

Particle packing algorithm for SPH schemes

Andrea Colagrossi^{a,b,*}, B. Bouscasse^a, M. Antuono^a, S. Marrone^{a,c}

^a CNR-INSEAN, The Italian Ship Model Basin, Rome, Italy

^b Centre of Excellence for Ship and Ocean Structures, NTNU, Trondheim, Norway

^c Department of Mechanics and Aeronautics, University of Rome "Sapienza", Italy

ARTICLE INFO

Article history:

Received 26 August 2011

Received in revised form 7 February 2012

Accepted 27 February 2012

Available online 8 March 2012

Keywords:

Meshless methods

Smoothed particle hydrodynamics

Particle initialization

Lagrangian systems

ABSTRACT

Using some intrinsic features of the Smoothed Particle Hydrodynamics (SPH) schemes, an innovative algorithm for the initialization of the particle distribution has been defined. The proposed particle packing algorithm allows a drastic reduction of the numerical noise due to particle resettlement during the early stages of the flow evolution. Moreover, thanks to its structure, it can be easily derived starting from whatever SPH scheme and applies under the hypotheses that the fluid is weakly-compressible or incompressible as well. A broad range of numerical test cases proved this tool to be fast, robust and reliable also for complex geometrical configurations.

© 2012 Elsevier B.V. All rights reserved.

0. Introduction

In the Smoothed Particle Hydrodynamics (SPH) scheme the matter of how to initialize the particle positions plays a relevant role. If particles are not initially set in “equilibrium” positions, they may resettle giving rise to spurious motions which can strongly affect the fluid evolution.

Here, the acceptance of the word “equilibrium” deserves a clarification. We refer to an equilibrium configuration as the set of particle positions which, under static conditions, does not lead to particle resettlement. As proved in the following, the spurious particle motion is caused by inaccuracies in the SPH representation of the pressure gradient. Specifically, these inaccuracies largely increase when the particle distribution is anisotropic and disordered. At worst, the pressure gradient is unable to approximate the correct static conditions and non-physical currents/vorticity are generated. Then, the aim of the present work is to provide an algorithm which automatically gives the equilibrium configuration, that is, the specific particle distribution for which the pressure gradient is accurate and no particle resettlement occurs.

Apart from a few cases characterized by simple geometries, the equilibrium configuration is not known “a priori”. Further, the generation of spurious currents/vorticity may be particularly strong in the presence of complex solid boundary profiles (i.e. corners, bended bodies, etc.).

A possible solution is to start numerical simulations with a high numerical damping term and leave a long enough time to make particles self-resettle in equilibrium positions (see, for example, Monaghan [1]). Unfortunately, the attainment of a stable configuration can require a very long evolution, this leads to a large increase of computational costs. Moreover, the high damping used for particle initialization does not exclude that a further resettlement occurs when the actual simulation is started with a real viscosity term. This behavior has been observed in some numerical simulations of Section 3.

In the SPH framework, the first attempt to define a proper algorithm for particle initialization is due to Oger et al. [2] who adapted the Bubble method described in Shimada [3] to SPH solvers. This algorithm is based on the use of Van der Waals-like forces to place particles throughout the fluid domain. This method proves to be quite fast, applies to general geometries and provides a regular particle distribution. One of the weak points is that the particle positions obtained through the Bubble algorithm may not satisfy the static solution predicted by the SPH scheme, leading to a further resettlement.

Then, the key point to build a robust packing algorithm relies on the capability of providing a regular particle distribution which is also a static solution of the SPH scheme. To this purpose a novel packing algorithm has been derived taking advantage of some intrinsic features of the SPH schemes. Thanks to this, the proposed method allows the attainment of a regular particle distribution compatible with the static solution. Further, it can be easily derived starting from whatever SPH code and applies to weakly-compressible or incompressible SPH schemes as well.

The paper is organized as follows: Section 1 introduces the SPH scheme and gives an insight of the constitutive features which are

* Corresponding author at: CNR-INSEAN, The Italian Ship Model Basin, Rome, Italy.

E-mail address: a.colagrossi@insean.it (A. Colagrossi).

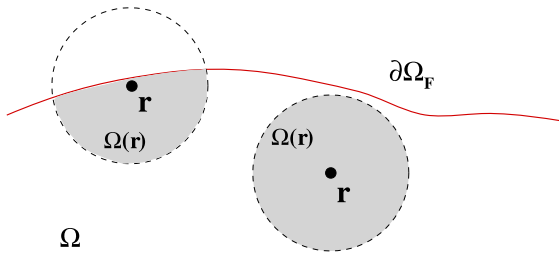


Fig. 1. Configurations of the kernel support $\Omega(\mathbf{r})$ with respect to the fluid domain boundary.

used to build the packing algorithm. Section 2 describes the proposed algorithm and highlights some interesting aspects about its Lagrangian structure. Finally, Section 3 provides a broad range of numerical test cases which prove the packing algorithm to be fast, robust and reliable also for complex geometrical configurations.

1. Governing equations

Two different approaches can be adopted in the SPH framework to model incompressible flows: the first relies on the direct use of the Navier–Stokes equations while the second is based on the assumption that the fluid is weakly-compressible and barotropic (that is, a one-to-one relation between pressure and density is assumed to hold true). Both approaches have benefits and drawbacks (for a detailed description we address the reader to Monaghan [4] and Shao [5]) and lead to the so-called incompressible SPH and weakly-compressible SPH schemes. The main advantage of the packing algorithm is that it applies to both schemes as well. In the present case, we adopt a weakly-compressible SPH scheme and, therefore, use the weakly-compressible Euler equations:

$$\begin{cases} \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0, \\ \frac{D\mathbf{u}}{Dt} = \mathbf{g} - \frac{\nabla p}{\rho}, \\ p = F(\rho), \\ \frac{D\mathbf{r}}{Dt} = \mathbf{u}, \end{cases} \quad (1.1)$$

where \mathbf{r} indicates a position inside the fluid domain Ω , \mathbf{g} is the gravity acceleration while \mathbf{u} , ρ and p are respectively the velocity, the density and the pressure fields. Function F represents the state equation linking the density field with the pressure one (we recall that the fluid is assumed to be barotropic). The weakly-compressibility assumption is enforced by choosing a sound velocity, $c_0 = \sqrt{dF/d\rho}$, which is much larger than the maximum expected velocity of the fluid (for more details we address the reader to the works of Madsen and Schäffer [6] and Monaghan [4]).

1.1. The SPH scheme

The SPH scheme is based on the filtering (*smoothing*) of any generic flow field f with a convolution integral over the fluid domain Ω

$$\langle f \rangle(\mathbf{r}) = \int_{\Omega} f(\mathbf{r}') W(\mathbf{r}' - \mathbf{r}; h) dV'. \quad (1.2)$$

$W(\mathbf{r}' - \mathbf{r}; h)$ is a weight function, which in practical applications must have a compact support $\Omega(\mathbf{r})$, and h (usually referred to as the *smoothing length*) is a characteristic length of such support (see Fig. 1). From a physical perspective, the smoothing length h represents the characteristic length of the domain of influence of the

fluid particle which is at the position indicated by \mathbf{r} . A very comprehensive review of the SPH framework can be found in Ref. [4].

The weight function $W(\mathbf{r}' - \mathbf{r}, h)$, called *smoothing function* or *kernel* in the SPH terminology, is positive, centered in \mathbf{r} and decreases monotonously with the distance $\|\mathbf{r}' - \mathbf{r}\|$. This monotonous decrease goes to zero at the border of its support $\Omega(\mathbf{r})$. The kernel considered in the present work is supposed to be isotropic in space, which is equivalent to being dependent only on the distance $s = \|\mathbf{r}' - \mathbf{r}\|$. The notation $W(\mathbf{r}' - \mathbf{r}; h)$ will be shortened hereinafter as $W(\mathbf{r}' - \mathbf{r})$ and the dependence on h will be implicitly assumed. In the limit as the smoothing length h goes to zero, the kernel has to converge weakly to a Dirac “function” and the original field of the convolution integral (1.2) has to be recovered. To this purpose, the kernel W must integrate to one (see e.g. [8]), that is:

$$\int_{\Omega} W(\mathbf{r}' - \mathbf{r}) dV' = 1 \quad \forall h > 0. \quad (1.3)$$

As extensively discussed in [8], such a property is not satisfied when the kernel domain is not completely immersed inside the fluid domain. This, for example, occurs in the neighborhood of the free surface $\partial\Omega_F$ (see Fig. 1) where the kernel domain is cut by the free surface. As a consequence, the lack of “mass” inside the kernel domain implies that the integral in (1.3) is smaller than one.

The filtering formula (1.2) can be applied to the gradient of a generic function

$$\langle \nabla f \rangle(\mathbf{r}) = \int_{\Omega} \nabla' f(\mathbf{r}') W(\mathbf{r}' - \mathbf{r}) dV' \quad (1.4)$$

with the prime on ∇ meaning that the derivatives are computed on the \mathbf{r}' variable. Eq. (1.4) can be further analyzed if it is integrated by parts:

$$\begin{aligned} \langle \nabla f \rangle(\mathbf{r}) &= \int_{\Omega} f(\mathbf{r}') \nabla W(\mathbf{r}' - \mathbf{r}) dV' \\ &+ \int_{\partial\Omega} f(\mathbf{r}') W(\mathbf{r}' - \mathbf{r}) \mathbf{n}' dS'. \end{aligned} \quad (1.5)$$

In this expression, ∇ indicates in turn the derivatives with respect to the variable \mathbf{r} and \mathbf{n}' is a unitary normal vector of $\partial\Omega$ pointing outward Ω . To obtain this equation, the antisymmetry property of the kernel gradient ($\nabla' W(\mathbf{r}' - \mathbf{r}) = -\nabla W(\mathbf{r}' - \mathbf{r})$) has been used.

