $t^{n+1}=t^n+\delta t$ is computed as follows:

$$V_I^{n+1}=h_I^{n+1}A_I^{n+1}=V_I^n=h_I^nA_I^n \implies h_I^{n+1}=h_I^n\frac{A_I^n}{A_I^{n+1}} \tag{14}$$

A_I^{n+1} is the area of the deformed cell associated with the node I at the time t^{n+1}. A_I^{n+1} is obtained after time integration of the coordinates of all the vertices of the Voronoi cell associated with node I. It is important to note that the geometric Voronoi cell, i.e. the cell related to node I corresponding to the dual of the Delaunay triangulation, does not correspond to the one we use to compute the new water depth. In fact, to ensure the local mass conservation, we have to use the initial Voronoi diagram (at $t=0\,\mathrm{s}$) and update its coordinates at each time step (Figure 5).

The coordinates updating of the Voronoi diagram needs the computation of the acceleration and the velocity values at the vertices of each cell. Such computations require that first we stock the

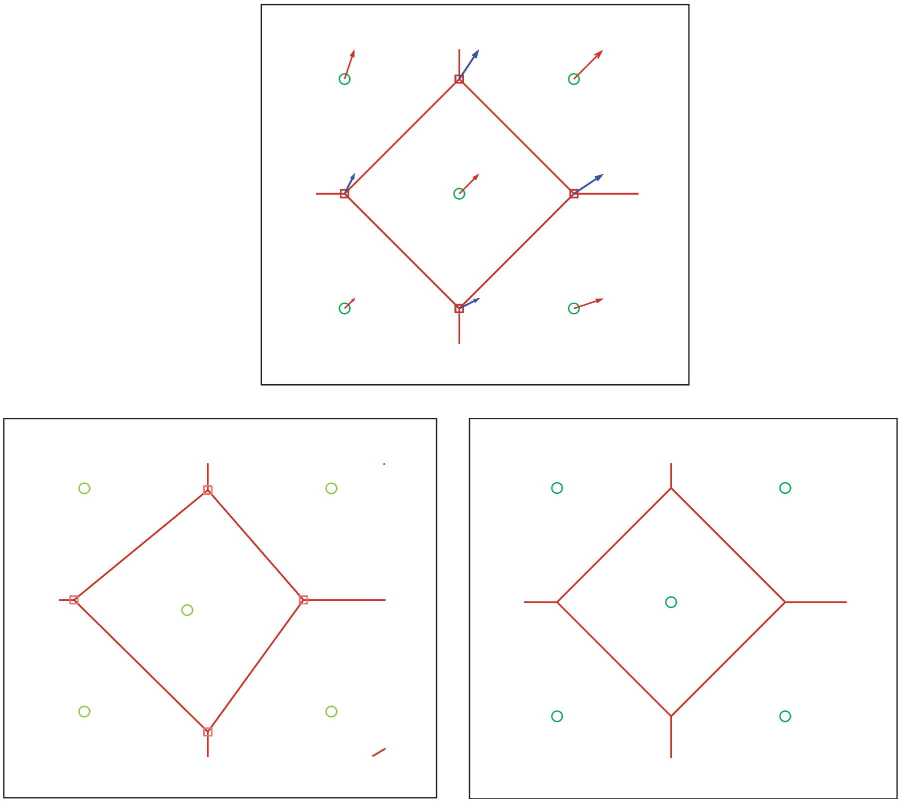

Figure 5. The mass conservation strategy (○: node, □: ‘mass’ cell vertex, →: velocity). Up: domain decomposition at time t^n. Down right: real Voronoi cell at time t^{n+1} used for momentum computation. Down left: mass cell or sub-domain used for mass conservation and water depth computation at time t^{n+1}.

coordinates and the old values of acceleration and velocity. Furthermore, we need to ‘interpolate’ such functions for each time step. Interpolation (11) can be used to carry out that purpose. For instance, to compute the velocity at a vertex $\mathbf{s}$ of the cell of a node I, $\mathbf{s}$ is first introduced in the tessellation, and the velocity is given by

$$u^h(\mathbf{s}) = \sum_{J=1}^{N} \phi_J^{\mathrm{C}}(\mathbf{s}) u_J \tag{15}$$

where $J = 1, \ldots, N$ are the visible natural neighbors of $\mathbf{s}$.

4.3. *Conservation of momentum*

We look for the application of the NEM in a collocation form to simulate shallow water flows using St Venant equations. Although, most NEM applications are in a Galerkin formulation, a finite-difference scheme of the method was first proposed by Sukumar in [34], and then discussed extensively in [11]. Unlike Sukumar and Cueto’s use of the terminology of finite difference, we prefer to use the term of finite volumes while it consists of the use of averaged values of the flux on all Voronoi diagram edges.

For a node I located at $\mathbf{x}$, the discrete nodal gradient can be approximated directly by deriving Equation (5) in the coordinate directions. This approach is adopted in the most meshless methods when used in a collocation manner. As a second way to approximate the nodal gradient, we can consider the finite-volume averaging for the strain operator, which was proposed by Chen *et al.* [35]. Generally, the jth component of the gradient of a field w is given by

$$\left(\widetilde{\frac{\partial w^h}{\partial x_j}}\right)_I = \frac{\int_{A_I} w^h_{,j}(\mathbf{x})\,\mathrm{d}A}{A_I} = \frac{\int_{\Gamma_I} w^h n_j\,\mathrm{d}\Gamma}{A_I} \tag{16}$$

where w is the field to be derived, A_I is the area, and Γ_I is the boundary of the Voronoi cell related to node I. n_j is the jth component of the unit outward normal vector to Γ_I. The Gauss divergence theorem has been used in Equation (16).

For the special case of a regular grid, we can approximate the integral using an average weighted value of w^h on each Voronoi edge [11]. The choice $w^h(x_m)=(w_I+w_J)/2$ renders an exact result for a linear field w on a regular grid ($\mathbf{x}_m$ is the middle point of the segment $[IJ]$):

$$\left(\widetilde{\frac{\partial w^h}{\partial x_j}}\right)_I = \frac{\sum_{J=1}^{N}(w_I+w_J)n_j^{IJ}s_{IJ}}{2A_I}, \quad A_I=\frac{1}{4}\sum_{J=1}^{N} s_{IJ}h_{IJ} \tag{17}$$

where n_j^{IJ} is the jth component of the unit outward normal to the Voronoi edge common to nodes I and J and s_{IJ} is the length of this edge, and h_{IJ} is the length of the segment $[IJ]$. We have a linear finite-element discretization here. In fact, the finite-element approximation is linear on the edges of the triangle and in the case of a regular nodal distribution, the middle point of the Voronoi edge relative to the segment $[IJ]$ is located on the edge of the triangle with I and J as vertices.

With the approximation of the gradient on hand, one can proceed to write a discrete version of the SWE. For a particle I, we apply Equation (17) to evaluate the gradient of the pressure $P=gh$, following which Equation (13) is expressed as

$$\frac{\mathrm{D}\mathbf{u}_I}{\mathrm{D}t} = -\frac{1}{2A_I}\sum_{J=1}^{N}[(gh_I+gh_J)\mathbf{n}^J s_{IJ}] \tag{18}$$

where $\mathbf{u}$ is the velocity field, g is the gravity acceleration, h_I is the water depth of the particle I, and N is the number of edges of the cell related to the particle I. Note that N is not always equal to the number of natural neighbors of I, and that it is only in the case of a regular distribution of nodes and for an internal node (far from the boundary) that N corresponds to the number of visible natural neighbors.

