Journal of Computational Physics 401 (2020) 109000

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Delta Voronoi smoothed particle hydrodynamics, δ-VSPH

David Fernández-Gutiérrez *, Tarek I. Zohdi

Department of Mechanical Engineering, University of California at Berkeley, CA, Berkeley, USA

ARTICLE INFO

Article history:
Received 13 August 2018
Received in revised form 6 August 2019
Accepted 30 September 2019
Available online 3 October 2019

Keywords:
CFD
SPH
Particles
Voronoi
Coupling

ABSTRACT

A Lagrangian scheme that combines Voronoi diagrams with smoothed particle hydrodynamics (SPH) for incompressible flows has been developed. Within the Voronoi tessellation, the Voronoi particle hydrodynamics (VPH) method is used, which is structurally similar to SPH. Two sub-domains are defined based on the proximity to the boundaries. The VPH formulation is used for particles close to solid boundaries, where SPH consistency and implementation of boundary conditions become problematic. Some overlapping of both sub-domains is allowed in order to provide a buffer zone to progressively transition from one method to the other. An explicit weakly compressible formulation for both sub-domains is used, with the diffusive term from the δ-SPH correction extended to the VPH formulation. In addition, the density field is periodically re-initialized and a shifting algorithm is included to avoid excessive deformation of the Voronoi cells. Solid, free-surface, and inlet/outlet boundary conditions are considered. A linear damping term is used during the initialization process to mitigate possible inconsistencies from the user-defined initial conditions. The accuracy of the coupled scheme is discussed by means of a set of well-known verification benchmarks.

1. Introduction

Many of the physical phenomena of interest involve free-surfaces, which are challenging to model when surface-breaking, fragmentation, and phase entrainment occur. Particle methods become attractive for these types of problems because the free-surface boundary condition is intrinsically solved in their formulation, not requiring any special treatment. However, their use is relatively recent [1]. *Smoothed particle hydrodynamics (SPH)* solvers are among the most popular ones in this category [2–6].

The SPH method was originally presented by Lucy [7] and Gingold & Monaghan [8] for astro-dynamical applications. Monaghan [9] applied it for the first time to incompressible free-surface flows, treating the fluid as weakly compressible with a stiff equation of state. The present paper focuses on how SPH models the interaction of the fluid with solid boundaries, which is recognized as one of its main issues. Monaghan [9] proposed discretizing the boundaries into particles with repulsive forces. Libersky et al. [10], Morris et al. [11] and Colagrossi & Landrini [12] use “ghost” particles. These are layers of particles within the solid domain that mirror the fluid domain and whose properties are adjusted to satisfy the boundary conditions. However, the application of this method with complex geometries and extension to 3D problems is still problematic, with various extensions proposed in the literature [13–16]. Other alternatives to deal with solid boundaries explore calculating the intersection of the kernel and the boundary [17], extend the immersed boundary technique used in

* Corresponding author.
E-mail address: dfg82@berkeley.edu (D. Fernández-Gutiérrez).

https://doi.org/10.1016/j.jcp.2019.109000
0021-9991/

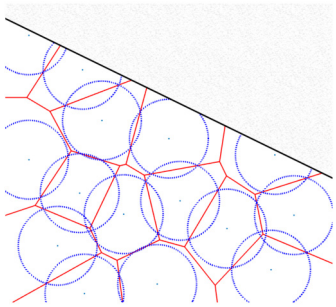


Fig. 1. Voronoi diagram close to the boundary. Continuous lines delimit Voronoi cells, while dotted lines indicate the SPH particle radius.

finite difference schemes [18], and extend the normal flux technique from finite volume schemes [19,20]. For an analysis of the stability and convergence of the method, and of the impact that the treatment of boundary conditions has on them, the reader is referred to the work by Vila, Ben Moussa, Dilts, and their collaborators [21–26]. An alternative approach to performing modifications to the SPH scheme is to couple it with other methods. Recent examples have explored coupling SPH with mesh-based finite volume method solvers [27–30] and have yielded promising results. An additional appealing approach is to couple SPH with another Lagrangian method with some improved characteristics, such as PFEM [31,32].