With this reformulation of Eq. (1.4), the gradient of any generic function can be accessible from the knowledge of the function itself; this is the key point of the SPH method. When the smoothing procedure is applied to the differential operators of the governing equations (1.1), shortening the notation $\langle f \rangle(\mathbf{r})$ by $\langle f \rangle$, we get:

$$\begin{cases} \frac{D\rho}{Dt} + \rho \langle \nabla \cdot \mathbf{u} \rangle = 0, \\ \frac{D\mathbf{u}}{Dt} = \mathbf{g} - \frac{\langle \nabla p \rangle}{\rho}, \\ p = F(\rho), \\ \frac{D\mathbf{r}}{Dt} = \mathbf{u}. \end{cases} \quad (1.6)$$

The consistency of system (1.6) for the modeling of Euler equations in the presence of solid boundaries and free surfaces has been deeply investigated in Colagrossi et al. [8]. In practical SPH methods, the following formulas are generally used for the divergence of the velocity field and for the pressure gradient (see for example [9] and [8]):

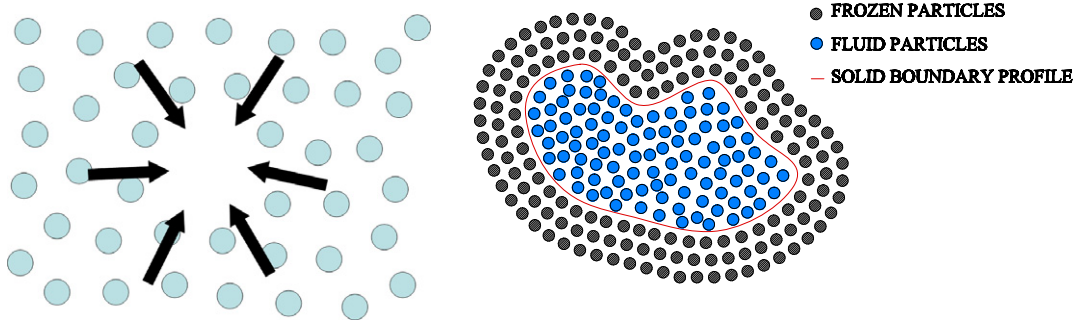


Fig. 2. Left panel: sketch of the vector $\mathbf{w} = -\nabla\Gamma$ in the neighborhood of a spatial anisotropy. Right panel: sketch of the solid and fluid particles in the packing algorithm framework.

$$\langle \nabla \cdot \mathbf{u} \rangle = \int_{\Omega} (\mathbf{u}' - \mathbf{u}) \cdot \nabla W(\mathbf{r}' - \mathbf{r}) dV', \quad (1.7)$$

$$\langle \nabla p \rangle = \int_{\Omega} (p' + p) \cdot \nabla W(\mathbf{r}' - \mathbf{r}) dV'. \quad (1.8)$$

When the SPH scheme is written at the discrete level, the fluid domain is represented through Lagrangian fluid particles carrying the main fluid properties (e.g. the velocity, pressure, density, etc.). In this framework, the integrals in (1.7) and (1.8) are replaced by summations over the fluid particles and the discrete SPH scheme reads:

$$\begin{cases} \frac{D\rho_i}{Dt} = -\rho_i \sum_j (\mathbf{u}_j - \mathbf{u}_i) \cdot \nabla_i W_{ij} V_j, \\ \frac{D\mathbf{u}_i}{Dt} = \mathbf{g} - \frac{1}{\rho_i} \sum_j (p_j + p_i) \nabla_i W_{ij} V_j + \mathbf{T}_i, \\ p_i = F(\rho_i), \\ \frac{D\mathbf{r}_i}{Dt} = \mathbf{u}_i. \end{cases} \quad (1.9)$$

Here, the subscripts indicate the quantities associated with the i -th and j -th fluid particle. In the specific, V_i is the particle volume, $\rho_i = m_i/V_i$, m_i is the particle mass and ∇_i indicates differentiation with respect to the position \mathbf{r}_i . The term \mathbf{T}_i indicates an artificial viscous force per unit of mass. This term is generally implemented in the SPH schemes for stability reasons (see, for example, [4]).

The main difference with the continuous framework is that the relation (1.3) is only approximately satisfied inside the fluid domain. This is due to local unevenness in the particle distribution. For the analysis which follows, it is convenient to introduce the following variables:

$$\Gamma_i = \sum_j W_{ij} V_j, \quad \nabla\Gamma_i = \sum_j \nabla_i W_{ij} V_j. \quad (1.10)$$

Variables Γ_i and $\nabla\Gamma_i$ give a “measure” of the unevenness in the particle distribution. In fact, if the particle distribution is perfectly uniform, $\Gamma_i = 1$ and $\nabla\Gamma_i = 0$ otherwise $\Gamma_i < 1$ and $\nabla\Gamma_i \neq 0$.

Further, the use of Γ_i and $\nabla\Gamma_i$ helps understand the convergence of the discrete differential operators. In fact, two different kinds of errors are made when the exact differential formulas are substituted with the discrete smoothed formulas. One kind is due to the interpolation procedure (errors proportional to the smoothing length, h) while the other is caused by the approximation of continuous integrals with finite summations (see, for example, [10]). In the latter case the error decreases as the number of particles inside the kernel domain increases. Then, if the mean number of particles in the kernel domain is large enough, this error can be

assumed to be smaller than $\mathcal{O}(h)$. Under this assumption and using the results obtained in Appendix A, the following expansions hold true:

$$\langle \nabla \cdot \mathbf{u} \rangle_i = \sum_j (\mathbf{u}_j - \mathbf{u}_i) \cdot \nabla_i W_{ij} V_j = \Gamma_i \langle \nabla \cdot \mathbf{u} \rangle_i + \mathcal{O}(h), \quad (1.11)$$

$$\begin{aligned} \langle \nabla p \rangle_i &= \sum_j (p_j + p_i) \nabla_i W_{ij} V_j \\ &= \Gamma_i \langle \nabla p \rangle_i + 2p_i \nabla\Gamma_i + \mathcal{O}(h). \end{aligned} \quad (1.12)$$

Here, both Γ_i and $\nabla\Gamma_i$ are responsible for a deviation from the exact differential operators. As shown in the following example, the greatest issue is caused by the pressure gradient.

Let us assume we want to start a SPH simulation with hydrostatic conditions. Then, we assign $\mathbf{u}_i = 0$, $\rho_i = \rho(\mathbf{r}_i)$ at $t = t_0$ and try to find the hydrostatic pressure. The continuity equation is satisfied exactly while, as a consequence of (1.12), the momentum equation gives:

$$\Gamma_i \langle \nabla p \rangle_i + 2p_i \nabla\Gamma_i - \rho_i \nabla\Phi = \mathcal{O}(h), \quad (1.13)$$

where Φ is the gravitational potential. This expression clearly shows that, unless $\Gamma_i = 1$ and $\nabla\Gamma_i = 0$, it is not possible to attain any hydrostatic solution. In this context, Γ_i plays only a minor role since it just causes an increase/decrease of the intensity of the correct pressure gradient. On the contrary, $\nabla\Gamma_i$ is responsible for an unbalance in both the intensity and the direction of the SPH differential operator. Further, $\nabla\Gamma_i$ diverges like $1/h$ when the particle distribution is strongly irregular. Then, the only way to get a good initialization of the SPH scheme is to reduce the magnitude of $\nabla\Gamma_i$ as much as possible and recover the consistency of the SPH pressure operator. This is the principal idea at the basis of the Particle Packing Algorithm described in the next section.

2. Particle Packing Algorithm

The Particle Packing Algorithm is built on a simple idea: to use the SPH features highlighted in the previous section to initialize the particle distribution and minimize $\|\nabla\Gamma\|$. This is made by observing that the vector $\mathbf{w} = -\nabla\Gamma$ always points in the direction of the maximum lack of “mass” and maximum anisotropy (see Fig. 2). Now, let's assume to use it to move particles during the initialization. If the fluid domain is bounded and particles are not allowed to escape from the boundaries, \mathbf{w} tends to fill all the asymmetries in the particle distribution and, at the same time, decreases as a consequence of the more regular distribution of particles themselves. Then, the final distribution would be the most regular possible and $\|\mathbf{w}\|$ (that is, $\|\nabla\Gamma\|$) would be minimized as requested.

The first step to build the Particle Packing Algorithm is to close the domain boundaries. As a consequence, this implies that the

free surface has to be treated as a solid boundary. The domain boundary is modeled through fixed solid particles, that is, particles with zero velocity and fixed positions. This approach can be regarded as a special use of the frozen particles (for details see [11]) or as a straightforward application of the fixed ghost particle technique proposed by Marrone et al. [7]. A sketch of this procedure is displayed in the right panel of Fig. 2. In principle, the proposed packing algorithm may be modified to include different techniques for the solid profile modeling (e.g., repellent particles [12] or boundary surface forces [13]). However, the use of fixed solid particles is straightforward and leads to important theoretical results (see Section 2.1). Note that particles do not need any specific rule to be positioned inside the fluid domain nor inside the solid bodies.

