Journal of Computational Physics 229 (2010) 3652–3663

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Fast free-surface detection and level-set function definition in SPH solvers

S. Marrone [a,c], A. Colagrossi [a,*], D. Le Touzé [b], G. Graziani [c]

[a] INSEAN, The Italian Ship Model Basin, Roma, Italy
[b] Fluid Mechanics Lab., École Centrale Nantes/CNRS, Nantes, France
[c] Department of Mechanics and Aeronautics, University of Rome "La Sapienza", Italy

ARTICLE INFO

Article history:
Received 15 July 2009
Received in revised form 11 January 2010
Accepted 13 January 2010
Available online 25 January 2010

Keywords:
Smoothed particle hydrodynamics
Free-surface flows
Fluid-structure impact flows
Level-set function
Flow visualization

ABSTRACT

The present paper proposes a novel algorithm to detect the free-surface in particle simulations, both in two and three dimensions. Since the proposed algorithms are based on SPH interpolations their implementation does not require complex geometrical procedures. Thus the free-surface detection can be easily embedded in SPH solvers, without a significant increase of the CPU time. Throughout this procedure accurate normal vectors to the free-surface are made available. Then it is possible to define a level-set function algorithm which is presented in detail. The latter allows in-depth analyses of three-dimensional free-surface simulations by using standard visualization tools, including internal features of the flow. The algorithms proposed for detecting free-surface particles and defining the level-set function are validated on simple and complex two- and three-dimensional flow simulations. The usefulness of the proposed procedures to post-process and analyze complex flows are illustrated on realistic examples.

1. Introduction

In recent years the SPH method has been successfully applied to problems involving free-surface flows with fragmentation. In order to analyze flows with complex free-surface patterns (fragmentations, air entrapment, etc.) and to face a larger range of problems it is required to know which particles belong to the free-surface. This detection can also be required for the enforcement of suitable boundary conditions along the free-surface (surface tension, isothermal condition, etc.) in order to deal with different physical phenomena and flow behaviors. Dilts [1] developed an algorithm for the free-surface tracking that can detect free-surface particles in a robust and reliable way and that is applicable to any meshless method. However, it is quite difficult to implement, particularly in its extension to three-dimensional simulations [2].

In this work a novel algorithm for free-surface detection is presented. Such a scheme, based on the properties of the SPH kernel, is easy to implement both in two and three dimensions, and computationally cheap. The accuracy of the method is comparable to that of the method proposed by Dilts. It is possible, indeed, to catch small cavities of diameter as small as $2h$ (h being the smoothing length) and fluid elements with dimension smaller than h (like jets and drops). Thanks to these valuable features, the proposed algorithm can be used at each time-step of the simulations, without an appreciable increase of the CPU time.

Moreover, free-surface detection permits strong improvement of the post-processing phase, particularly in three-dimensional simulations with complex flow features. In fact, if one uses merely a SPH output, flow analysis is problematic since data are known on scattered points, and it is difficult to obtain contour plots, slices and iso-surfaces. Such an analysis can be performed in a straightforward way using standard tools if data are interpolated on a regular grid. In this context it is

* Corresponding author. Tel.: +39 0650299343; fax: +39 065070619.
E-mail address: a.colagrossi@insean.it (A. Colagrossi).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.01.019

Notation

$\boldsymbol{B}$	renormalization matrix
$\boldsymbol{x}$	spatial position
W	kernel function
ΔV	particle volume
λ	minimum eigenvalue of $\boldsymbol{B}^{-1}$
$\mathbb{N}$	set of all the fluid particles
$\mathbb{F}$	set of the particles belonging to the free-surface
$\mathbb{E}$	set of particles with $\lambda \leqslant 0.20$
$\mathbb{B}$	set of particles with $0.20 < \lambda \leqslant 0.75$
$\mathbb{I}$	set of particles with $0.75 < \lambda$
h	kernel smoothing length
dx	average particle spacing
$\boldsymbol{n}$	unit vector normal to the free-surface
$\boldsymbol{\tau}$	unit vector tangential to the free-surface
ϕ	level-set function

useful to define a level-set function among the grid nodes. This function permits to distinguish between nodes inside and outside of the fluid domain. To define the level-set function the detection of the free-surface particles is required as a first step. Besides its utility for flow analysis the definition of this function may be useful to extend remeshing techniques, see *e.g.* [3], to free-surface flow problems.

In the present paper the algorithm for free-surface detection is described and validated in Sections 2 and 3, for both two- and three-dimensional cases. To assess the accuracy of the algorithm, the validation has first been performed using simple geometries, and then on complex flow cases. Finally, in Section 4, we describe the procedure to define a level-set function which can be useful to interpolate flow data on a regular grid. Post-processing results are shown to illustrate the proposed algorithm capabilities.

2. 2D algorithm

2.1. Algorithm details

The algorithm is composed of two steps: in the first one the properties of the renormalization matrix, defined by Randles and Libersky [4], are used to find particles next to the free-surface. This first step strongly decreases the number of particles that will be processed in the second step. In the second step the algorithm, by means of geometric properties, detects particles that actually belong to the free-surface and evaluates their local normals.

In order to validate it the algorithm has been applied to simple geometries as well as to a dam-break problem (see *e.g.* [5]) which sketch and initial conditions are displayed in Fig. 1. The complex free-surface behavior of the impact flow simulated is displayed in Fig. 2. In this figure the plotted rectangles delimit the zones which are enlarged in Fig. 3 to highlight the flow complexity there, and which are the most challenging for the detection algorithm.

The method used to perform the first step of the algorithm was proposed by Doring [6]; it exploits eigenvalues of the renormalization matrix [4] defined as:

$$\boldsymbol{B}(\boldsymbol{x}_i) = \left[\sum_j \nabla W_j(\boldsymbol{x}_i) \otimes (\boldsymbol{x}_j - \boldsymbol{x}_i)\Delta V_j\right]^{-1} \tag{1}$$

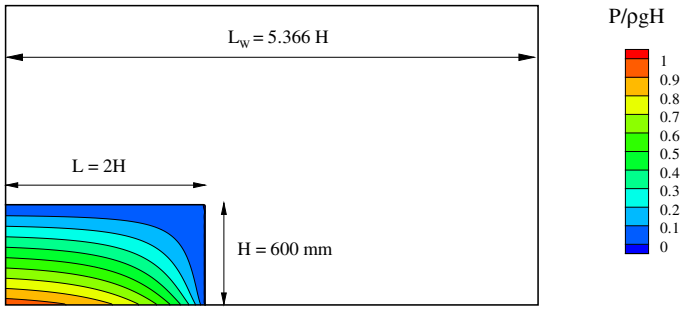


Fig. 1. Sketch and initial condition of the dam-break problem considered, see *e.g.* [5].