The approximation given by Equation (18) is the easiest and lowest cost choice. However, it is limited to cases where the nodal distribution is either mostly regular or where the solution is linear or quasi-linear. For more complicated cases, any other approximation type may be used (finite element, finite difference, natural element, or any Godunov-type scheme, etc.). For our case, the use of the Voronoi diagram favors the trivial choice of the natural element approximation.

As mentioned above, Equation (18) may only be used for cases with regular nodal distribution or simple geometries and for linear or quasi-linear solutions. In our case, the Lagrangian simulation induces a relative chaotic displacement of nodes. Therefore, the regularity of the initial nodal distribution is quickly lost. Figure 6 shows a very simple example where, for three nodes $\{1,2,3\}$, Equation (18) cannot be applied. In fact, for the Voronoi edge relative to nodes 1 and 2, the interpolation of any function w in the middle point of this edge using $w^h(x_m)=(w_I+w_J)/2$ is far from accurate. To overcome this problem, an interpolation of the flux on the Voronoi edges is achieved using the NEM. The easiest way to quantify Equation (16), i.e. to compute the flux on the Voronoi edge, is to integrate this equation on a unique integration point (the middle point of the edge). For the case of a convection–diffusion problem under some regularity conditions, Mishev demonstrated, in [36], that this quadrature (as well as the one given by Equation (18)) ensures a $\mathcal{O}(h^{s-1})$ convergence, where s is the order of the Sobolev space (H^s) in which the solution is defined and regular (see the work of Mishev [36] for a full mathematical study of this quadrature scheme)

$$\left(\frac{\partial \widetilde{w^h}}{\partial x_j}\right)_I = \frac{\int_{A_I} w^h_{,j}(\mathbf{x})\,\mathrm{d}\Omega}{A_I} = \frac{\int_{\partial A_I} w^h n_j\,\mathrm{d}\Gamma}{A_I} = \frac{1}{A_I}\sum_J^N (\breve{w}_{IJ} s_J n_j^{IJ}) \tag{19}$$

where $\breve{w}_{IJ}$ is the NEM approximation of w estimated at M_J, the middle point of the edge, s_J is the edge length, and N is the number of edges of the Voronoi cell of I. $\breve{w}_{IJ}$ is obtained by applying Equation (11), which is a 'richer' approximation than the finite-element approximation used by Chen in Equation (17) as it involves all the natural neighbors of M_J rather than only I and J for the other approximations. Applying Equation (19) on the non-conservative form of the momentum equation of the SWE (Equation (13)) gives

$$\dot{\mathbf{u}} = \frac{\mathrm{D}\mathbf{u}_I}{\mathrm{D}t} = -\frac{1}{A_I}\sum_{J=1}^N (g\breve{h}_J)\mathbf{n}^{IJ} s_J \tag{20}$$

where $\breve{h}_J = \sum_{K=1}^{N_J} \phi_K(M_J) h_K$ and N_J is the number of edges in the Voronoi cell related to M_J. Using the identity given by $\oint \mathbf{n}\,\mathrm{d}s = 0$, or in a discretized form $\sum_{J=1}^N \mathbf{n}^{IJ} s_J = 0$, Equation (20)

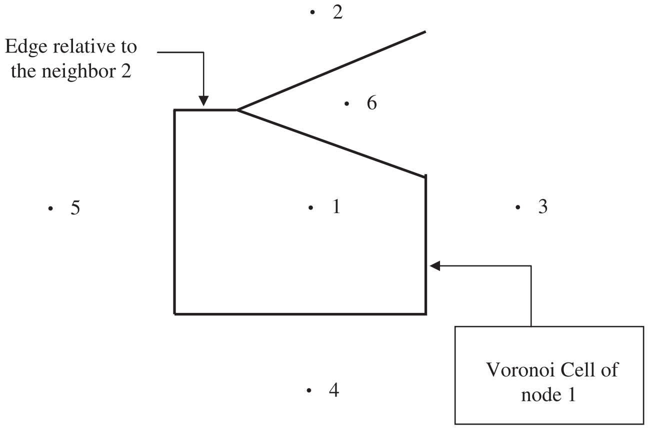

Figure 6. The case frequently obtained in Lagrangian simulations where Equation (18) does not produce good approximation (especially for neighbor 2).

becomes

$$\dot{\mathbf{u}}_I = \frac{\mathrm{D}\mathbf{u}_I}{\mathrm{D}t} = -\frac{1}{A_I}\sum_{J=1}^{N}(g\breve{h}_J + g\breve{h}_I)\mathbf{n}^{IJ}s_J \tag{21}$$

Equation (21) is the final approximation of the momentum equation, which is preferred to Equation (18). This equation will be used for all tests that will be discussed in Section 10. However, owing to the symmetry of the scheme given by this equation, an upwinding technique must be used for stabilization purposes.

Remark
As described in the previous paragraphs, the Voronoi diagram is constructed at each time step. This procedure induces a geometrical reconstruction of the cells associated with each particle at every time step. Rigorously, the time derivative of the momentum should be discretized, for instance, as $\mathrm{D}m_I u_I/\mathrm{D}t \simeq (h_I^{n+1}A_I^{n+1}u_I^{n+1} - h_I^n A_I^n u_I^n)/\Delta t$, which defines a non-linear problem (while h_I^{n+1} and A_I^{n+1} depend on u_I^{n+1}) that should be treated implicitly. However, an implicit treatment becomes very expensive in CPU time. For this reason, we opted for an explicit linearization, which assumes that $h^{n+1}A^{n+1} \simeq h^n A^n$.

5. CONSISTENCY AND STABILIZATION ISSUES

The study of the consistency of the method on non-uniform grid poses interesting challenges. The consistency basically ensures that in the limit, when the grid spacing tends to zero, the approximated solution tends to the analytical one. Using Taylor series, Cueto *et al.* [11] have demonstrated that using the finite-difference method, the approximation of the Laplacian operator is $O(h_m)$ for a non-uniform grid, with h_m being the mean of the internodal distance. At the very least, a first-order consistency is thus obtained on non-uniform grids. In the case of a regular grid, the consistency order raises to the second order. For more details, see the paper by Sukumar [34]. In our applications, the regularity of the nodal distribution cannot be conserved in the time because of the Lagrangian character of the simulations. That is why, we cannot guarantee more than a first-order consistency.

While the solutions for the SWE are discontinuous, shock waves are encountered when simulating shallow flows. On the other hand, Equations (18) and (21) offer a symmetric centered-difference-like scheme that is are mostly unstable. That is why a stabilization technique must be used in order to avoid spurious spatial oscillations that may grow in amplitude especially near discontinuities. As is the case in Ata and Soulaïmani [10], we can adopt a stabilization technique by the analogy of what is often adopted with Riemann solvers. Doing so will lead to an upwinding of the scheme given by Equations (18) and (21) as it will introduce an artificial viscosity.