The second step consists in assuming the density, the pressure and the volumes constant all over the fluid domain. We indicate them through symbols ρ_0, p_0 and V_0 respectively. Since volumes are constant and the packing algorithm has to converge toward a static solution, we neglect the continuity equation. Conversely, the momentum equation of system (1.9) becomes:

$$\begin{cases} \frac{D\mathbf{u}_i}{Dt} = -\beta \nabla \Gamma_i + \mathbf{T}_i^{(\zeta)}, \\ \frac{D\mathbf{r}_i}{Dt} = \mathbf{u}_i \end{cases} \quad (2.14)$$

where $\beta = 2p_0/\rho_0$ and $\Gamma_i = \sum_j W_{ij}V_0$. The damping force can be chosen independently from the adopted SPH scheme since $\mathbf{T}_i^{(\zeta)}$ is just used to ensure the convergence of the Particle Packing Algorithm (more details are given in Section 2.1 and in Appendix B). Similarly to Monaghan [1], we choose a linear damping term:

$$\mathbf{T}_i^{(\zeta)} = -\zeta \mathbf{u}_i \quad \text{with } \zeta = \alpha \frac{\sqrt{\beta}}{V_0^{1/d}} \quad (2.15)$$

where d is the spatial dimension and α is a free dimensionless parameter. By numerical simulations, we heuristically found that a good choice for α ranges between $1 \cdot 10^{-3}$ and $5 \cdot 10^{-3}$. Then, the particle packing system becomes:

$$\begin{cases} \frac{D\mathbf{u}_i}{Dt} = -\beta \nabla \Gamma_i - \zeta \mathbf{u}_i, \\ \frac{D\mathbf{r}_i}{Dt} = \mathbf{u}_i. \end{cases} \quad (2.16)$$

The initial conditions for the Particle Packing Algorithm are obtained by setting all the particle velocities to zero and $V_0 = V_{tot}/N_{part}$ where V_{tot} is the total fluid volume and N_{part} is the total number of particles. The time-step adopted for the present algorithm is:

$$\Delta t = CFL \frac{V_0^{1/d}}{\sqrt{\beta}}, \quad (2.17)$$

where $CFL = 1$.

As discussed in detail in the following sections, system (2.16) tends to converge as much as possible toward a steady state characterized by $\mathbf{u}_i = 0$ and $\nabla \Gamma_i = 0$. When the fluid system is sufficiently close to this state, the packing algorithm is stopped and the particle positions are used to initialize the SPH simulations. Since the spatial distribution is very regular, the particle volumes can be assumed to be identical. Then, the volume used for the initialization of the SPH is $V^* = V_0$. The initial particle pressure, p_i^* , is assigned by using the analytical expression for the hydrostatic pressure and the particle positions. Then, inverting the state equation, the initial density ρ_i^* is computed and, finally, the particle mass is obtained through $m_i^* = \rho_i^* V^*$. During the SPH simulations,

the particle masses are kept constant while the densities and the volumes are updated using the continuity equation and the relation $V_i = m_i/\rho_i$.

2.1. The particle packing system as a Lagrangian system

Let us consider (2.14) and neglect the damping force $\mathbf{T}_i^{(\zeta)}$. Under this hypothesis, the particle packing system is a Lagrangian system and its Lagrangian function is $\mathcal{L} = \mathcal{T} - \mathcal{V}$ where:

$$\mathcal{T} = \sum_i^* \frac{\|\mathbf{u}_i\|^2}{2}, \quad \mathcal{V} = \frac{\beta}{2} \sum_i^* \sum_j^* W_{ij}V_0 + \beta \sum_i^* \sum_j^* \overline{W}_{ij}V_0. \quad (2.18)$$

Here the starred series indicates the summation over the fluid particles while the barred one denotes the summation over the fixed solid particles. Symbols \mathcal{T} and \mathcal{V} are the specific kinetic energy and specific potential energy respectively. According to the theory of dynamical systems, the total energy (that is, $\mathcal{E} = \mathcal{T} + \mathcal{V}$) keeps constant during the evolution and the system has stable equilibria at the points where the potential attains its local minima. The addition of any dissipative term (like, for example, that in (2.15)) forces the dynamical system to converge toward the static (and stable) solution $\mathbf{u}_i = 0, \nabla \Gamma_i = 0$. This is a fundamental point since it proves that for a given arbitrary geometry (modeled through fixed ghost particles), the packing algorithm always converges toward a stable particle configuration. All details on the Lagrangian structure of the packing algorithm are given in Appendix B.

2.2. Stability and equilibria in \mathbb{R}^2

Let us consider the system (2.16) in the two-dimensional plane, \mathbb{R}^2 . In this case, there are no boundaries and, therefore, for $\zeta \neq 0$, the system has stable equilibria at the points where the potential attains its local minima.

If the number of particles is finite, their spatial distribution cannot be homogeneous nor isotropic and, therefore, $\nabla \Gamma \neq 1$ somewhere in \mathbb{R}^2 . Since $\mathbf{w} = -\nabla \Gamma$ points outward the fluid domain in the direction of maximum anisotropy, particles start going away one from the other and their reciprocal distance grows. This implies a decrease of $W(\|\mathbf{r}_j - \mathbf{r}_i\|)$ and, consequently, a decrease of the potential energy. The final configuration predicts particles escaping at infinity and corresponds to zero potential energy (that is, a local and absolute minimum for the potential energy).

If the particle number is infinite, there exist only three regular tessellations of \mathbb{R}^2 which ensure $\nabla \Gamma = 0$. These configurations (which corresponds to local extrema for the potential energy and, therefore, to equilibria of the particle system) are the triangular, square and hexagonal tessellation of the plane. Note that they are all invariant for rigid rotations and translations which, therefore, represent “directions” of neutral stability.

From a practical and theoretical point of view, it is interesting to study the stability of such equilibria. Since the theoretical analysis is very prohibitive, this is made through numerical simulations. The first step consists in the construction of a bounded fluid domain Ω using one of the tessellations mentioned above. Such a domain has to be large enough to assume that the most inner particles are not influenced by $\partial \Omega$. The boundary is fixed and modeled through fixed solid particles. Note that the presence of a fixed boundary automatically eliminates the “directions” of neutral stability (that is, rigid rotations and translations). After the domain Ω and its boundary have been built, particles are perturbed from their equilibrium positions. Specifically, we put $\zeta = 0$, since we do not want the viscosity to influence the stability of the tessellations under consideration. Under this hypothesis, sys-

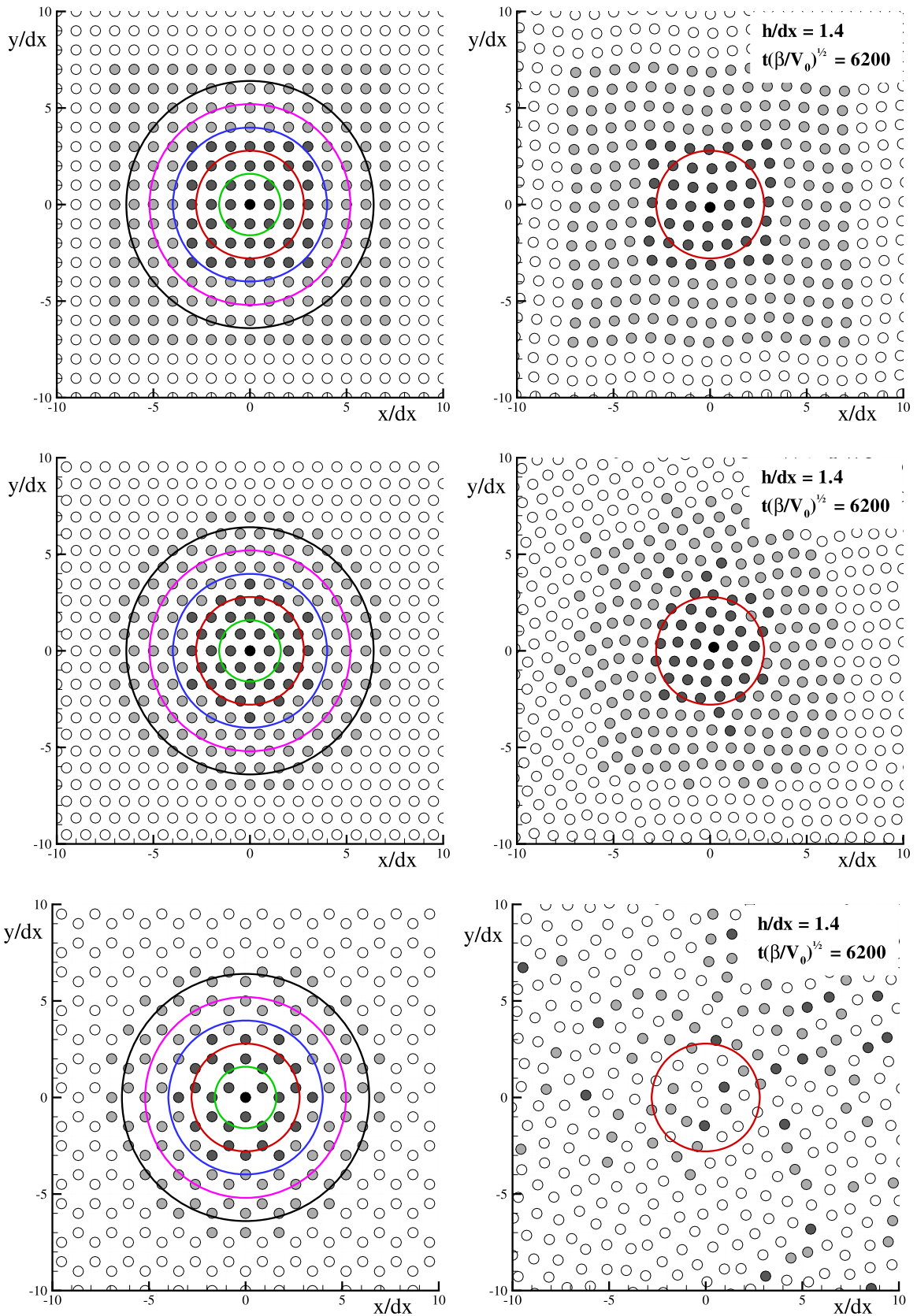


Fig. 3. Cartesian, triangular and hexagonal tessellations (left column, from top to bottom) and some snapshots of their evolution (right column, from top to bottom) for $h/dx = 1.4$. The circumferences indicate the domains of the adopted kernel functions.