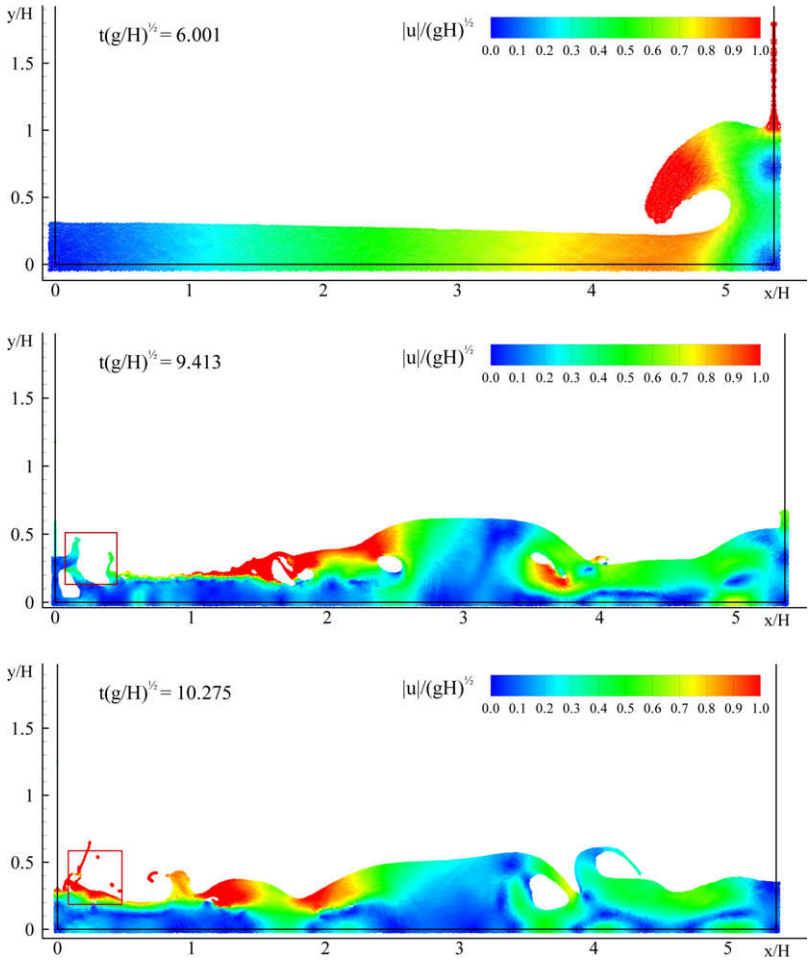


Fig. 2. Different time instants of the impact flow after the dam-break. The rectangles delimit the zones enlarged in Fig. 3.

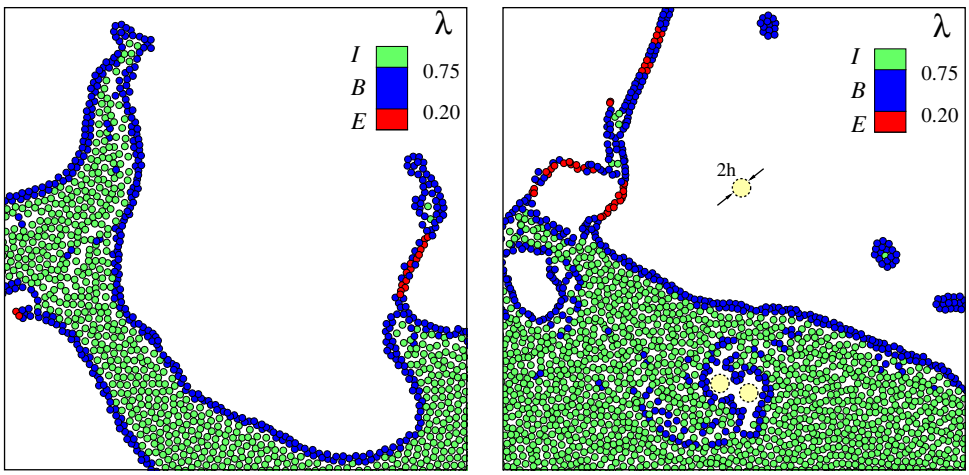


Fig. 3. Values of the minimum eigenvalue λ of the matrix $\boldsymbol{B}^{-1}$. Red particles belong to subset $\mathbb{E}$; green particles belong to subset $\mathbb{I}$; blue particles belong to subset $\mathbb{B}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where ΔV_j is the volume of the jth particle and $W_j(\boldsymbol{x}_i)$ is the interpolating kernel centered on particle j and evaluated in the position $\boldsymbol{x}_i$. The spatial derivatives of $W_j(\boldsymbol{x}_i)$ are referred with respect to the position $\boldsymbol{x}_i$.

Doring showed that the value of the minimum eigenvalue, λ, of the matrix $\boldsymbol{B}^{-1}$ depends on the spatial organization of the particles j in the neighborhood of the considered calculation point i. When going away from the fluid domain this eigenvalue λ tends theoretically to 0 while inside this domain λ tends theoretically to 1. This allows to determine regions of the fluid domain where the free-surface can lie or not.

Let us define $\mathbb{N}$ as the set of all the fluid particles and $\mathbb{F} \subset \mathbb{N}$ as the subset of particles belonging to the free-surface. Then, computing λ for each particle, it is possible to further define three complementary subsets: $\mathbb{E}$ composed by particles belonging to thin jets and drops, characterized by low values of λ; $\mathbb{I}$ composed by interior particles far from the free-surface, characterized by high values of λ; and $\mathbb{B}$ composed by particles which are close to the free-surface or are in regions of the domain where particles are not uniformly spread. Particles belonging to the last subset are characterized by intermediate values of λ. We have that $\mathbb{N} = \mathbb{E} \cup \mathbb{I} \cup \mathbb{B}$. The free-surface particles subset $\mathbb{F}$ is thus composed of all the particles of subset $\mathbb{E}$ and a part of the elements of subset $\mathbb{B}$.