The quantity $F_h(x_I) = gh_I$ can be considered as the flux through the Voronoi edge. Thus, Equation (18) could be expressed as

$$\dot{\mathbf{u}}_I = \left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right)_I = -\frac{1}{A_I}\sum_{J=1}^{N}\left[\frac{(F_h(\mathbf{x}_I) + F_h(\mathbf{x}_J))}{2}\mathbf{n}^{IJ}s_{IJ}\right] \tag{22}$$

For the purpose of stabilization, the term $(F_h(\mathbf{x}_I)+F_h(\mathbf{x}_J))/2$ in Equation (22) is replaced by a simple Lax–Friedrich scheme given by

$$\tfrac{1}{2}((F_h(\mathbf{x}_I)+F_h(\mathbf{x}_J))-\lambda(\mathbf{u}_J-\mathbf{u}_I)\cdot\mathbf{n}^{IJ}) \tag{23}$$

We can thus define an artificial viscosity that will stabilize the system and will especially not allow interpenetration between particles. This artificial viscosity is given by (we refer to [10] for more details)

$$\Pi_{IJ}=\alpha\frac{-\lambda\mathbf{u}_{IJ}\cdot\mathbf{r}_{IJ}}{\sqrt{\mathbf{r}_{IJ}^2+\varepsilon^2}}=\alpha\frac{-\lambda[(v_J-v_I)(y_J-y_I)+(u_J-u_I)(x_J-x_I)]}{\sqrt{\mathbf{r}_{IJ}^2+\varepsilon^2}} \tag{24}$$

where $\lambda=\frac{1}{2}(c_I+c_J)=\frac{1}{2}(\sqrt{gh_I}+\sqrt{gh_J})$ is the averaged wave speed, $\mathbf{u}_{IJ}=\mathbf{u}_I-\mathbf{u}_J$ the difference between the velocities of I and J. u and v are the components of the velocity field in the x and y directions, respectively, $\mathbf{r}_{IJ}=\mathbf{r}_I-\mathbf{r}_J$, ε is a small constant used to avoid the division by zero, and α is a free parameter. The final momentum equation will be expressed as

$$\dot{\mathbf{u}}_I=\left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right)_I=-\frac{1}{2A_I}\sum_{J=1}^{N}((gh_I+gh_J)+\Pi_{IJ})\mathbf{n}^{IJ}s_{IJ} \tag{25}$$

The stabilization-upwinding term Π_{IJ} is introduced in the same manner as for Equation (25). The final form of the discretized moment equation that we used is then

$$\dot{\mathbf{u}}_I=\left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\right)_I=-\frac{1}{A_I}\sum_{J=1}^{N}(g\breve{h}_J+g\breve{h}_I+\Pi_{IJ})\mathbf{n}^{IJ}s_J \tag{26}$$

Remarks

1. We are using two different cells: one for the mass and one for momentum conservation. The Voronoi cell is the one used for the momentum equation; the cell used for the computation of the water depth is a ‘squeezed’ Voronoi cell and then it does not verify any of the known properties of the Voronoi cell.
2. As for all Lagrangian formulations with large deformations, we could possibly see distorted cells appear. The distortion could appear for cells used for mass conservation (which are Voronoi cells at $t=0$ s and only at $t=0$ s). It cannot affect Voronoi cells which are always convex and which can never be distorted in a finite-element manner. In such a case, the reference configuration would be updated and even a mass redistribution could be required. For all the cases presented in Section 10, the cells did not attain a high level of distortion, and so we did not update any of them.

6. BOUNDARY CONDITIONS

The treatment of the boundary conditions is greatly simplified for the NEM as compared with other meshless methods. The interpolant character of the method simplifies the Dirichlet boundary imposition. For our cases, we need a perfect slipping condition on the wall, meaning that the normal component of the velocity and acceleration fields of particles situated on the boundary

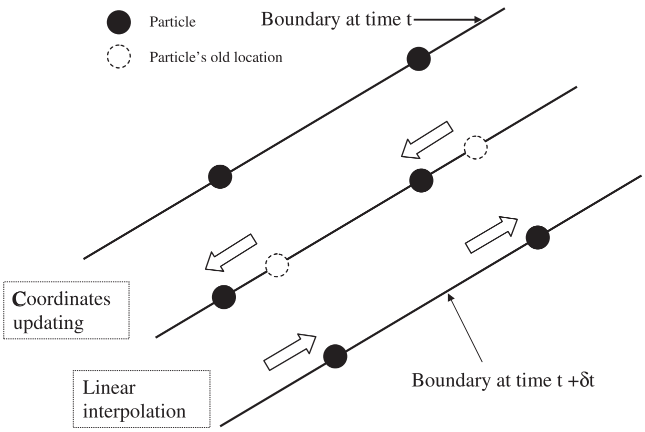

Figure 7. Boundary treatment: a particle on the boundary is fixed and its physical parameters ($\dot{\mathbf{u}}$, $\mathbf{u}$, and h) are interpolated from parameters of boundary particles at time t^{n+1}.

have to be zero

$$u_I^n = 0 \quad \text{and} \quad \dot{u}_I^n = 0 \;\; \forall I \text{ on the domain boundary} \tag{27}$$

where u_I^n and $\dot{u}_I^n$ are the normal components of the velocity and the acceleration of the particle I. To that end, velocity and acceleration vectors can be projected onto the tangent vector.

The Lagrangian character of the formulation induces several difficulties in the treatment of boundary conditions especially when the geometry is complex. In fact, the boundary of the domain (the geometry of the channel) is defined by a set of geometrical nodes. A fluid particle that moves at every time step is attached to every node. Therefore, the geometry could be lost or at least modified when the particles of the boundary move. Thus, to maintain a well-defined geometry, the boundary particles are fixed to the geometric locations of their corresponding nodes. This procedure must physically ensure a perfect slipping condition. To compute the value of acceleration, velocity, or water depth at the old position of the particle (the geometric position), a linear interpolation is done using the two closest particles. Figure 7 illustrates such a procedure. We must pay particular attention to a particle located in a corner or at an angular point in general. In this case, a direction is chosen (privileged) between the two directions that form the corner. However, the direction that is ‘parallel’ to the flow direction matches better with the flow’s nature and naturally captures the physics of the problem. Such a problem is well known in all finite-volumes and finite-elements-type approaches. It is one of the points to be studied in further works.

We can also let boundary particles move freely at the edges of the domain. Such a procedure guarantees better results whereas it does not require accelerations’ or velocities’ interpolation. However, it causes more difficulties when following a particle over the boundary. The angular points and corners are handled in a similar manner as in the first procedure.

7. TIME INTEGRATION

The time integration is treated using an explicit scheme. This avoids the high resolution of systems but requires specific time step selection in order to stabilize time integration. In this work, we used an explicit Euler scheme in some cases, but mostly it was Newmark explicit scheme that we used. The velocity and the coordinates of a particle I are given by the following:

$$\mathbf{u}_I^{n+1} = \mathbf{u}_I^n + \Delta t((1-\gamma)\dot{\mathbf{u}}_I^n + \gamma\dot{\mathbf{u}}_I^{n+1}) \tag{28}$$

$$\mathbf{x}_I^{n+1} = \mathbf{x}_I^n + \Delta t\mathbf{u}_I^n + \Delta t^2((\tfrac{1}{2}-\beta)\dot{\mathbf{u}}_I^n + \beta\dot{\mathbf{u}}_I^{n+1}) \tag{29}$$

where Δt is the time step, $\mathbf{u}$ is the velocity, $\dot{\mathbf{u}}$ is the acceleration, and the notation $\mathbf{u}^n$ means $\mathbf{u}(t^n)$. By choosing the parameters γ and β as, respectively, $\frac{1}{2}$ and $\frac{1}{4}$, the scheme corresponds to the trapezoidal rule and is second-order accurate [37].

Loukili and Soulaïmani suggested in [38] that according to a linearized stability analysis applied to the SWE, the following Courant–Friedrichs–Lewy (CFL) restriction must be respected:

$$\text{CFL} = \Delta t \frac{\max_I(\sqrt{gh} + \sqrt{u_x^2 + u_y^2})}{\min(d_{IJ})} < 1 \quad \forall I, J \tag{30}$$

For most finite-volume applications in Eulerian formulation, the CFL value is often greater than 0.5. In our cases, when particles are moving, and the distances between them change with every time step, we reduce the CFL values to a maximum of 0.5.