tem (2.16) is Lagrangian (that is, it preserves the total energy, \mathcal{E}) and, therefore, only two cases are possible: particles keep oscillating around the equilibrium configuration or they move toward

other configurations. In the former case the equilibrium (and the related tessellation) is said to be Liapunov stable (see, for example, [14]) otherwise the equilibrium is unstable.

Table 1
Stability of the tessellations. ‘S’ = Liapunov stable, ‘U’ = unstable.

	$h/dx = 0.8$	$h/dx = 1.4$	$h/dx = 2.0$	$h/dx = 2.6$	$h/dx = 3.2$
Cartesian	U	S	S	U	U
Triangular	S	U	S	S	U
Hexagonal	U	U	U	U	U

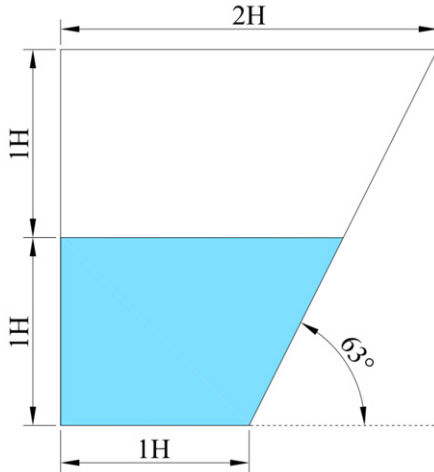


Fig. 4. Sketch of the trapezoidal tank.

Fig. 3 displays the tessellations (left column) along with different supports of the kernel functions (circumferences). In all the simulations we used a Wendland kernel function with $h/dx = 0.8, 1.4, 2.0, 2.6, 3.2$ and the particle positions have been perturbed from their stable configuration using a noise of order $dx/100$. In the right column, some snapshots of the evolutions of the different tessellations have been drawn for $h/dx = 1.4$. In this specific case, the Cartesian tessellation is Liapunov stable while the triangular and the hexagonal ones are unstable. **Table 1** briefly summarizes the results obtained. We found that the hexagonal tessellation is always unstable while the stability of the Cartesian and the triangular tessellations cannot be established “a priori”. Note that behavior of a tessellation generally depends on the specific kernel used.

3. Applications

In the present section we show some applications of the particle packing algorithm. We first deal with the initialization of hydrostatic conditions in complex geometrical configurations, then we show a dynamical problem which evolves after the particle packing initialization. In all the simulations, the standard SPH scheme (see [4] for more details) has been implemented using a Wendland kernel. Further, a linear state equation has been adopted (see, for example, [15]) and the solid profiles have been modeled through the fixed ghost particles described in Marrone et al. [7]. Incidentally, we highlight that the qualitative results obtained in the following sections also hold true for those SPH schemes that implement frozen particles.

3.1. Trapezoidal tank

We first consider a trapezoidal tank like that drawn in **Fig. 4** (H is the filling height) and study the influence of the particle initialization on the capability of the SPH of simulating the hydrostatic solution. As stated in Section 2, the first step is to “close” the fluid domain. This means that the free surface has to be substituted by a solid boundary and modeled accordingly. This procedure

is displayed in **Fig. 5** where the fluid domain has been initialized through a Cartesian grid (left panel) and using the packing algorithm (right panel).

In the former case, the use of a Cartesian grid leads to the generation of large spatial anisotropies along the inclined plane. Here, $\|\nabla\Gamma\| = O(1)$ and, therefore, an intense particle resettlement is expected during the early stages of the fluid evolution. Conversely, the particle packing algorithm eliminates the spatial anisotropies and drastically reduces the magnitude of $\|\nabla\Gamma\|$ (whose order of magnitude is about $10^{-13}H$). Incidentally, we note that it is possible to derive SPH schemes which may reduce the spatial anisotropies close to the solid profiles through the use of special boundary conditions (see, for example, [16]). However, these schemes are generally more complex than the standard SPH model and leads to higher computational costs.

It is also interesting to analyze the dependence of the proposed algorithm on the spatial resolution. In **Fig. 6** the specific kinetic energy of the packing scheme (that is, \mathcal{E}) is displayed for three different spatial resolutions versus the number of iterations. The overall behavior during the initial stages is similar for all the cases and the specific kinetic energy rapidly decreases to $10^{-7}\beta$. This heuristically shows that the Particle Packing Algorithm is weakly influenced by the adopted spatial resolution. Then, a further decrease occurs for longer time depending on the specific resolution. In any case, particles are practically motionless after 2500 iterations when the order of magnitude of the specific kinetic energy is less than $10^{-8}\beta$. This means that the equilibrium configuration has been attained and that the packing algorithm can be stopped. Obviously, the number of iterations required for the attainment of equilibria may vary according to the specific problem at hand, to the choice of the kernel function (i.e., Gaussian, cubic spline, quintic spline, etc.) and to the adopted damping but is generally of order 10^3 .

When the particle initialization is complete, the hydrostatic solution is assigned to the fluid domain (that is, hydrostatic pressure field and zero initial velocity) and we start the simulation through the standard SPH scheme [4].

As shown in the left panel of **Fig. 7**, the initialization through the Cartesian grid, because of the high values of $\|\nabla\Gamma\|$ near the sloping plane, leads to the generation of high spurious currents ($\max_i |\mathbf{u}_i| \simeq 0.1\sqrt{gH}$ at $t = 10\sqrt{H/g}$). On the contrary, these are completely absent when the simulations are initialized through the particle packing algorithm (right panel of the same figure).

A global measure of the particle resettlement phenomenon is easily obtained by inspecting the evolution of the maximum intensity of velocity (that is, $\max_i |\mathbf{u}_i|$) during the simulation of the hydrostatic solution. As shown in **Fig. 8**, the maximum intensity of velocity after the use of packing algorithm is at least two orders of magnitude smaller than that predicted by the simulation which starts on a Cartesian grid. In the latter case the particle motion still persists at $t = 100\sqrt{H/g}$ while the simulation after the packing algorithm is practically motionless (see right panel of **Fig. 7**).

3.1.1. Particle Packing Algorithm versus initialization through SPH with linear damping term

As already mentioned in the Introduction, an alternative solution to reduce particle resettlement is to start the SPH simulation using a high numerical damping term and leave a long enough

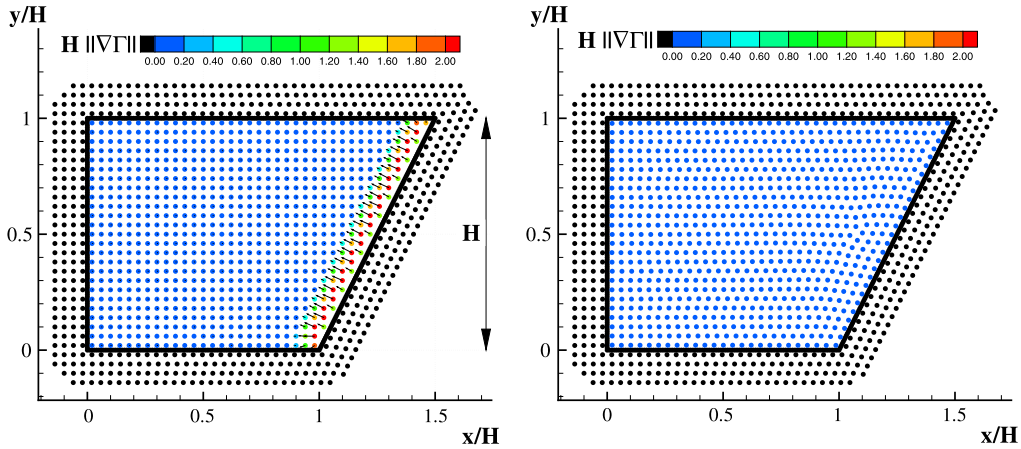


Fig. 5. Trapezoidal tank ($H/dx = 25$). Initialization using a Cartesian grid (left) and through the particle packing algorithm (right).