To identify these subsets it is possible to define threshold values of λ which depend on the considered kernel. Here we use a renormalized gaussian kernel shape, see [5], with a support radius equal to $3h$, where h is the smoothing length and is equal to $1.33\,dx$. dx is the average particle spacing which means that in two dimensions a particle has $\simeq dx^2$ for volume and an average number of neighbors equal to 50. All the results and the conclusions presented in this work have to be considered valid only for the aforementioned h/dx ratio. Several tests have been performed to set the proper thresholds for λ. Being i the particle under examination, the values found are the following:

$$\begin{cases} i \in \mathbb{E} & \Longleftrightarrow & & \lambda \leqslant 0.20 \\ i \in \mathbb{B} & \Longleftrightarrow & 0.20 & < \lambda \leqslant 0.75 \\ i \in \mathbb{I} & \Longleftrightarrow & 0.75 & < \lambda \end{cases} \tag{2}$$

In this way the first step of the algorithm computes the minimum eigenvalue for each particle and gives a first rough detection of the free-surface. This operation has a very low computational cost especially if the renormalization matrix is already computed in the SPH scheme, as *e.g.* in the formulation proposed in [4]. In Fig. 3 one can observe the result of this first detection. Particles next to the free-surface and near cavities are correctly detected but this also happens for some particles within internal fluid regions characterized by non-uniform distribution. Conversely, particles which belong to drops and thin jets are easily captured by the lower threshold and directly identified as free-surface particles (red coloured).

In the second step of the algorithm, a more precise and reliable control is performed on particles belonging to $\mathbb{B}$ in order to complete the free-surface detection. The proposed method is based on the fact that, inside the fluid domain, the sum of the kernel gradient over neighbors is very close to zero. When a particle, instead, is near the free-surface, such sum is a good approximation of the local normal $\boldsymbol{n}$ to the free-surface, see [4]. Since the accuracy of the evaluation of this vector depends on the particle disorder, it is possible to get a more accurate evaluation by using again the renormalization matrix:

$$\boldsymbol{n}(\boldsymbol{x}_i) = \frac{\boldsymbol{v}(\boldsymbol{x}_i)}{|\boldsymbol{v}(\boldsymbol{x}_i)|}; \quad \boldsymbol{v}(\boldsymbol{x}_i) = -\boldsymbol{B}(\boldsymbol{x}_i) \sum_j \nabla W_j(\boldsymbol{x}_i) \Delta V_j \tag{3}$$

This is a standard way to improve accuracy of the SPH interpolation within the fluid domain, see *e.g.* [4]. Here we use the same principle to improve the accuracy of the evaluation of the local normal. Once this vector $\mathbf{n}$ is known, it is possible to define a region of the domain like the one sketched in Fig. 4. The algorithm then checks whether or not at least one neighbor particle lies in this region, hereinafter referred to as *scan region*. If no neighbor is found inside it, the candidate particle belongs to the free surface. It must be noted that inside the fluid domain this region cannot be void since $h = 1.33\,dx$.

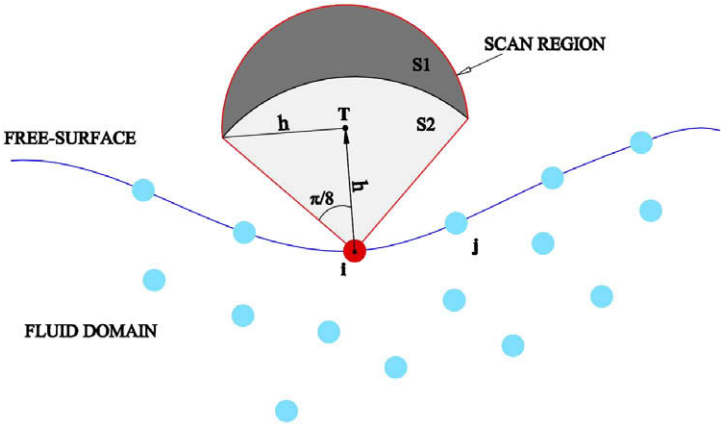


Fig. 4. Sketch of the regions used in the algorithm.

This control is carried out by the algorithm in the following way. We denote by $i \in \mathbb{B}$ the particle that is under examination and by $j \in \mathbb{N}$ the neighbor of i which is included in a $3h$ radius distance. We define also the point T at distance h from i in the normal direction and the unit vector $\boldsymbol{\tau}$ perpendicular to $\boldsymbol{n}$. The conditions to assess whether particle i belongs or not to the free-surface are therefore:

$$\begin{cases} \forall j \in \mathbb{N}\left[|\boldsymbol{x}_{ji}| \geqslant \sqrt{2}h, \quad |\boldsymbol{x}_{jT}| < h\right] & \Rightarrow \quad i \notin \mathbb{F} \\ \forall j \in \mathbb{N}\left[|\boldsymbol{x}_{ji}| < \sqrt{2}h, \quad |\boldsymbol{n} \cdot \boldsymbol{x}_{jT}| + |\boldsymbol{\tau} \cdot \boldsymbol{x}_{jT}| < h\right] & \Rightarrow \quad i \notin \mathbb{F} \\ \text{otherwise} & \quad i \in \mathbb{F} \end{cases} \tag{4}$$

where notation $A_{ij} = A_i - A_j$ has been used. If the first condition is true it means that the neighbor under examination is in the dark grey region (S1) in Fig. 4 while, if the second condition is true, it means that the neighbor is in the pearl grey region (S2). The two regions together form the *scan region*.

If any neighbor is located in the *scan region* it means that there is no cavity in the normal direction, or that this cavity has a diameter less than $2h$, so that particles inside it are deeply interacting and it is not a true cavity. Therefore, through this process we are tracking the free-surface of cavities of diameter larger than $2h$. More details on the geometry used for the *scan region* are given in Appendix A.1.