8. ADAPTIVITY AND NODES REFINEMENT

Given the fact that the computational procedure is fully Lagrangian, the nodes' distribution changes with every time step. Therefore, in certain cases, we could obtain strongly irregular particle concentration in which a big number of nodes is located in a small part of the domain, and the remaining few nodes are scattered across the other part. Actually, the introduction of an artificial viscosity given by Equation (24) ensures a minimum regularity for the node distribution, and as long as there is no interpenetration between particles, the Lagrangian aspect of the formulation may dominate the viscosity effect. In such cases, an adaptation or a node redistribution on the entire domain will become necessary. That is what is known as 'adaptivity' or 'mesh refinement.'

The adaptivity procedure must be carried out with respect to the mass conservation. The total mass of the fluid flowing in the channel, before and after nodal refinement, must be conserved. In our case, this condition is easily respected. To begin the refinement procedure, we need to use an indicator that will provide information regarding whether the concentration of some particles is increasing in a particular area, which could lead to two or more particles sticking together in short order. The indicator we use in this work is based on the distance between the nodes. For every node I, the following entity is computed:

$$R = \frac{\min_J d_{IJ}}{\min_J d_{IJ}^0}, \quad J \text{ is a natural neighbor of } I \tag{31}$$

where d_{IJ} is the distance between nodes I and J at time t and d_{IJ}^0 is the distance between nodes I and J at time $t = 0$ s. When R is less than 10%, it means that particle I is too close to one of its neighbors and this kicks in the adaptivity procedure. The two particles must be merged into one particle located at an equal distance between them. The new mass is now equal to the sum of the two masses, and the acceleration and velocity are interpolated linearly.

If the indicator R increases up to 190%, then an insertion of a node is achieved. The new particle will be located at an equal distance between the particle and its furthest neighbor. The mass is distributed between them avoiding the occurrence of any numerical perturbance. To that end, few iterations are completed in order to stabilize the simulation before going on to the next time step.

It should be noted here that for the benchmark tests conducted in this work and which are presented in the following paragraphs, the adaptivity procedure was used only when adding new nodes in corners.

9. COMPUTATIONAL ALGORITHM

The global algorithm of the method is given as follows:

1. Carry out geometrical initialization and construct the Voronoi diagram.
2. Stock all Voronoi cell lattices. They will be used for water depth computing.
3. For every time step:
   - For every node $I = 1, 2, \ldots, M$ the acceleration is computed using Equation (26).
   - If I is a boundary node, treat it as described in Section 6.
   - The velocity is computed in every node using time integration.
   - Using Equation (11), compute the accelerations at the initial Voronoi lattices (for mass conservation).
   - Update the velocities and the coordinates of the initial Voronoi diagram lattices using Equations (28) and (29).
   - Update the particle coordinates using Equation (29).
   - With the new node positions, re-mesh the entire domain: Delaunay triangulation and Voronoi subdivision.
   - Compute the new water depths using Equation (14).
   - Compute the new time step using Equation (30).

It should be noted that Voronoi cell reconstruction in every time step is simply a ‘luxury’ we accorded ourselves, while we possess a performing and fast Voronoi constructor. However, it does ensure better accuracy and consequently good results.

10. NUMERICAL EXAMPLES

In the following, we will present some applications that represent a validation step of our new approach. As seen in the preceding paragraphs, the formulation we present is applied to inviscid shallow water flows modelized with St Venant equations (12) and (13) by neglecting source terms (friction and bed slope terms). The introduction of these terms will be covered in future publications.

10.1. Rectangular channel

The first benchmark test is obviously the rectangular section channel. This case is a generalization of the 1D case in which the transversal component does not—or only slightly—influence the behavior of the flow. This flow could be seen as a set of 1D flows set one next to the other

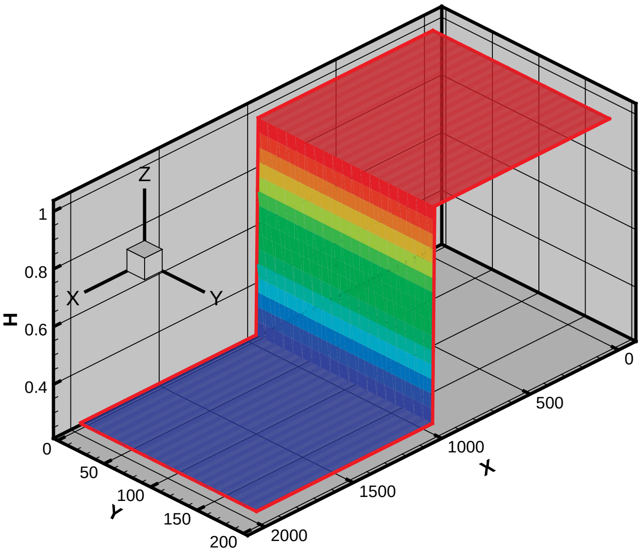

Figure 8. Initial conditions for the rectangular channel dam break problem.

transversally. An analytical solution for the 1D case was given by Stoker in [39]. The initial conditions are presented in Figure 8. The water depth is 1 m upstream and 0.25 m downstream. The channel length is 2000 m and its width is 80 m. The dam is located at $x = 1000$ m. The depth discontinuity is linearized on a distance of 15 m. The number of particles used for the simulation was between 1000 and 30 000. However, a minimum of 1000 in the longitudinal direction is necessary to well capture the shape of the solution. In Figure 8, 10 000 particles were used as 10×1000 particles.

The shock-capturing term (24) was used with a free parameter α chosen at between 0.5 and 2. In this case $\alpha = 2$ is used. We can note, therefore, that the solution of the 1D dam break case is properly reproduced. Figure 9 shows a tri-dimensional view of the depth at $t = 15$ s, whereas Figure 10 makes a comparison with the analytic solution. We can see the presence of the diffusive effect of the artificial viscosity especially at the rarefaction wave location. This could be explained by the fact that the Lax–Friedrich scheme does not include any MUSCL-type reconstruction to reduce the diffusive effect commonly associated with first-order Godunov-type schemes [10]. Therefore, the shock capturing is very successful and the stabilization precess works well when there is no numerical oscillation. The use of a free parameter as a flux limiter that depends on the depth gradient (such as $\alpha = \alpha_0 |h_i - h_j| / h_i$ with $\alpha_0 = 2$) gives a better result from a stabilization point of view, but has more diffusive effect.

Figure 11 shows the evolution of L^2 error in time. We can see that this latter remains acceptable even though the Lagrangian character of the flow implies frequent relative displacement between particles, which could affect the solution accuracy and thus could increase the global error.

10.2. Convergent and divergent channels

For now, we will try to explore the effect of the geometry on the solution. We will first examine the influence of a transversal component on the behavior of the solution. To that end, we will present

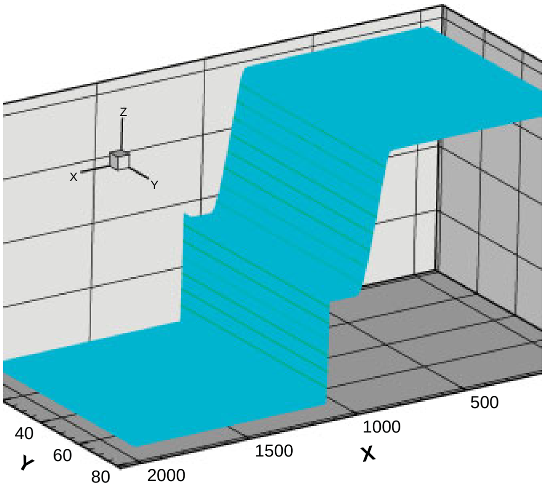

Figure 9. Water depth at time $t = 15$ s.