The application of Voronoi diagrams within the SPH framework has been quite recent. Shobeyri & Ardakani [33] proposed using Voronoi diagrams to improve the computation of the second order derivatives, Ghaffari & Xiao [34] used centroid Voronoi tessellations to adjust the particle positions and improve the consistency of the method, and Chiaki & Yoshida [35] explored using them as part of a particle splitting algorithm. Each Voronoi cell represents the space closer to one particle than to any other [36–39], being a logical choice to link non-overlapping zones to each particle. The idea of a hybrid Lagrangian Voronoi–SPH scheme was first proposed by Barcarolo et al. [40,41]. They used a finite volume formulation for the Voronoi sub-domain and Riemann-SPH for the SPH sub-domain.

However, the fully Lagrangian version of Voronoi particle hydrodynamics, VPH [42,43], emerges as an even more attractive option for this kind of coupling because its formulation is much closer to SPH. VPH evolved from the Voronoi dynamics method created by Serrano & Español [44,45], who developed a meshless scheme that is first-order consistent for the first-order differential operators, regardless of the geometrical distribution of the particles.

This work completes presenting the basis of a coupled scheme between SPH and VPH, initially introduced by Fernandez-Gutierrez et al. [46,47]. Particles close to the solid boundaries are taken as seeds to partition the spatial domain into Voronoi cells (Fig. 1), where the VPH formulation is used. SPH is employed in the rest of the fluid domain, with a transition zone where both methods are combined. This paper first presents the continuous model and the SPH and VPH schemes. The coupling strategy is then discussed in detail. The way boundary conditions are enforced deserves a dedicated analysis that follows next, with special emphasis on the treatment of solid walls. Some modifications are introduced in the formulation published previously [46,47], with significant impact on the stability of the method. This is followed by presenting the implementation of the coupled explicit scheme, including the initialization and time-marching algorithms. The scheme is then tested with some verification cases to evaluate its accuracy and conservation properties. Conclusions and future lines of work are enumerated to close the paper.

2. Continuum model

The scope of this research covers monophasic free-surface viscous liquid flows, governed by the Navier–Stokes equations. Moreover, the expected fluid regimes allow us to accurately model them as barotropic and weakly compressible. In addition, no thermal effects are included, and only uniform gravitational body forces and Newtonian fluids are considered. Based on these assumptions, we can simplify the general Navier–Stokes equations to obtain the final governing equations of the continuum model:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho\,\nabla\cdot\boldsymbol{u}\,, \tag{1}$$

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{g} - \frac{\nabla p}{\rho} + \frac{\lambda+\mu}{\rho}\,\nabla\left(\nabla\cdot\boldsymbol{u}\right) + \nu\,\nabla^2\boldsymbol{u}\,, \tag{2}$$

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{u}\,, \tag{3}$$

where $\mathrm{d}/\mathrm{d}t$ is the Lagrangian time derivative, ρ the fluid density, t the time, $\boldsymbol{u}$ the fluid velocity vector defined as the derivative of the position $\boldsymbol{r}$ of a fluid material point, $\boldsymbol{g}$ the gravity acceleration vector, p the pressure, μ and λ the first and second coefficients of viscosity, and $\nu = \mu/\rho$ the kinematic viscosity. Furthermore, the compressible viscosity term is negligible in the weakly compressible regime for the flows studied in this work [48,49], and it is not further considered. Boundary conditions will be addressed specifically when discussing the numerical methods.

The relaxation from incompressible to weakly compressible is generally followed within the SPH method. By allowing the density to fluctuate due to the particles movement, we can avoid solving the Poisson equation for the pressure field and use an explicit time integration scheme. Instead, a stiff equation of state is used to determine the pressure values. The formulation proposed by Monaghan [9] is followed in this work, which enforces very low density variations and is efficient to compute.

$$p = \frac{c^2 \rho_0}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] + p_0 \,, \tag{4}$$

where c is the speed of the sound, p_0 and ρ_0 are the reference pressure and density of the fluid, respectively, and γ is a coefficient that determines the level of compressibility of the fluid. Typically, $\gamma = 7$. However, the leading order of the Taylor expansion of Eq. (4) around $\rho = \rho_0$ shows that the dependency on c is more relevant than γ. Therefore, we can choose $\gamma = 1$ without a major impact on the pressure fluctuations and obtain a simpler equation of state, as proposed by Antuono et al. [50]:

$$p = c^2 \, (\rho - \rho_0) + p_0 \,. \tag{5}$$

Furthermore, compressibility effects should be physically negligible under the weakly compressible assumption. Therefore, we can modify the value of c with minimum consequences in the flow, as long as Mach number $Ma = u/c << 1$. Given that c is critical to determine the time step for the stability of the explicit scheme, it is common practice to set an upper limit of $Ma = 0.1$, which maintains density fluctuations below 1% ($c^2 = \Delta p / \Delta \rho \rightarrow \Delta \rho / \rho \approx u^2 / c^2$). While the velocity field is unknown a priori, the following considerations are suggested to choose the value of c:

Velocity driven flows. $u_{\max} \approx u_0$

$$c^2 = \frac{\rho u^2}{\Delta \rho} > \frac{u_0^2}{0.01} \rightarrow c > 10\, u_0 \,. \tag{6}$$

Gravity driven flows. $\Delta p \approx \rho g H$, where H is the reference height

$$c^2 = \frac{\rho g H}{\Delta \rho} > \frac{g H}{0.01} \rightarrow c > 10 \sqrt{g H} \,. \tag{7}$$

Pressure driven flows. $\| \nabla p \| \approx (\mathrm{d}p/\mathrm{d}x)_0$

$$\| \nabla p \| \approx \frac{c^2 \Delta \rho}{\Delta x} \quad \rightarrow \quad c^2 = \left(\frac{\mathrm{d}p}{\mathrm{d}x} \right)_0 \frac{\Delta x}{\Delta \rho} > \left(\frac{\mathrm{d}p}{\mathrm{d}x} \right)_0 \frac{\Delta x}{0.01 \rho} \,, \tag{8}$$

where $\| \, \|$ is the 2-norm and Δx is the particle size, which gives the order of magnitude of the particle spacing.

Shear driven flows. $\mu \nabla^2 \boldsymbol{u} \approx \nabla p \rightarrow \mu \, \Delta u / \Delta x^2 \approx \Delta p / \Delta x$

$$\Delta p = c^2 \Delta \rho = \mu \frac{\Delta u}{\Delta x} \quad \rightarrow \quad c^2 = \frac{\mu}{\Delta \rho} \frac{\Delta u}{\Delta x} > \frac{\mu}{0.01 \rho} \frac{\Delta u}{\Delta x} \,. \tag{9}$$

3. Method

3.1. SPH

3.1.1. Fundamentals

The SPH method is known for its flexibility to easily model complex physics and its outstanding conservation properties. The fluid is treated as a continuum medium, with the numerical schemes able to address problems ranging from micrometers to millions of kilometers. Conceptually, the method can be explained as a two-step approximation that involves smoothing and particle approximations. Any field variable f at a given point is approximated as the weighted average of its value at the particles around this point:

$$< f(\boldsymbol{r}) > = \int_{\Omega} f(\boldsymbol{r}^*) \, W(\boldsymbol{r} - \boldsymbol{r}^*, h) \, \mathrm{d}\mathcal{V}^* \approx \sum_b f(\boldsymbol{r}_b) \, W(\boldsymbol{r} - \boldsymbol{r}_b, h) \, \mathcal{V}_b \,, \tag{10}$$

where $< >$ indicates a smoothed value, Ω the fluid domain, $\boldsymbol{r}^*$ a local integration variable, $\mathrm{d}\mathcal{V}^*$ a differential volume, W a finite weighting function known as kernel, h the smoothing length that characterizes the radius of action of the kernel, and b a counter that runs through the particles that fall within the smoothing domain with non-zero W. The particles

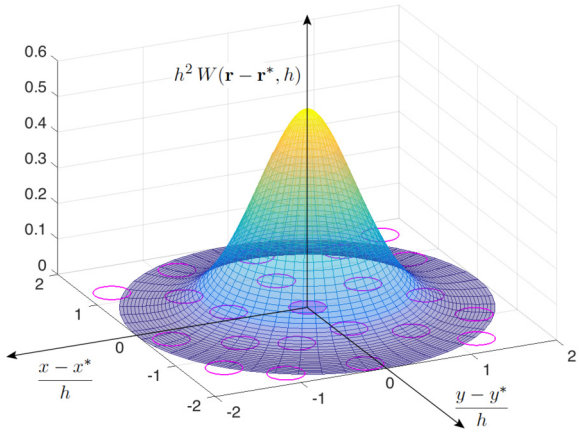


Fig. 2. WC2 $W(\boldsymbol{r}-\boldsymbol{r}^*,h)$ in 2D centered at $\boldsymbol{r}$.

themselves are interpolation nodes, but we can define their associated volume based on the mass m lumped on them as $\mathcal{V}_b = m_b/\rho_b$.