It can be noticed that for some particle distributions, *e.g.* for thin jets, the sum of the kernel gradient can go to zero, giving wrong values for vector $\boldsymbol{n}$. However, in this circumstance the eigenvalue λ will be very low and the particle will hence have already been detected as a free-surface particle in the first step of the algorithm. This first step has thus two functions: it is an efficient selection allowing to quickly perform the second step, and a tool to detect particles belonging to jets and drops which could hardly be detected by the second step of the algorithm.

2.2. Validation

In the left part of Fig. 5 the algorithm is validated for an elliptic fluid domain, with 0.661 eccentricity. Since the evaluation of the free-surface normals is critical for the algorithm effectiveness, the ones calculated by the algorithm are compared to the analytical values. In this comparison a uniform distribution of particles had to be used inside the fluid domain in order to assign exact volumes to the particles. Results are reported in Table 1 in terms of average relative angle error. The convergence is close to quadratic.

In the right part of Fig. 5 an elliptic cavity with minor axis equal to $2h$ was introduced in the ellipse. This cavity has thus the minimum dimension which the algorithm is able to detect.

After this simple test, the algorithm is further assessed on the complex flow situations presented in Fig. 3. The free-surface particles detected by the algorithm are plotted in pink in Fig. 6. Despite the geometrical complexity of these configurations, the proposed method is able to provide a very good qualitative estimation of the particles which form the boundary. In particular one can observe that approximately circular cavities of diameter just larger than $2h$ are well detected.

Even though the algorithm requires a cycle on the neighbors for each particle involved in the second step, it has still a very low computational cost. Actually, all but a few percents of the particles are filtered out in the first step.

The CPU time cost of the algorithm is lower than 5% of the total cost of a standard SPH calculation, where the latter includes the calculation of the neighbor list and the summations for the continuity and momentum equation. It is the computation of the renormalization matrix which takes the most part of the CPU time required by the algorithm (about 90%). Hence, if the renormalization matrix is already evaluated in the numerical scheme, see *e.g.* [7,4], the increase of CPU time

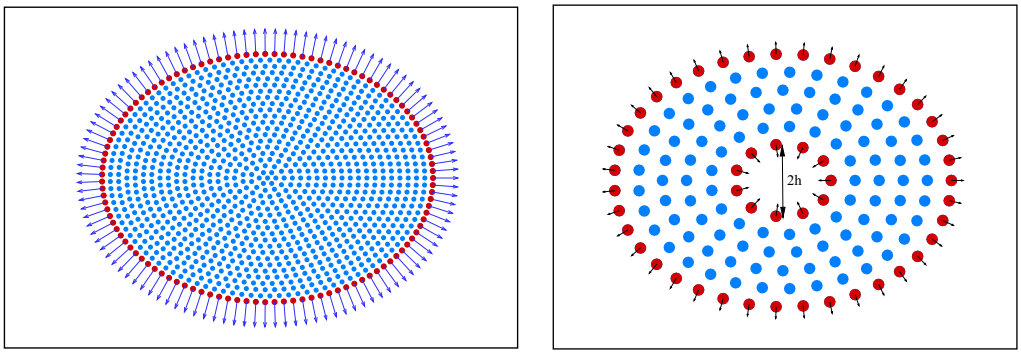


Fig. 5. Left: free-surface detection of an elliptic fluid domain. Right: free-surface detection in an elliptic cavity of minor axis equal to $2h$. Red particles are those detected by the proposed free-surface algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Average relative angle error between the analytically computed normals and the one evaluated by Eq. (3). h/a denotes the ratio between the smoothing length and the semimajor axis.

$h/a \times 10^{-2}$	7.52	3.76	1.88	0.94	0.63	0.38
ε	2.16	1.23	0.68	0.34	0.23	0.13

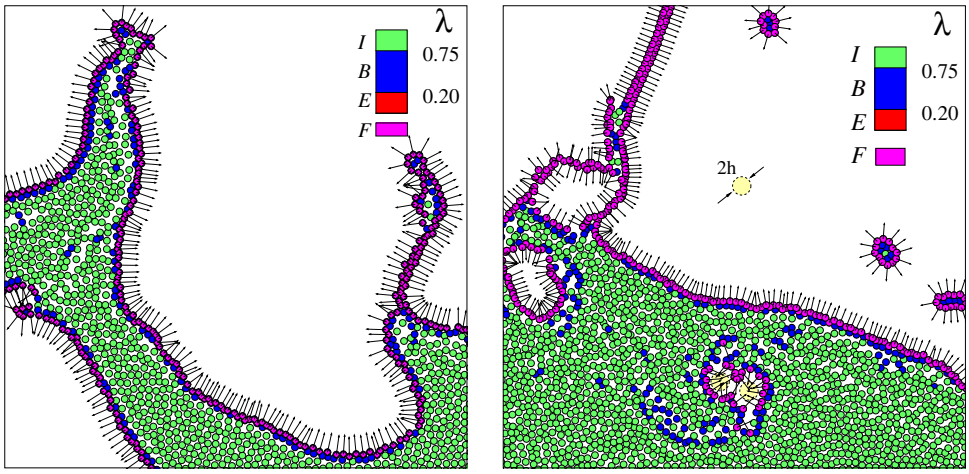


Fig. 6. Results after the application of free-surface detection algorithm: free-surface particles are displayed in pink with their normals; in the bottom part small cavities of $2h$ diameter are shown. A circle with dimension $2h$ is reported for comparison.

due to free-surface detection is absolutely negligible. Moreover, if the algorithm is used only for visualization or analysis purposes, it has to be applied only periodically (typically every 100 time-steps), and its cost becomes again negligible.

3. 3D algorithm

3.1. Algorithm details

The extension of the algorithm to the third dimension is rather straightforward and easy to implement. The algorithm has still the same structure as the two-dimensional one: it is composed of a first step where particles far from the free-surface are filtered out, and of a second step to refine the detection only to the particles belonging to the free-surface. In the first step the minimum eigenvalue λ of the renormalization matrix is again needed for each particle. Again, two thresholds are used which are the same as in two dimensions. These thresholds are valid for the renormalized gaussian kernel with support radius equal to $3h$ and $h/dx = 1.33$ (this means an average number of neighbors equal to 266).