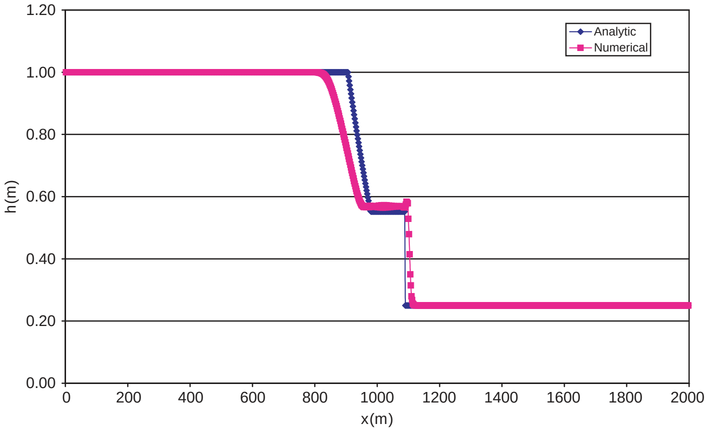

Figure 10. Comparison with the analytical solution.

the cases of a convergent channel and of a divergent channel. Figure 12 presents the divergent and convergent channel cases. The water depth is 1 m upstream and 0.25 m downstream. The artificial viscosity parameters are the same as in the last test. To achieve this simulation 10 760 particles were used. Here the domain length was still 2000 m and the dam was located at $x = 1000$ m, whereas the discontinuity was linearized over a distance of 15 m.

As we can see in these figures, the introduction of the transversal component of the flow due to the change in the geometry in this direction does not affect the performance of the solution. The shocks are well captured and the stabilization tools perform as in the last case.

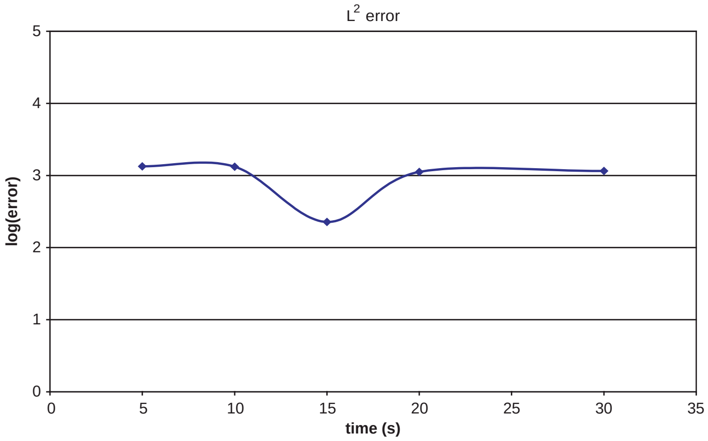

Figure 11. Evolution of L^2 error in time.

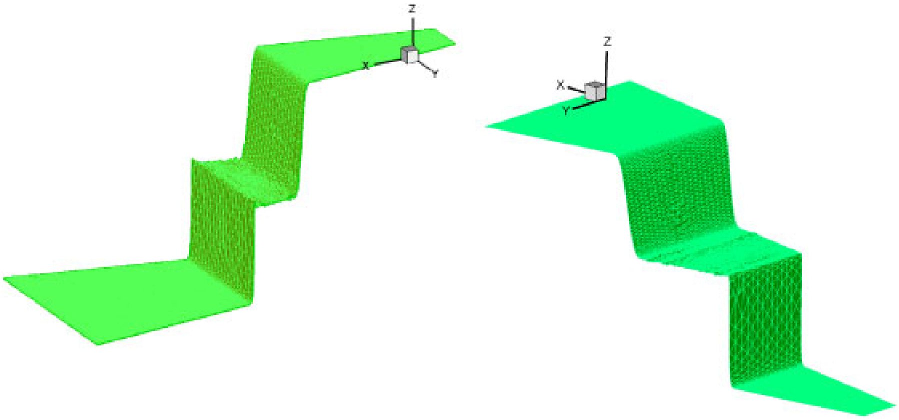

Figure 12. Divergent channel at time $t = 15$ s (left) and convergent channel at $t = 30$ s (right).

Figure 13 gives us an idea about the velocity distribution on a divergent channel. We can see that the velocity vectors follow the geometry of the domain and the perfect slipping condition is verified.

10.3. Cylindrical dam break

In his book [40], Toro presents the cylindrical dam break test. It consists of a cylindrical dam with a radius $R = 5$ m located in an unbounded domain. For the numerical implementation, the domain is taken as measuring 100 m × 100 m; the water depth is 1 m in the dam and 0.95 m outside; the number of particles used here is 2490 and it was increased up to about 10 000. Figures 14 and 15 illustrate this case. The behavior of the method with respect to the number of particles and the variation of the artificial viscosity parameter is quite similar as illustrated in Ata and Soulaïmani [10] to which we therefore refer the reader for more details. In our case, we will focus on the

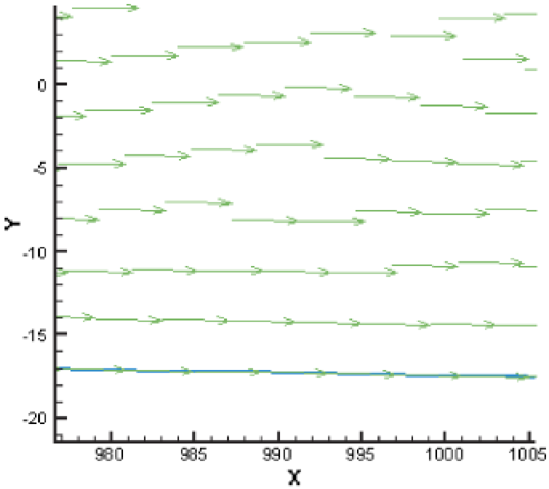

Figure 13. Velocity distribution for the case of divergent channel (zoom).

importance of the 'nature' of the mesh or the initial nodal distribution. Figure 14 shows the case of a structured mesh of 80 × 80. This figure gives the initial nodal distribution, the initial mesh, the contour of the solution, the velocity distribution and, finally, a 3D view of the water depth. As we can see, the axisymmetry of the problem is lost once the wave starts to move. This problem is overcome in the second case. In fact, Figure 15 illustrates the case of a pseudo-circular nodal distribution. The quality of the solution is strongly improved and, in particular, the axisymmetry of the solution is almost revealed. The caption of shocks is well done and the birth of a second wave in the middle of the domain is shown clearly in this figure. Actually, we should note the presence of numerical oscillations in both cases and especially in Figure 15. This could be explained by the low nodal density in the vicinity of the central point of the dam. Another probable reason is the non-compatibility of the Voronoi subdivision of the domain (which is a polygonal subdivision) with the circular geometry and the circular nature of the flow.

10.4. Rectangular channel with a slight contraction

In the second step of the investigation of the sensitivity of the method to the geometry used, a slight contraction is introduced in the case of a rectangular channel dam break. Thus, two angular points are introduced into both sides of the channel. The inclination angle is 2.29° (10 m/100 m). The initial conditions are presented in Figure 16. The channel length is 200 m; its width is 80 m in its broad part and 60 m in its narrow part. The dam is located at $x = 85$ m; water depth is 1 m upstream and 0.95 m downstream. The CFL is taken as 0.3. The discontinuity in the water depth was linearized across a distance of 15 m.