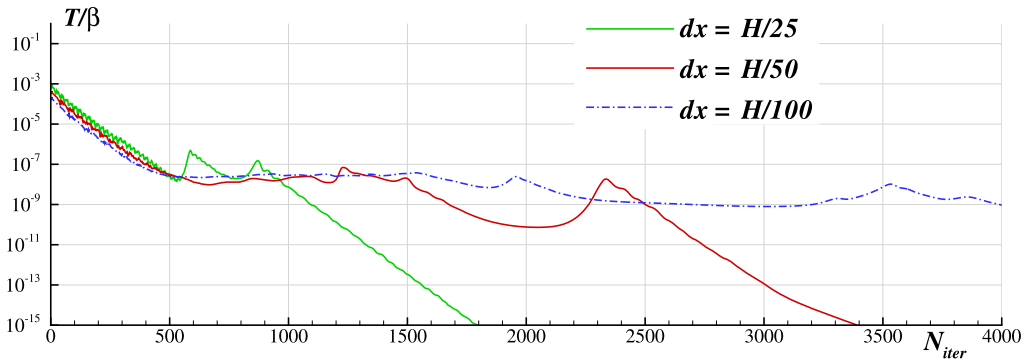


Fig. 6. Trapezoidal tank. Evolution of the specific kinetic energy during the initialization through the particle packing algorithm.

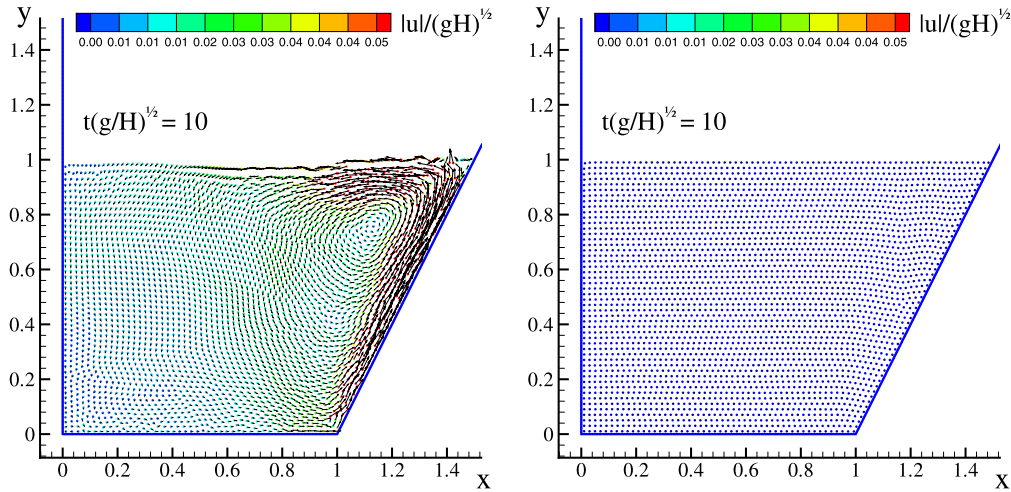


Fig. 7. Hydrostatic solution for the trapezoidal tank ($H/dx = 50$). Evolution using a Cartesian grid (left) and after the initialization through the particle packing algorithm (right).

time to make particle self-resettle in equilibrium positions. The actual numerical simulation starts after the equilibrium configuration is attained. Here, we show that such a procedure (that is, the initialization using the SPH scheme itself) only leads to minor improvements.

Following Monaghan [1], we use the standard SPH scheme with a linear damping term. By definition, this is identical to the damping term adopted in the packing algorithm and, consequently, its coefficient has been denoted by ζ . Apart from these similari-

ties, the physical meaning of the damping coefficient used in the SPH simulations is slightly different from that of the ζ coefficient adopted in the particle packing algorithm. In the latter case, the damping term has to be regarded as an “inter-particle” dissipation while in the former case it represents artificial dissipations. For this reason, the ζ coefficient of the SPH simulations has been made dimensionless using physical variables (specifically, the sound velocity c_0 and the filling height H).

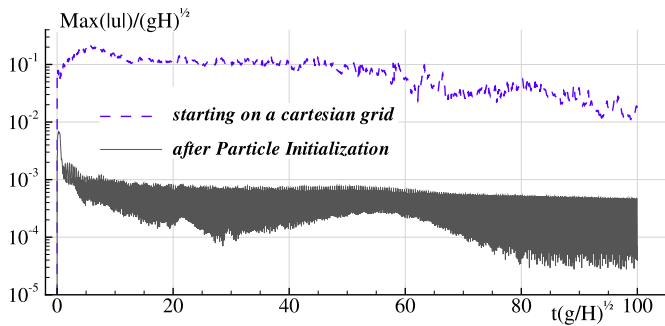


Fig. 8. Hydrostatic solution for the trapezoidal tank ($H/dx = 50$): time evolution of the maximum intensity of velocity.

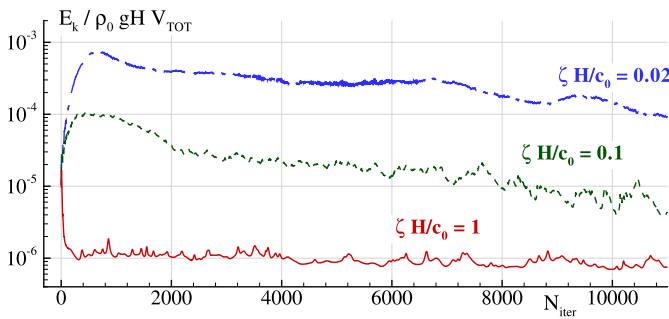


Fig. 9. Kinetic energy evolution during the initialization with the standard SPH scheme for different values of the damping coefficient ζ .

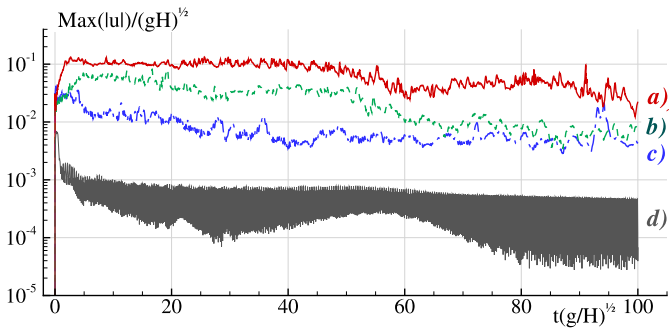


Fig. 10. Evolution of the maximum intensity of velocity after the particle initialization with the standard SPH scheme (case a: $\zeta = 1.0c_0/H$; case b: $\zeta = 0.1c_0/H$; case c: $\zeta = 0.02c_0/H$) and with the particle packing algorithm (case d).

The evolution of the kinetic energy during the particle initialization through the SPH scheme is displayed in Fig. 9 for different values of ζ . Incidentally we highlight that it has not been possible to use values of ζ smaller than 0.02 since the SPH scheme was unstable. The attainment of the equilibrium becomes faster and faster as the magnitude of the damping coefficient increases. However, this behavior does not correspond to a reduction of particle resettlement when the actual simulation is started but, on the contrary, leads to the generation of larger spurious currents which persist for very long times. This is briefly summarized in Fig. 10 where the evolution of the maximum intensity of velocity is displayed for the same cases of Fig. 9. It is also interesting to note that the use of values of ζ smaller than 0.1 does not lead to any significant improvement for long time evolution. Note that particles are still moving at $t = 100\sqrt{H/g}$ since $\max_i |\mathbf{u}_i| \simeq 0.01\sqrt{gH}$. Conversely, the initialization through the packing algorithm lead to the correct equilibrium configuration and avoids any further resettlement (see case d of Fig. 10).

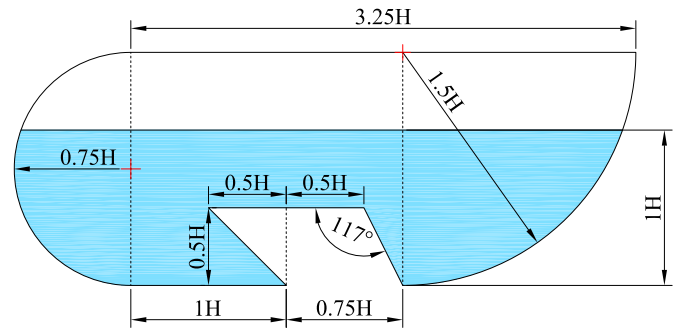


Fig. 11. Sketch of the complex tank geometry.

3.2. A complex tank geometry

As a second example, we consider a complex geometry characterized by bended profiles with different curvatures and by acute and obtuse solid angles (see Fig. 11). Because of these features, the particle initialization of such a geometry represents a very difficult problem.

The top panel of Fig. 12 displays the fluid evolution under hydrostatic conditions after the initialization on a Cartesian grid. In this case, the generation of spurious currents and vorticity near corners and bended profiles is very strong and persists for long times. On the contrary, the use of the particle packing algorithm eliminates such an undesirable behavior and gives a uniform particle distribution which keeps stable for long times (middle panel of Fig. 12). The evolution of the kinetic energy confirms the findings above proving that, after the use of the proposed algorithm, particles are almost motionless (bottom panel of Fig. 12).

3.3. A freely floating problem

Here we consider a ship hull section floating in hydrostatic conditions. Under such a hypothesis, the hull section should remain motionless. However, because of the particle resettlement, an unphysical deviation of the ship hull from the initial position may be observed.