While the first step is formally unchanged in the extension to three dimensions, the second step is slightly modified. The vector $\boldsymbol{n}$ is still evaluated through Eq. (3) but the conditions which define the region to scan become:

$$\begin{cases} \forall j \in \mathbb{N} \quad \left[|\boldsymbol{x}_{ji}| \geqslant \sqrt{2}h, \quad |\boldsymbol{x}_{jT}| < h \right] & \Rightarrow \quad i \notin \mathbb{F} \\ \forall j \in \mathbb{N} \quad \left[|\boldsymbol{x}_{ji}| < \sqrt{2}h, \quad \arccos\left(\frac{\boldsymbol{n}\cdot\boldsymbol{x}_{ji}}{|\boldsymbol{x}_{ji}|}\right) < \frac{\pi}{4} \right] & \Rightarrow \quad i \notin \mathbb{F} \\ \text{otherwise} & \quad i \in \mathbb{F} \end{cases} \tag{5}$$

Therefore, in an intuitive way, the triangular region in Fig. 4 becomes a cone in three dimensions, while the semicircle becomes a hemisphere.

3.2. Validation

In the three-dimensional case it is more complex to test the algorithm and assess its accuracy.

Indeed, unlike in 2D cases, in 3D problems even a qualitative evaluation of the particles belonging to the free-surface could be quite difficult. In particular cavities and jets are generally blurred since particles are spread in the space in a disordered way.

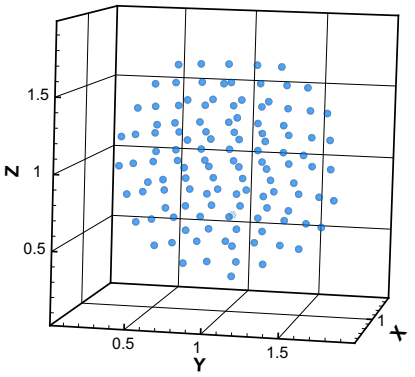
In order to overcome this problem the algorithm was tested on a particle distribution with a free-surface which is known *a priori*. According to this strategy two tests were performed. In the first one particles are arranged to form a sphere with a

spherical cavity inside, as sketched in left panel of Fig. 7. In order to have a regular distribution, particles are placed on concentric spheres with radius increasing by dx, where dx is cubic root of the particle volume. On each sphere, particles are equispaced with a distance approximatively equal to dx. The cavity inside the sphere has a diameter equal to $4dx \simeq 3h$. This quantity is actually slightly larger than $2h$ which is the limit size for detecting a cavity. However, choosing a smaller cavity radius was not possible in practice. Indeed, this would have resulted in having too few particles distributed on the cavity surface, resulting in a bad approximation of their volumes, and consequently providing an inappropriate test of the algorithm. The particle distribution is shown in the right panel of Fig. 7. In such a view it is not possible to detect the cavity inside the sphere.

The result given by the algorithm is presented in Fig. 8. The two detected free-surfaces (the ones from the inner and outer spheres) are shown separately to better evidence the normals. We notice that free-surface particles and normals are correctly evaluated both in the cavity and along the external surface. Similarly to the 2D case, in Table 2 calculated free-surface normals are compared to analytical values in terms of average relative angle error for a spherical fluid domain. Again, the convergence rate is close to quadratic.

In order to assess the capability of the algorithm to capture cavities of dimension equal to $2h$, a more complex test was performed. As displayed in the left panel of Fig. 9, a toroidal cavity was created inside the sphere. The width of the torus is $2h$ and the torus ring diameter is $14dx$. Again, when showing the whole particle set, it is not possible to distinguish the toroidal cavity inside the sphere (see right panel of Fig. 9). The free-surface particles detected by the algorithm are shown in Fig. 10. Also in this case free-surface particles and normals are correctly evaluated.

To illustrate the method capabilities on an actual complex 3D situation we show the results of the application of the algorithm on an impact flow (see *e.g.* [8] and [9]). The sketch of the problem geometry is displayed in Fig. 11. In Fig. 12 the whole



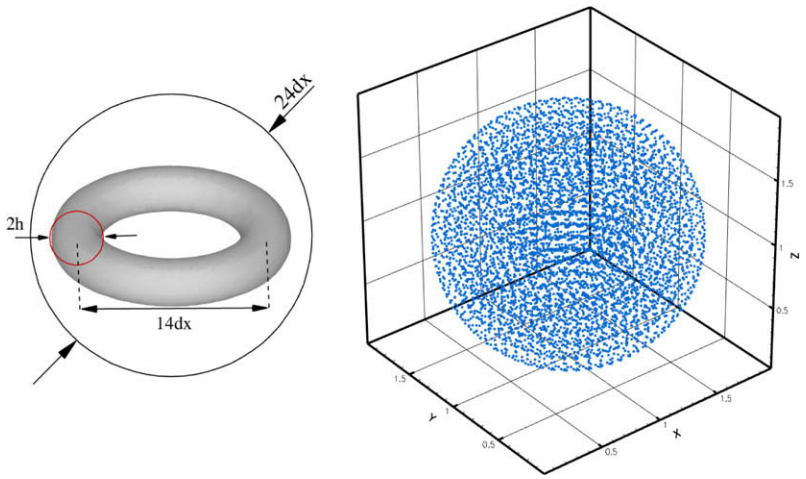
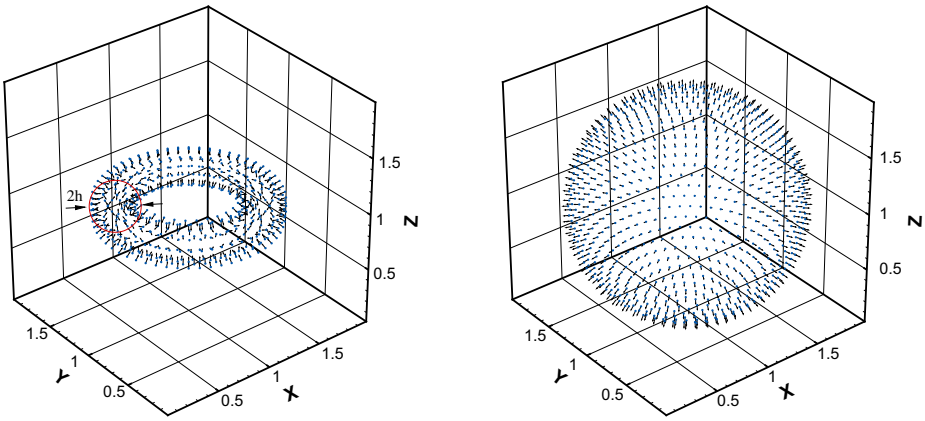


Fig. 9. Left panel: sketch of the second tested geometry. Right panel: corresponding set of particles.



particle distribution and the detected free-surface particles are shown for a time instant just after the impact. Anyway in these plots only a few flow features can be analyzed. Therefore, a useful tool to improve the flow analysis is proposed in the next section.