Figure 17 gives a 3D view of the water depth obtained at $t = 5$ s. We can see that the sudden geometry change induces the birth of a wave perturbance, which is expected and is physically explained. In fact, when the shock (front) wave reaches the geometry deviation, it submits to a reflection, which is concretized by a sudden water-height elevation that vanishes as the wave continues to move downstream. We can, consequently, affirm that, the method overcomes problems

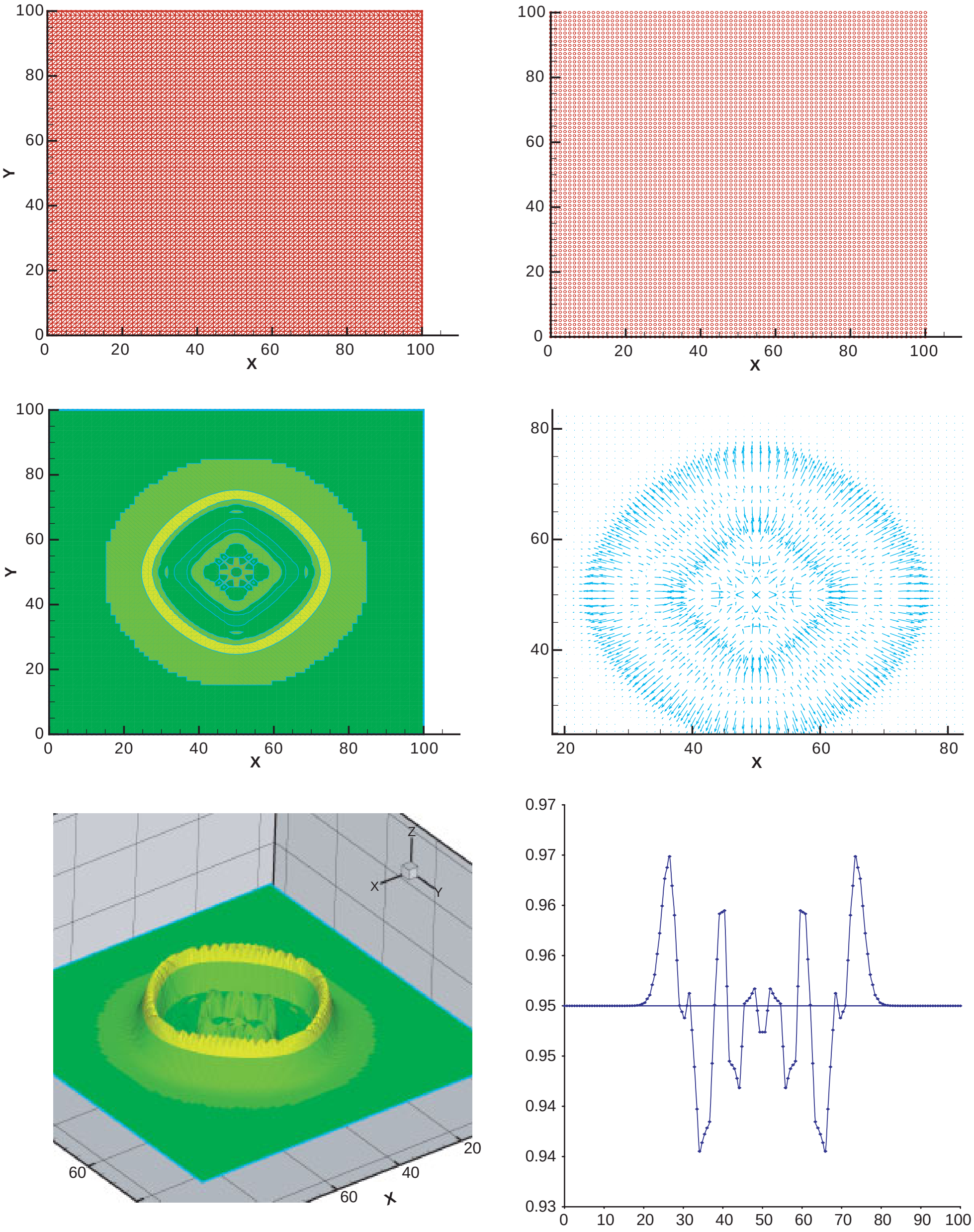

Figure 14. The cylindrical dam break problem—structured mesh. Nodal distribution and its related initial mesh (up right and left), the contour of the water depth and the velocity distribution (middle left and right), and eventually a 3D view of the water depth and a slice of the water depth (bottom left and right).

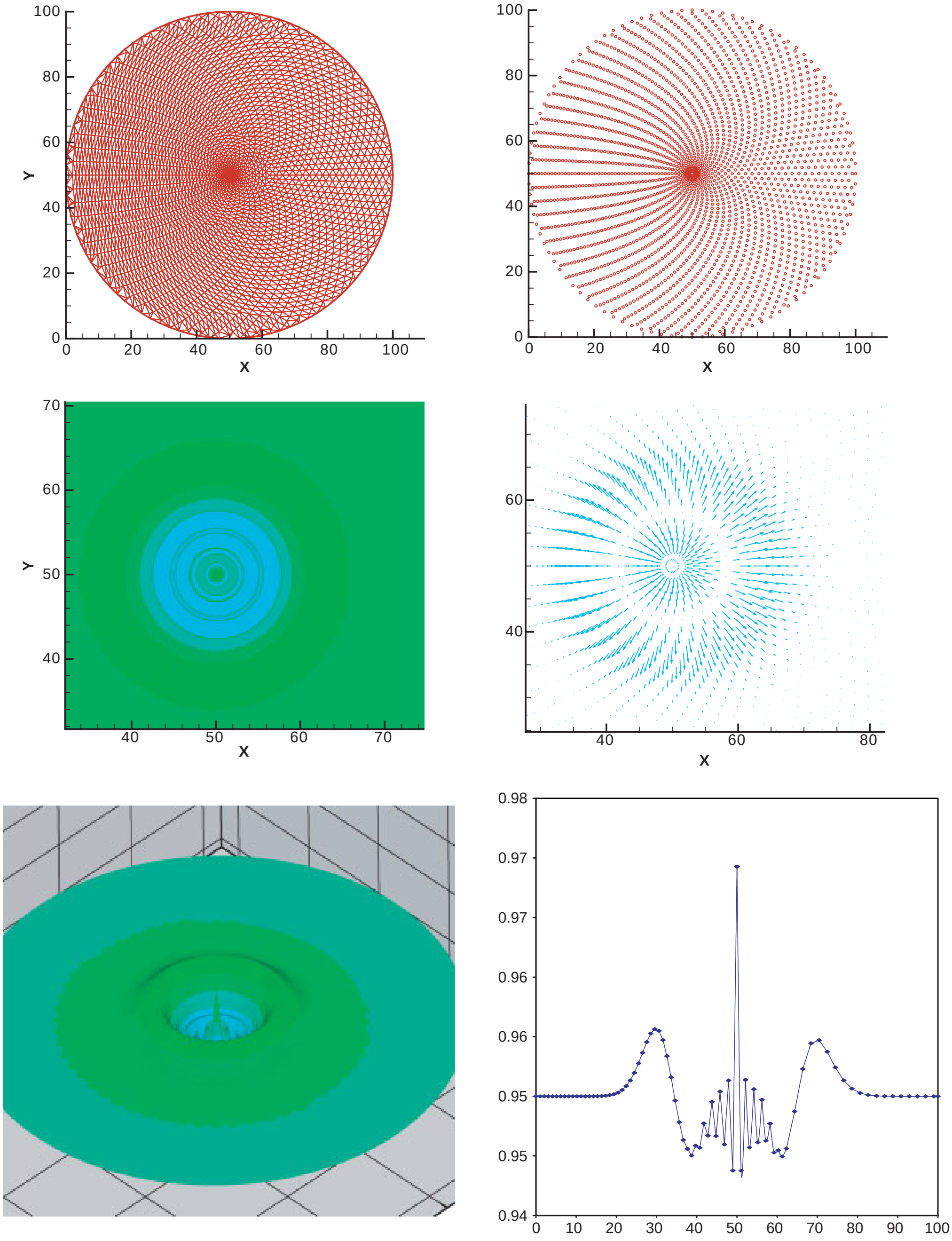

Figure 15. The cylindrical dam break problem—adapted mesh. Nodal distribution and its related initial mesh (up right and left), the contour of the water depth and the velocity distribution (middle left and right), and eventually a 3D view of the water depth and a slice of the water depth (left and right).