Similarly to the test cases studied in the previous sections, we initialize the fluid domain using a Cartesian grid (top left panel of Fig. 13) and the packing algorithm (top right panel of the same figure). The bottom panels display the related SPH simulations. In this case, the spurious currents that generate in the neighborhood of the hull because of the Cartesian grid force the ship hull to move. Fig. 14 shows the motion of the mass center (top panels) and the roll motion (bottom panel) of the ship hull. Because of the reduction of the spurious currents, the packing algorithm drastically reduces the unphysical ship motion ensuring the attainment of the correct hydrostatic solution.

3.4. Evolution of an elliptical drop

Let us consider a fluid domain Ω which at the initial time is a two-dimensional ball of radius R , subjected to the velocity field:

$$\begin{cases} u_0(x, y) = A_0 x, \\ v_0(x, y) = -A_0 y \end{cases} \Rightarrow \nabla \mathbf{u}_0 = \begin{pmatrix} A_0 & 0 \\ 0 & -A_0 \end{pmatrix}. \quad (3.19)$$

The initial pressure field is derived using the Poisson equation [17] and reads:

$$p_0(x, y) = \frac{\rho_0 A_0^2}{2} [R^2 - (x^2 + y^2)]. \quad (3.20)$$

Assuming the flow to be inviscid, Ω preserves an elliptical form during the evolution and this form can be derived analytically

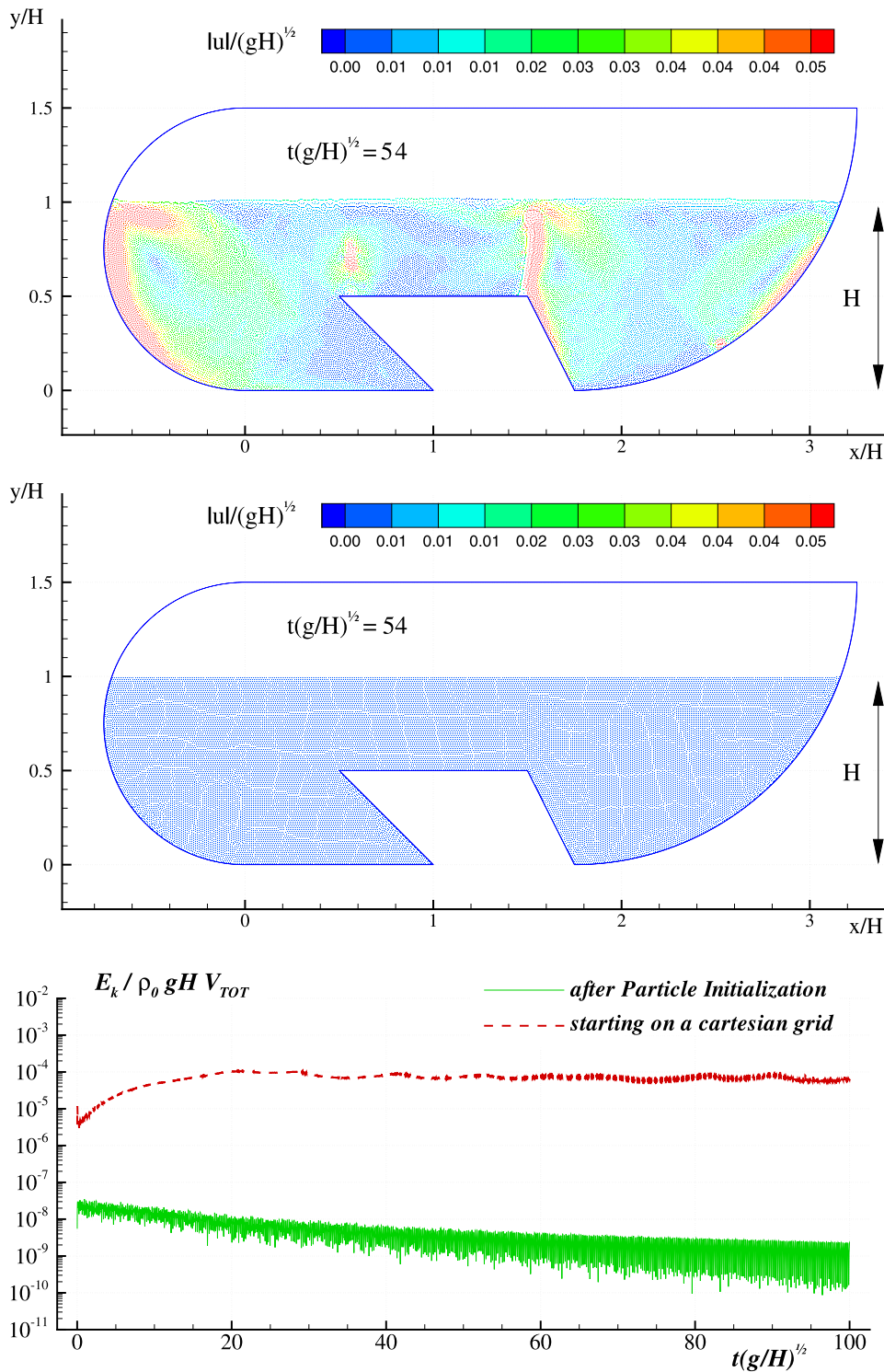


Fig. 12. Hydrostatic solution for a complex tank geometry. Top: evolution after initialization on a Cartesian grid. Middle: evolution after initialization through packing algorithm. Bottom: time history of the kinetic energy.

(see [1] and [17] for details). This domain is initialized using a Cartesian grid (left panel of Fig. 15) and the proposed particle packing algorithm (right panel of the same figure). Since the SPH is a Lagrangian scheme, particles move along stream lines. Consequently, when particles are initially set on a Cartesian grid, the flow evolution given by (3.19) leads particles to clump along straight lines (see top panel of Fig. 16). This partially prevents the SPH solution to match with the analytical solution for the domain boundary (dashed lines in Fig. 16). On the contrary, the

flow evolution after the use of the packing algorithm displays a more uniform particle distribution and, consequently, leads to a better agreement with the analytical solution (bottom panel of Fig. 16).

4. Conclusions

Using some intrinsic features of the SPH scheme, a novel packing algorithm has been derived for the particle initialization. The

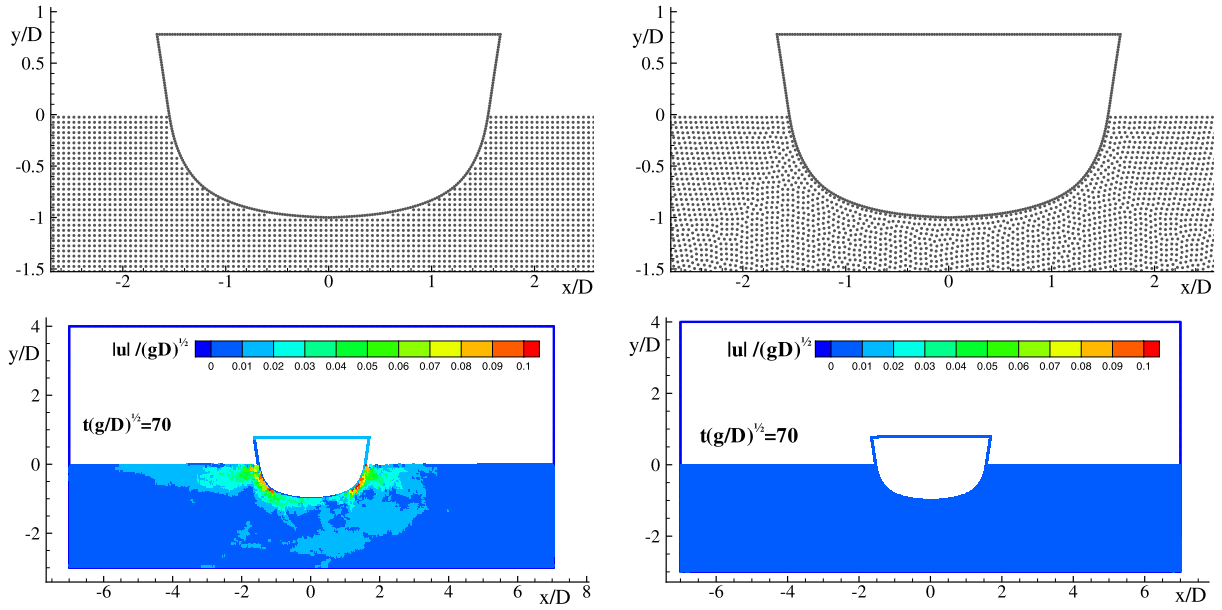


Fig. 13. Freely floating of a ship hull section after initialization on a Cartesian grid (left column) and with the particle packing algorithm (right column).