4. Definition of a level-set function

The interpolation of particle data onto a regular grid can be useful both for allowing an in-depth analysis of the flow features through adequate post-processing, or for remeshing the particles during free-surface flow simulations. In order

to perform this procedure it is necessary to locate the free-surface across the grid. In other words, one needs to separate nodes inside the fluid domain from those outside. This can be done from the knowledge of the free-surface particle subset

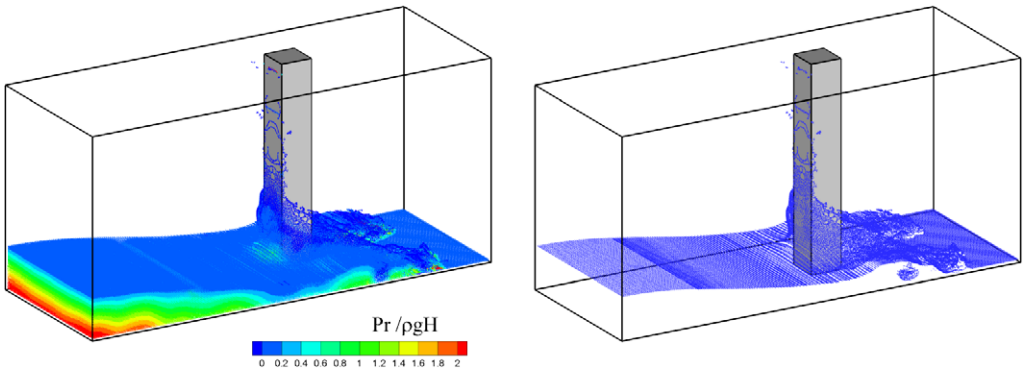
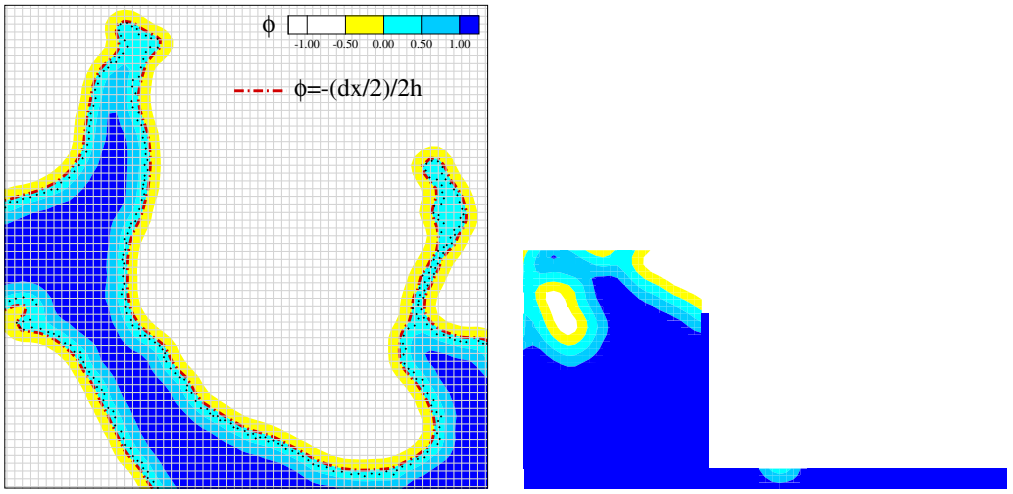


Fig. 12. 3D impact against a tall structure after a dam-break. Left: whole particle distribution over the fluid domain; the colours represent the pressure. Right: free-surface particles detected by the algorithm at same time instant.



$\mathbb{F}$. Let us consider a regular Cartesian grid of spatial resolution dx, and which encloses all the computational domain. For each node N close to free-surface particles of subset $\mathbb{F}$, the nearest free-surface particle F_N is detected and the scalar quantity d_{NF_N} is evaluated through:

$$d_{NF_N} = (\boldsymbol{x}_{F_N} - \boldsymbol{x}_N) \cdot \boldsymbol{n}_{F_N} \tag{6}$$

where $\boldsymbol{n}_{F_N}$ is the normal to the free-surface evaluated in F_N. For each node it is now possible to define a level-set function $\phi(\boldsymbol{x}_N)$:

$$\phi(\boldsymbol{x}_N) = \begin{cases} -1 & d_{NF_N} \leqslant -2h \\ d_{NF_N}/2h & -2h < d_{NF_N} < 2h \\ +1 & d_{NF_N} \geqslant 2h \end{cases} \tag{7}$$