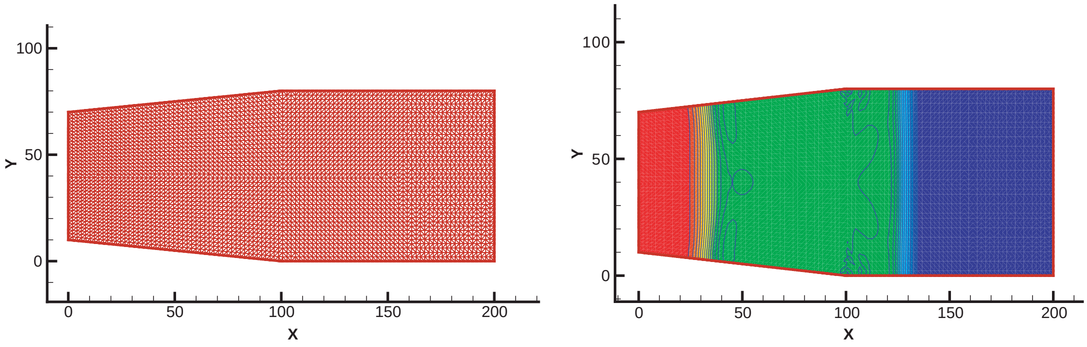

Figure 16. Geometry and contour lines for the rectangular channel with a slight contraction.

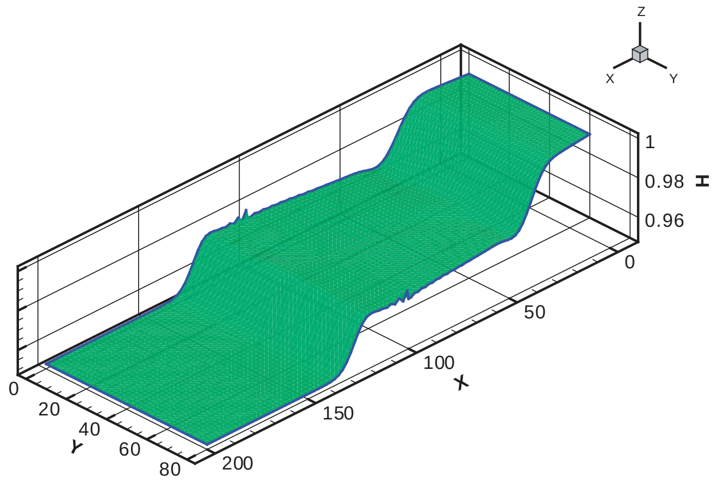

Figure 17. Water depth for the rectangular channel with a slight contraction at time $t = 5\,\mathrm{s}$. Note the birth of a water elevation when the front wave reaches the geometry irregularity.

generated by the presence of a slight geometrical deviation. More complicated geometrical applications will be covered in the future works.

10.5. Rectangular channel with a sudden contraction

The ultimate test is a rectangular channel with a sudden contraction. It is a test with a severe change in the geometry. This case will show the capacity of the scheme to handle wide variations in the geometry. The length of the domain is 200 m; its width is 80 m upstream the dam and 20 m downstream. The dam is located exactly at $x = 100\,\mathrm{m}$. The water depth is 1 m upstream and 0.95 m downstream. Figure 18 shows the initial conditions for this problem. The water discontinuity is linearized over a distance of 10 m. Figures 19 and 20 both give, respectively, the velocity, contour, and the water depth of the problem at $t = 5\,\mathrm{s}$. In this simulation 2490 particles were used. A CFL

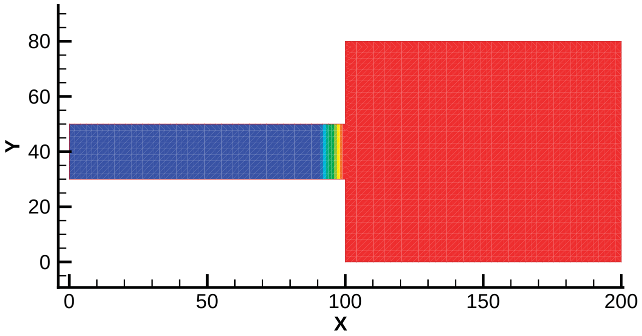

Figure 18. Initial conditions for the problem of rectangular channel with a sudden contraction.

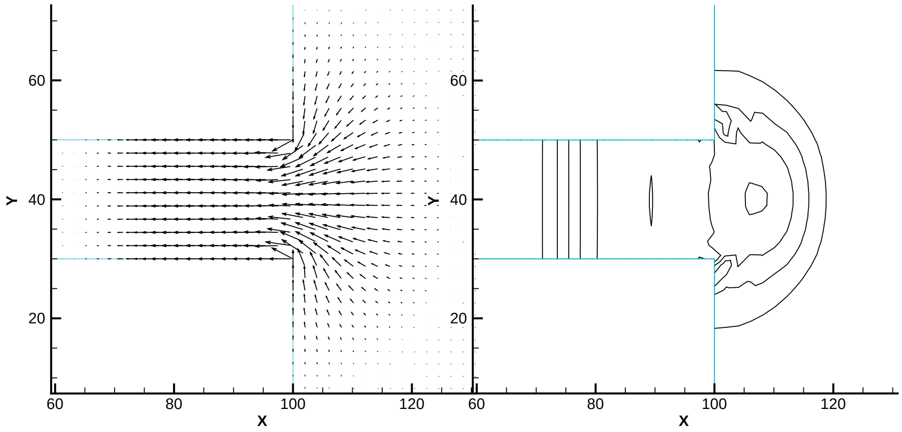

Figure 19. Rectangular channel with a sudden contraction: velocity distribution (left) and water depth contour (right).

of 0.5 was fixed for the time integration, a value that induces a time step in the neighborhood of 0.123 s in the beginning, which decreases down to 0.121 s at the end of the simulation.

We can conclude that the solution is generally well reproduced and that the waves are well captured. The shape and the smoothness of the solution is slightly perturbed in the vicinity of sharp angles of the geometry. This could be naturally explained by the fact that the linear interpolation discussed in Section 6 cannot be achieved for nodes located in the corner. In this case, the application of the continuity equation (12) is used to determine the water depth in this node. Few iterations are needed in order to reach the optimal value of h.

11. CONCLUSION

For the numerical approximation of the inviscid SWE, a new finite-volume method based on a flux computation using a natural neighbor approximation on the Voronoi diagram is applied.

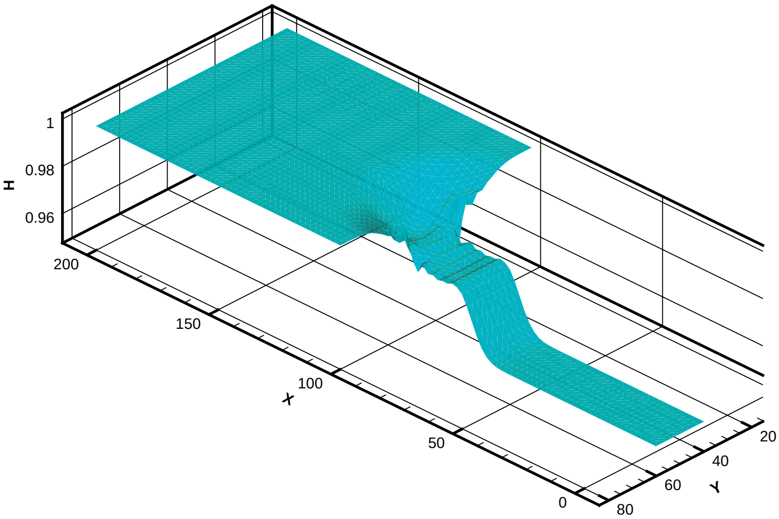

Figure 20. Water depth for the rectangular channel with a sudden contraction at time $t=5\,\mathrm{s}$.