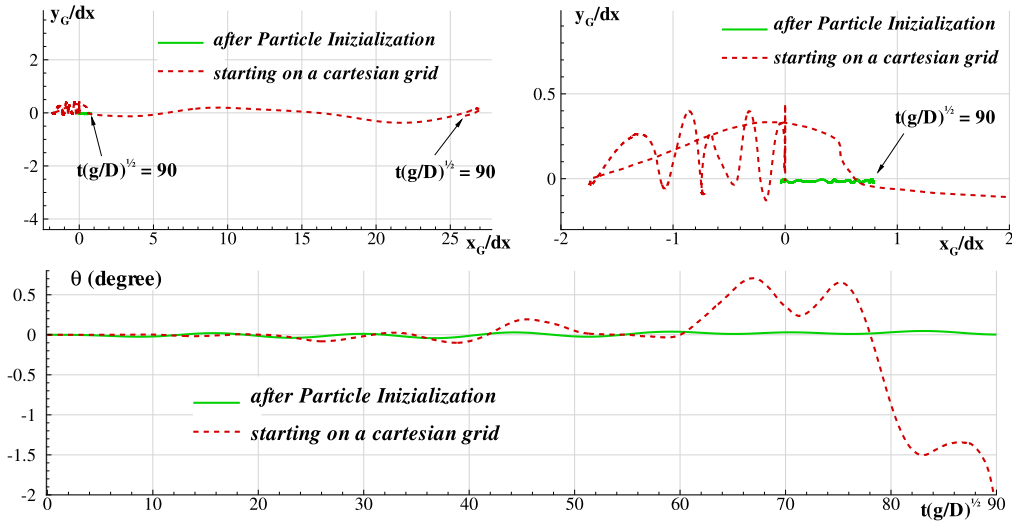


Fig. 14. Freely floating of a ship hull. Top panels: motion of the mass center (left panel) and a detail of its evolution (right panel). Bottom panel: roll motion.

proposed algorithm has been validated against several test cases proving to be robust, fast and reliable also for complex geometrical configurations. As shown for the evolution of the elliptical drop, the particle distribution obtained through the packing algorithm may even avoid the formation of those filamentous structures that are caused by the Lagrangian nature of the SPH.

Acknowledgements

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007–2013) under grant agreement No. 225967 “NextMuSE”. This work was also partially supported by the Centre of Excellence for Ship and Ocean Structures of NTNU Trondheim (Norway) within the “Violent Water–Vessel Interactions and Related Structural Loads”.

Appendix A. Interpolation formulas

Let us consider a generic scalar function f and the following convolution integral:

$$\langle \nabla f \rangle = \int_{\Omega} \nabla' f' W dV' \tag{A.21}$$

where, for the ease of notation, $f' = f(\mathbf{r}')$ and $W = W(\mathbf{r}' - \mathbf{r})$. Integrating by parts and using the divergence theorem, we get:

$$\begin{aligned} \langle \nabla f \rangle &= \int_{\Omega} \nabla' (f' W) dV' - \int_{\Omega} f' \nabla' W dV' \\ &= \int_{\partial\Omega} f' W \mathbf{n}' dV' + \int_{\Omega} f' \nabla W dV', \end{aligned}$$

where \mathbf{n} is the normal to $\partial\Omega$. Note that the last integral has been obtained using the identity $\nabla' W(\mathbf{r}' - \mathbf{r}) = -\nabla W(\mathbf{r}' - \mathbf{r})$. Now, consider the following expansion:

$$f' = f + \nabla f \cdot (\mathbf{r}' - \mathbf{r}) + \mathcal{O}(\|\mathbf{r}' - \mathbf{r}\|^2). \tag{A.22}$$

Then, substituting into the first integral and noting that $\|\mathbf{r}' - \mathbf{r}\| = \mathcal{O}(h)$, we find:

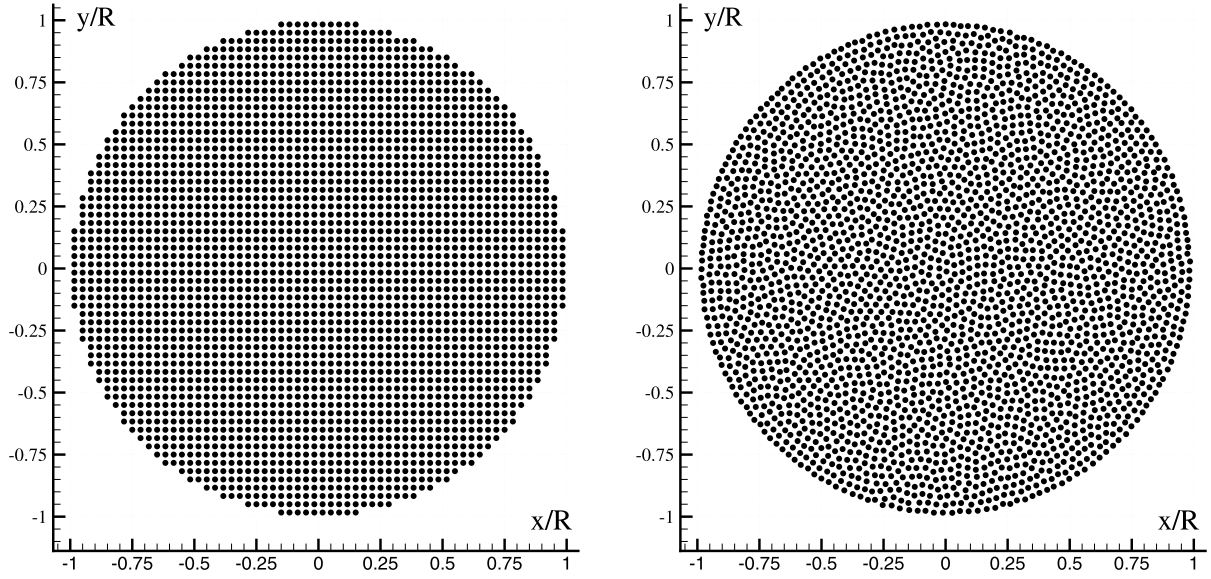


Fig. 15. Evolution of an elliptical drop. Initialization using a Cartesian grid (left) and the particle packing algorithm (right).

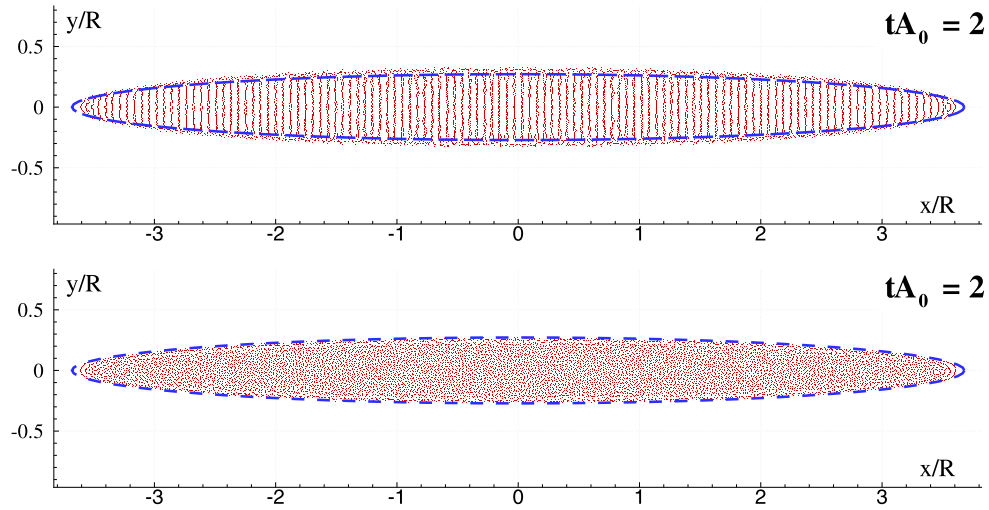


Fig. 16. Evolution of an initially circular patch of fluid using a Cartesian grid (top) and the particle packing algorithm (bottom). Dashed lines indicate the analytical solution for the domain boundary.

$$\begin{aligned} \langle \nabla f \rangle &= f \int_{\partial \Omega} W \mathbf{n}' dV' + (\nabla f)_k \int_{\partial \Omega} (\mathbf{r}' - \mathbf{r})_k W \mathbf{n}' dV' \\ &+ \int_{\Omega} f' \nabla W dV' + \mathcal{O}(h). \end{aligned}$$

Now, applying the divergence theorem once again and the property $\nabla' W(\mathbf{r}' - \mathbf{r}) = -\nabla W(\mathbf{r}' - \mathbf{r})$, it follows:

$$\begin{aligned} \langle \nabla f \rangle &= -f \nabla \Gamma + (\nabla f)_k \int_{\Omega} \nabla' [(\mathbf{r}' - \mathbf{r})_k W] dV' \\ &+ \int_{\Omega} f' \nabla W dV' + \mathcal{O}(h) \\ &= -f \nabla \Gamma + \nabla f \int_{\Omega} W dV' + \nabla f \int_{\Omega} (\mathbf{r}' - \mathbf{r}) \otimes \nabla' W dV' \\ &+ \int_{\Omega} f' \nabla W dV' + \mathcal{O}(h) \end{aligned}$$

$$\begin{aligned} &= -f \nabla \Gamma + \Gamma \nabla f - \nabla f \int_{\Omega} (\mathbf{r}' - \mathbf{r}) \otimes \nabla W dV' \\ &+ \int_{\Omega} f' \nabla W dV' + \mathcal{O}(h). \end{aligned}$$

This formula can be rewritten as follows:

$$\begin{aligned} \langle \nabla f \rangle &= \int_{\Omega} (f' - f) \nabla W dV' + \Gamma \nabla f \\ &- \nabla f \int_{\Omega} (\mathbf{r}' - \mathbf{r}) \otimes \nabla W dV' + \mathcal{O}(h). \end{aligned} \quad (\text{A.23})$$