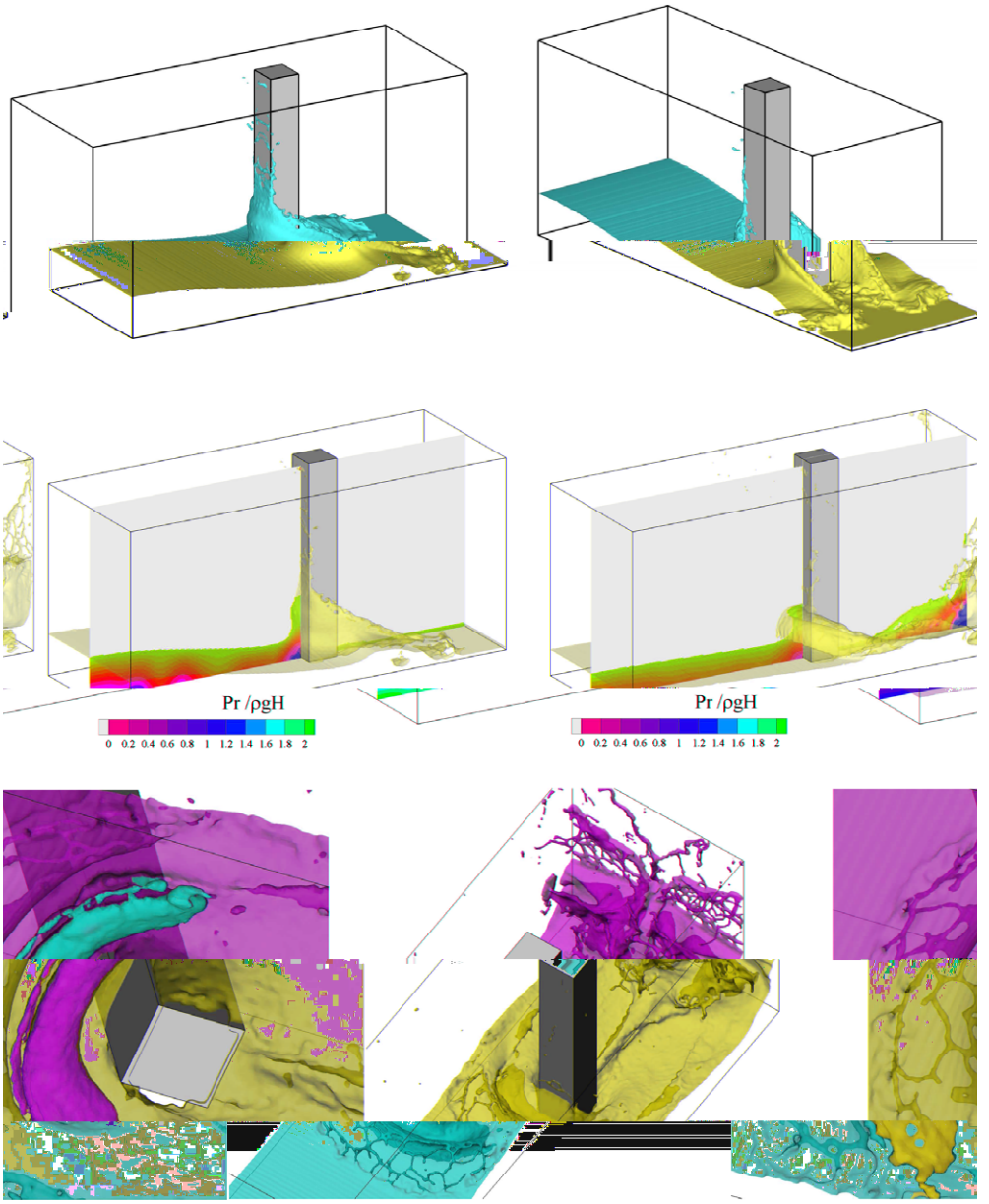


Fig. 15. 3D Flow impact against a tall structure after a dam-break. Top: two views of the free-surface represented by iso-surface $\phi = -dx/4h$. Middle: free-surface and pressure contours interpolated on a vertical plane at two different time instants. Bottom: two views of the free-surface iso-surface at a later stage of the flow evolution; on the right plot a detail of the cavity generated in front of the column is given (seen from below).

This function is positive inside the fluid, negative outside it and equal to 0 along the free-surface where $d_{NF_N} = 0$. More precisely, under the assumption that the actual free-surface location is at a distance $dx/2$ from the center of mass of particles belonging to $\mathbb{F}$, the value of ϕ on the free-surface has to be $\phi = -dx/4h$ and not zero anymore.

At the computational level, the procedure to evaluate ϕ can be executed in a fast manner. At each node we first identify the particles present within a $2h$-distance. We denote $\mathbb{N}_{2h}$ this subset of particles through which we can evaluate the function $\phi(\boldsymbol{x}_N)$ as:

$$\begin{aligned} &-1 \quad \mathbb{N}_{2h} = \emptyset \\ &+1 \quad \mathbb{N}_{2h} \neq \emptyset, \quad \mathbb{N}_{2h} \cap \mathbb{F} = \emptyset \\ &d_{NF_N}/2h \quad \mathbb{N}_{2h} \neq \emptyset, \quad \mathbb{N}_{2h} \cap \mathbb{F} \neq \emptyset, \quad \lambda_{F_N} \geqslant 0.1 \\ &-|\boldsymbol{x}_N - \boldsymbol{x}_{F_N}|/2h \quad \mathbb{N}_{2h} \neq \emptyset, \quad \mathbb{N}_{2h} \cap \mathbb{F} \neq \emptyset, \quad \lambda_{F_N} < 0.1 \end{aligned} \tag{8}$$

where λ_{F_N} is the minimum eigenvalue of the matrix $\boldsymbol{B}^{-1}$ [see Eq. (1)] for the particle F_N. For $\lambda_{F_N} < 0.1$ the nearest free-surface particle for the node N is a solitary particle. In such a case the vector $\boldsymbol{n}_{F_N}$ is null (see Eq. (3)) and consequently the scalar product d_{NF_N} is meaningless. Therefore, the latter is directly substituted with the distance $|\boldsymbol{x}_N - \boldsymbol{x}_{F_N}|$. Further, to smooth out the function ϕ on the whole mesh, a Gaussian filter on the nodes is performed once ϕ is evaluated by Eq. (8).

In Fig. 13 the contour plots of function ϕ for the two time instants of Fig. 3 are shown. The free-surface is represented by the red dash-dotted line which corresponds to the contour level $\phi = -dx/4h$, in close agreement with the free-surface particles positions if they were shifted by $dx/2$.

In 3D simulations the interpolation on a regular mesh clearly brings higher benefits. Indeed, the visualization and the flow analysis of a 3D SPH simulation is generally quite difficult. This is highlighted in Fig. 14 where a spherical fluid domain with three concentric toroidal cavities and a small spherical cavity of $3h$ diameter is considered. Even though only the free-surface particles are shown (left panel), it is obvious how difficult it is to detect the geometry of the fluid domain. On the right side of the same figure, the iso-surface $\phi = -dx/4h$ representing the free-surfaces conversely gives a clear representation of the fluid domain, thanks to transparency features which are a standard visualization tool on a regular mesh.