Moreover, the application of this method is achieved within a purely Lagrangian collocation scheme. A perfect slipping condition is assumed at the boundary. The stabilization of the obtained scheme is achieved using an upwinding artificial viscosity that is obtained from an analogy with Riemann solver techniques. The first benchmark tests performed and shown in this paper reveal good results and hold out the promise for more interesting ones in the future. The presence of a complicated geometry and the introduction of source terms (bathymetry and bed friction) will be covered in the following steps before we go on to handle real cases of river or flood simulations.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the contribution of Dr L. A. Illoul for his invaluable help in the realization of this work. This work is partially funded by the CRSNG and Hydro-Quebec http://www.interscience.wiley.com/jpages/0271-2091/.

REFERENCES

1. Dorfi EA, Drury LO. Simple adaptive grids for 1-D initial value problems. *Journal of Computational Physics* 1987; **69**:175–195.
2. Miller K, Miller RN. Moving finite elements I. *SIAM Journal on Numerical Analysis* 1981; **18**:1019–1032.
3. Miller K. Moving finite elements II. *SIAM Journal on Numerical Analysis* 1981; **18**:1033–1057.
4. Idelsohn SR, Oñate E, Calvo N, Del Pin F. The meshless finite element method. *International Journal for Numerical Methods in Engineering* 2003; **58**(6):893–912.
5. Oñate E, Idelsohn SR, Del Pin F, Aubry R. The particle finite element method. An overview. *International Journal of Computational Methods* 2004; **1**(2):267–307.
6. Huang W, Russell RD. A moving collocation method for solving time dependent partial differential equations. *Applied Numerical Mathematics* 1996; **20**:101–116.
7. Monaghan JJ. Simulating free surface flows with SPH. *Journal of Computational Physics* 1994; **110**(2):399–406.

8. Bonet J, Kulasegaram S, Rodriguez-Paz MX, Profit M. Variational formulation for the smooth particle hydrodynamics (SPH) simulation of fluid and solid problems. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(12):1245–1257.
9. Lin P, Liu PF. A numerical study of breaking waves in the surf zone. *Journal of Fluid Mechanics* 1998; **59**:239–264.
10. Ata R, Soulaïmani A. A stabilized SPH method for inviscid shallow water flows. *International Journal for Numerical Methods in Fluids* 2005; **47**:139–159.
11. Cueto E, Sukumar N, Calvo B, Martinez MA, Cegoñino J, Doblaré M. Overview and recent advances in natural neighbor Galerkin methods. *Archives of Computational Methods in Engineering* 2003; **10**(4):307–384.
12. Cueto E, Cegoñino J, Calvo B, Doblaré M. On the imposition of essential boundary conditions in natural neighbor Galerkin methods. *Communications in Numerical Methods in Engineering* 2003; **19**(5):361–376.
13. Yvonnet J, Ryckelynck D, Lonrong P, Chinesta F. Interpolation naturelle sur les domaines non-convexes par l'utilisation du diagrame de Voronoi contraint. *Revue Europeenne des éléments Finis* 2003; **12**(4):487–509.
14. Sibson R. A brief description of natural neighbor interpolation. In *Interpreting Multivariate Data*, Barnett V (ed.). Wiley: Chichester, 1981.
15. Sukumar N, Moran B, Belytschko T. The natural elements method in solid mechanics. *International Journal for Numerical Methods in Engineering* 1998; **43**:839–887.
16. Missoum-Benziane D, Ryckelynck D, Chinesta F. A new fully coupled two-scales modelling for mechanical problems involving microstructure: the 95/5 technique. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:2325–2337.
17. Liang Q, Borthwick AG, Stelling G. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **46**:127–162.
18. Del Pin F. A Lagrangian approach based on the natural neighbor interpolation. *INRIA Report No. 4090*, Sophia Antipolis, France, 2000.
19. Sukumar N, Moran B, Yu Semenov A, Belikov VV. Natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:1–27.
20. Farin G. Surfaces over Dirichlet tesselations. *Computer Aided Geometric Design* 1990; **7**(1–4):281–292.
21. Belikov VV, Ivanov VD, Kontorovich VK, Korytnik SA, Semenov AY. The non-Sibsonian interpolation: a new method for interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics* 1997; **37**(1):9–15.
22. Yvonnet J, Ryckelynck D, Lorong P, Chinesta F. A new extension of the natural element method for nonconvex and discontinuous problems: the constrained natural element method (C-NEM). *International Journal for Numerical Methods in Engineering* 2004; **60**(8):1451–1474.
23. Christ NH, Friedberg R, Lee TD. Weights of links and plaquettes in a random lattice. *Nuclear Physics B* 1982; **210**(3):337–346.
24. González D, Cueto E, Martínez A, Doblaré M. Numerical integration in natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2004; **60**:2077–2104.
25. Bowyer A. Computing Dirichlet tesselations. *Computer Journal* 1981; **24**(2):162–166.
26. Watson D. Computing the *n*-dimensional Delaunay tesselation with application to Voronoi polytopes. *Computer Journal* 1981; **24**(2):167–172.
27. Sambridge M, Braun J, McQueen H. Geophysical parametrization and interpolation of irregular data using natural neighbours. *Geophysical International Journal* 1995; **122**:837–857.
28. Cueto E, Doblaré M, Gracia L. Imposing essential boundary conditions in the natural elements method by mean of density-scaled alpha-shapes. *International Journal for Numerical Methods in Engineering* 2000; **49**:519–546.
29. Klein R, Lingas A. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. *International Journal of Computational Geometry and Applications* 1996; **6**:263–278.
30. Martinez MA, Cueto E, Alfaro I, Doblaré M, Chinesta F. Updated Lagrangian free surface flow simulations with natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2004; **60**:2105–2129.
31. Soulaïmani A, Saad Y. An arbitrary Lagrangian–Eulerian finite element method for solving three dimensional free surface flows. *Computer Methods in Applied Mechanics and Engineering* 1998; **162**:79–106.
32. González D, Cueto E, Chinesta F, Doblaré M. A natural element updated Lagrangian strategy for free-surface fluid dynamics. *Journal of Computational Physics* 2007; **223**:127–150.
33. Birknes J, Pedersen G. A particle finite element method applied to long wave run-up. *International Journal for Numerical Methods in Fluids* 2006; **52**:237–261.

34. Sukumar N. Voronoi cell finite difference method for the diffusion operator on arbitrary unconstructed grids. *International Journal for Numerical Methods in Engineering* 2003; **57**(1):1–34.
35. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:435–466.
36. Mishev ID. Finite volume methods on Voronoi meshes. *Numerical Methods for Partial Differential Equations* 1998; **14**(2):193–212.
37. Bardella L, Genna F. Newmark's time integration method from the discretization of extended functionals. *Journal of Applied Mechanics* 2005; **72**(4):527–537.
38. Loukili Y, Soulaïmani A. Numerical tracking of shallow water waves by the unstructured finite volume WAF approximation. *International Journal for Computational Methods in Engineering Science and Mechanics* 2007; **8**(2):75–88.
39. Stoker JJ. *Water Waves*. Wiley-Interscience: New York, 1957.
40. Toro EF. *Shock-capturing Methods for Free Surface Flows*. Wiley: New York, 2001.