Then, using the expansion (A.22) once again, we finally obtain:

$$\langle \nabla f \rangle = \Gamma \nabla f + \mathcal{O}(h). \quad (\text{A.24})$$

In a similar way, we get:

$$\langle f \rangle = \int_{\Omega} f' W dV' = f \int_{\Omega} W dV' + \mathcal{O}(h) = \Gamma f + \mathcal{O}(h). \quad (\text{A.25})$$

Now, basing on (A.23), we can write:

$$\begin{aligned} \langle \nabla f \rangle &= \int_{\Omega} f' \nabla W dV' = \langle \nabla f \rangle + f \nabla \Gamma - \Gamma \nabla f \\ &\quad + \nabla f \int_{\Omega} (\mathbf{r}' - \mathbf{r}) \otimes \nabla W dV' + \mathcal{O}(h). \end{aligned}$$

Then, substituting (A.24) and (A.25), we finally get:

$$\int_{\Omega} (\mathbf{r}' - \mathbf{r}) \otimes \nabla W dV' = \Gamma \mathbf{1} + \mathcal{O}(h). \tag{A.26}$$

Substituting this formula back into (A.23), we obtain a consistent formulation for $\langle \nabla f \rangle$, that is:

$$\langle \nabla f \rangle = \int_{\Omega} (f' - f) \nabla W dV' + \mathcal{O}(h). \tag{A.27}$$

The expansion (1.11) for the divergence of the velocity field comes directly by applying (A.27) and (A.24) on each component of \mathbf{u} . Similarly, the expansion (1.12) for the pressure gradient is obtained by summing $2f \nabla \Gamma$ to (A.27) and, then, using the relation (A.24).

Appendix B. Lagrangian structure of the particle packing system

To show the Lagrangian structure of the particle packing algorithm, we first need to prove the following equation:

$$\frac{DW_{ij}}{Dt} = -(\mathbf{u}_j - \mathbf{u}_i) \cdot \nabla_i W_{ij}. \tag{B.28}$$

Proof. Since $W_{ij} = W(s_{ij}, h)$ where $s_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|$, we get:

$$\frac{DW_{ij}}{Dt} = \frac{\partial W}{\partial s_{ij}} \frac{Ds_{ij}}{Dt} = \frac{\partial W}{\partial s_{ij}} \frac{(\mathbf{u}_j - \mathbf{u}_i) \cdot (\mathbf{r}_j - \mathbf{r}_i)}{s_{ij}}, \tag{B.29}$$

and the last equality has been obtained using $D\mathbf{r}_i/Dt = \mathbf{u}_i$. Then, the equality in (B.28) is obtained by using the following identity:

$$\nabla_i W_{ij} = -\frac{(\mathbf{r}_j - \mathbf{r}_i)}{s_{ij}} \frac{\partial W}{\partial s_{ij}}. \quad \square \tag{B.30}$$

Multiplying system (2.16) by \mathbf{u}_i , we find:

$$\frac{D}{Dt} \left(\frac{\|\mathbf{u}_i\|^2}{2} \right) = -\beta \sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 - \zeta \|\mathbf{u}_i\|^2. \tag{B.31}$$

Then, applying the summation over the fluid particles (here indicated by the starred summation), we get:

$$\frac{D\mathcal{T}}{Dt} = -\beta \sum_i^* \sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 - 2\zeta \mathcal{T}, \tag{B.32}$$

where $\mathcal{T} = \sum_i^* \|\mathbf{u}_i\|^2/2$ is the specific kinetic energy. Denoting the summations over the solid particles through a barred series, the double summation on the right-hand side of (B.32) can be rearranged as follows:

$$\begin{aligned} &\sum_i^* \sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 \\ &= \sum_i^* \left[\sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 + \overline{\sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0} \right] \\ &= \sum_i^* \sum_j^* \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 + \sum_i^* \overline{\sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0}. \end{aligned}$$

The velocity inside the first double summation is rewritten as $\mathbf{u}_i = (\mathbf{u}_i + \mathbf{u}_j)/2 + (\mathbf{u}_i - \mathbf{u}_j)/2$. Conversely, the velocity inside the second double summation can be rewritten as $\mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_j$, since the solid particles have null velocity (that is, $\mathbf{u}_j = 0$). Then, we obtain:

$$\begin{aligned} \sum_i^* \sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 &= \sum_i^* \sum_j^* \frac{(\mathbf{u}_i + \mathbf{u}_j)}{2} \cdot \nabla_i W_{ij} V_0 \\ &\quad + \sum_i^* \sum_j^* \frac{(\mathbf{u}_i - \mathbf{u}_j)}{2} \cdot \nabla_i W_{ij} V_0 \\ &\quad + \sum_i^* \overline{\sum_j (\mathbf{u}_i - \mathbf{u}_j)} \cdot \nabla_i W_{ij} V_0. \end{aligned}$$

Because of its symmetry properties, the double summation containing $(\mathbf{u}_i + \mathbf{u}_j)$ is identically null. Then, using the identity (B.28), we finally get:

$$\begin{aligned} &\sum_i^* \sum_j \mathbf{u}_i \cdot \nabla_i W_{ij} V_0 \\ &= \sum_i^* \sum_j^* \frac{(\mathbf{u}_i - \mathbf{u}_j)}{2} \cdot \nabla_i W_{ij} V_0 + \sum_i^* \overline{\sum_j (\mathbf{u}_i - \mathbf{u}_j)} \cdot \nabla_i W_{ij} V_0 \\ &= \frac{D}{Dt} \left[\frac{1}{2} \sum_i^* \sum_j^* W_{ij} V_0 + \sum_i^* \overline{\sum_j W_{ij} V_0} \right]. \end{aligned}$$

Substituting this equality into (B.32), we obtain the total energy equation of the packing algorithm:

$$\frac{D\mathcal{E}}{Dt} = -2\zeta \mathcal{E}, \tag{B.33}$$

where $\mathcal{E} = \mathcal{T} + \mathcal{V}$ and

$$\mathcal{V} = \frac{\beta}{2} \sum_i^* \sum_j^* W_{ij} V_0 + \beta \sum_i^* \overline{\sum_j W_{ij} V_0} \tag{B.34}$$

is the specific potential energy of the system. The first term on the right-hand side represents the component coming from the fluid domain while the second one accounts for the action of the solid boundary. If the damping effects are neglected, the total energy of the system is conserved during the evolution.

References

- [1] J.J. Monaghan, Simulating free surface flows with SPH, *Journal of Computational Physics* 110 (1994) 399–406.
- [2] G. Oger, C. Leroy, E. Jacquin, D. Le Touzé, B. Alessandrini, Specific pre/post treatments for 3-D SPH applications through massive HPC simulations, in: *Proc. 4th Spheric Workshop on SPH, Nantes, France, May 26–29, 2009*.
- [3] K. Shimada, Physically-based mesh generation; automated triangularization of surface and volumes via Bubble packing, PhD thesis, MIT, 1993.
- [4] J.J. Monaghan, Smoothed particle hydrodynamics, *Rep. Prog. Phys.* 68 (2005) 1703–1759.
- [5] S.D. Shao, Incompressible SPH simulation of wave breaking and overtopping with turbulence modeling, *Int. J. Numer. Methods Fluids* 50 (5) (2006) 597–621.
- [6] P.A. Madsen, H.A. Schäffer, A discussion of artificial compressibility, *Coast. Eng.* 53 (2006) 93–98.
- [7] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, G. Graziani, δ -SPH model for simulating violent impact flows, *Comput. Methods Appl. Mech. Engrg.* 200 (2011) 1526–1542.
- [8] A. Colagrossi, M. Antuono, D. Le Touzé, Theoretical considerations on the free surface role in the SPH model, *Phys. Rev. E* 79 (5) (2009) 113 (056701).
- [9] J. Bonet, T.S.L. Lok, Variational and momentum preservation aspects of smoothed particle hydrodynamics formulations, *Comput. Methods Appl. Mech. Engrg.* 180 (1999) 97–115.
- [10] N.J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, *Int. J. Numer. Methods Engrg.* 66 (2006) 2064–2085.

- [11] M. Liu, G. Liu, Smoothed particle hydrodynamics (SPH): An overview and recent developments, *Arch. Comput. Methods Eng.* 17 (1) (2010) 25–76.
- [12] J.J. Monaghan, J.B. Kajtár, SPH particle boundary forces for arbitrary boundaries, *Comput. Phys. Commun.* 180 (10) (2009) 1811–1820.
- [13] J.C.M. Marongiu, Méthode numérique lagrangienne pour la simulation d'écoulements à surface libre – Application aux turbines Pelton, PhD thesis, École Central de Lyon, 2007.
- [14] M. Vidyasagar, *Nonlinear Systems Analysis*, second ed., SIAM, 2002.
- [15] M. Antuono, A. Colagrossi, S. Marrone, C. Lugni, Propagation of gravity waves through an SPH scheme with numerical diffusive terms, *Comput. Phys. Commun.* 182 (2011) 866–877.
- [16] M. Ferrand, D. Laurence, B.D. Rogers, D. Violeau, Improved time scheme integration approach for dealing with semi analytical boundary conditions with SPARTACUS2D, in: 5th International Spheric Workshop, Manchester, 22–25 June 2010.
- [17] A. Colagrossi, A meshless Lagrangian method for free-surface and interface flows with fragmentation, PhD thesis, Università “Sapienza”, Rome.