Some interpolation results of the impact flow shown in Fig. 12 are presented in the following. This operation has been carried out with a Moving-Least Square (MLS) interpolator that exactly interpolates a linear field on a regular grid from scattered points, see *e.g.* [10]. The representation of the free-surface is given by the iso-contour $\phi = -dx/4h$ in Fig. 15. In particular, the pressure distribution during the impact is evidenced through contours interpolated on a vertical plane inside the domain. Finally, the last plots of Fig. 15 show how it is possible to analyze flow internal details through the reconstruction of the free-surface as an iso-surface. In particular, entrapped bubbles and a large tube cavity due to a strong wave breaking are clearly identifiable.

5. Conclusion

An efficient algorithm capable to detect the free-surface in SPH methods has been proposed in 2D and 3D simulations. It is composed of two stages. The first stage consists in detecting the particles composing the free-surface. From this information, it is possible to define a level-set function throughout the domain in a second stage. This function can be used to interpolate flow quantities on a cartesian grid, which makes possible the visualization and analysis of flow features using standard visualization tools. The two stages have been carefully validated on test cases of increasing complexity, up to real 3D impact flow.

On these test cases it has been shown how the method allows detecting flow details such that thin jets, drops, entrapped air structures, etc. as well as to recover contour slices, iso-surfaces, etc. in an efficient way. The algorithm proposed in the present paper could be exploited to impose specific conditions on the free-surface, or to extend remeshing techniques to free-surface flow problems.

Acknowledgments

This work was partially supported by Programma Ricerche INSEAN 2007–2009 and Programma di Ricerca sulla Sicurezza funded by Ministero Infrastrutture e Trasporti.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement n225967 "NextMuSE".

Appendix A

A.1. Motivation for the shape of the scan region adopted

The choice of the shape of the *scan region*, composed by a half circle and a half square rather than by a simple circle for instance, is explained in the following.

In top sketch of Fig. 16 a uniform distribution of particles is considered. In such a configuration only the first row of particles should be detected as free-surface. In the left part of the sketch the *scan region* is reported for a particle belonging to the

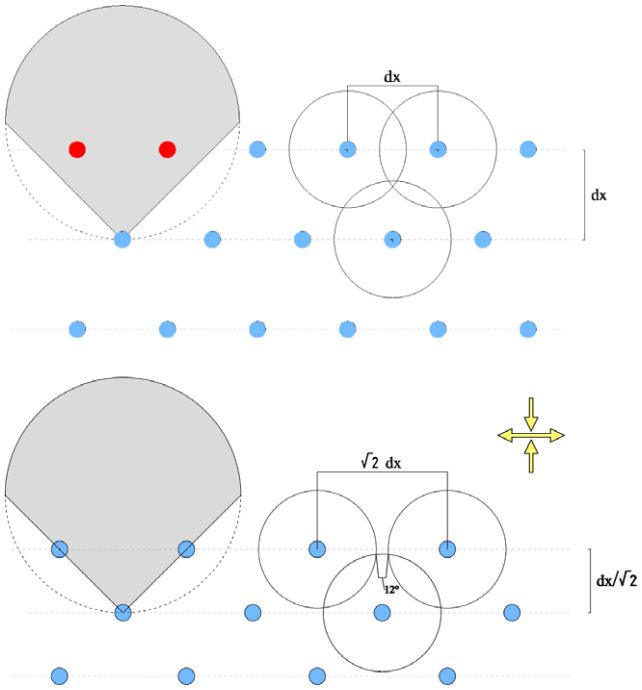


Fig. 16. Sketch of the *scan region* applied on uniform distribution (top panel) and on free-divergence stretched distribution (bottom panel). On the right a comparison with the *arc-method*, see [1].

second row. Two particles are present in the *scan region* which means that the second row is not detected as free-surface. On the right part of the same sketch circles with diameter equal to h have been reported in order to show that also the *arc-method* proposed in [1] gives the same result. In the bottom sketch of Fig. 16 a divergence-free stretching is applied to the previous particle distribution. Due to this stretching, the horizontal distances between the particles are equal to $\sqrt{2}dx$ while the vertical ones are equal to $dx/\sqrt{2}$. There is hence a ratio 2 between these two distances. We can consider it a limiting case: for a ratio 2 or higher, the considered particle of the "second row" belongs to the free-surface, for a lower ratio it does not belong to it. Using the procedure described in [1] an arc of 12^o is not covered by other neighbor circles; hence the particles of the second row are considered as free-surface particles. This occurs also using the proposed algorithm since the *scan region* is now empty. If a fully circular shape was used as *scan region*, the particles of the second row would not be detected as free-surface particles. Other shapes could be adopted as *scan region*. Anyway, in order to correctly detect the free-surface, two requirements have to be satisfied: (a) the detection of circular cavities of diameter equal to $2h$, (b) in the example discussed above the second row of particles in bottom plot of Fig. 16 has to be detected as free-surface.

References

[1] G.A. Dilts, Moving least-squares particle hydrodynamics II: conservation and boundaries, Int. J. Numer. Meth. Eng. 48 (2000) 1503–1524.
[2] H. Aamer, G.A. Dilts, Three-dimensional boundary detection for particle methods, J. Comp. Phys. 226 (2007) 1710–1730.
[3] A.K. Chaniotis, D. Poulikakos, P. Koumoutsakos, Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows, J. Comp. Phys. 182 (2002) 67–90.
[4] P.W. Randles, L.D. Libersky, Smoothed particle hydrodynamics: some recent improvements and applications, Comput. Meth. Appl. Mech. Eng. 139 (1996) 375–408.
[5] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, J. Comp. Phys. 191 (2003) 448–475.
[6] M. Doring, Développement d'une méthode SPH pour les applications à surface libre en hydrodynamique, Ph.D. Thesis, Ecole Centrale Nantes, 2005.
[7] J. Bonet, T.-S.L. Lok, Variational and momentum preservation aspects of smooth particle hydrodynamic formulations, Comput. Meth. Appl. Mech. Eng. 180 (1999) 97–115.
[8] P. Raad, Mitigation of local Tsunami effects, <http://engr.smu.edu/waves/project.html>.
[9] M. Gomez-Gesteira, R.A. Dalrymple, Using a three-dimensional method for wave impact on a tall structure, J. Waterway, Port, Coast. Ocean Eng. 130 (2004) 63–69.
[10] T.P. Fries, H.G. Matthies, Classification and overview of meshfree methods, Informatikbericht 2003-03, Institute of Scientific Computing, Technical University Braunschweig, Brunswick, Germany, 2004.