In general, no direction is preferred, so the kernel is expressed as a function of a normalized radial coordinate, $q = \|\boldsymbol{r}-\boldsymbol{r}^*\|/h$, such that

$$W(\boldsymbol{r}-\boldsymbol{r}^*,h) = \frac{K_k}{h^D} M(q)\,, \tag{11}$$

$$\nabla_{\boldsymbol{r}} W(\boldsymbol{r}-\boldsymbol{r}^*,h) = (\boldsymbol{r}-\boldsymbol{r}^*)\frac{K_k}{h^{D+2}\,q}\frac{\mathrm{d}M(q)}{\mathrm{d}q} = (\boldsymbol{r}-\boldsymbol{r}^*)F(q,h)\,, \tag{12}$$

where D is the number of spatial dimensions, and K_k is a constant normalization coefficient whose general form can be written using the Gamma function, Γ, as

$$K_k = \frac{\Gamma\left(\frac{D}{2}\right)}{2\pi^{D/2}\int_\Omega M(q)\,q^{D-1}\mathrm{d}q}\,. \tag{13}$$

The main advantage of the method is the ability to express the spatial derivatives as a function of the field values at the particle locations, without requiring a predefined connectivity between particles [4,6]. As a result, the governing PDEs transform into a set of evolution ODEs used in the SPH sub-domain to compute the numerical solution. In this research, the C2 Wendland kernel (WC2) shown in Fig. 2 is used following Macià et al. [51], which has a $2h$ compact support that corresponds to an average number of neighbor particles of about 50 in 2D.

$$M(q) = \begin{cases} (1.0-0.5\,q)^4\,(1.0+2.0\,q) & q \le 2 \\ 0 & q > 2 \end{cases}\,, \tag{14}$$

$$F(q,h) = \begin{cases} -\dfrac{K_k}{h^{D+2}}\,5.0\,(1.0-0.5q)^3 & q \le 2 \\ 0 & q > 2 \end{cases}\,. \tag{15}$$

To simplify the notation, the common convention in SPH is adopted herein: a for the analyzed particle, b for all particles within the smoothing domain, $< f(\boldsymbol{r}_a) >= f_a$, $f_{ab} = f_a - f_b$, $\overline{f}_{ab} = (f_a+f_b)/2$, $W(\boldsymbol{r}_a-\boldsymbol{r}_b,h) = W_{ab}$, and $\nabla_{\boldsymbol{r}_a} W(\boldsymbol{r}_a-\boldsymbol{r}_b,h) = \nabla_a W_{ab}$.

3.1.2. Artificial viscosity

When no viscous dissipation is included, such as in inviscid flows, the explicit SPH formulation leads to an unstable scheme. Thus, we introduce an artificial viscous component. In this research, we use the Monaghan-Cleary-Gingold's formulation (MCG) discussed by Macià et al. [52]. It conserves linear and angular momentum, vanishes in the limit of $h \to 0$ and for rigid rotations, and is Galilean invariant [2].

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)_a^{AV,\,SPH} = -\sum_b m_b \frac{\alpha\,\overline{h}_{ab}\,\overline{c}_{ab}\,\overline{\rho}_{ab}}{\rho_a\,\rho_b}\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\nabla_a W_{ab}\,, \tag{16}$$

where α is a coefficient used to adjust the magnitude of this artificial viscous term relative to the pressure forces that drive the flow. Typically, $\alpha \approx 0.02$. We can trace this term back to its continuous form following the work by Español & Revenga [53] and Violeau [54], arriving at

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)^{AV} \approx \frac{1}{2(D+2)}\,\alpha\, h\, c\left[\nabla^2\boldsymbol{u} + 2\nabla(\nabla\cdot\boldsymbol{u})\right], \tag{17}$$

which matches the viscous term in the Navier-Stokes momentum Eq. (2) if

$$\frac{1}{2\,(D+2)}\,\frac{\alpha\, h\, c}{\rho} = \mu = \lambda\,. \tag{18}$$

Given that the compressible component in weakly compressible flows is usually negligible, we can also determine an approximation to the Laplacian operator, independent of the kernel used and consistent with Monaghan [2], Hu & Adams [55], and Macià et al. [16].

$$\nabla^2\boldsymbol{u}_a \approx -\sum_b 2(D+2)\, m_b \frac{\overline{\rho}_{ab}}{\rho_a\,\rho_b}\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\nabla_a W_{ab}\,. \tag{19}$$

3.1.3. Diffusive term (δ-correction)

In order to avoid the excessive pressure oscillations, Ferrari et al. [56] considered adding a diffusive term in the smoothed continuity equation, inspired by the work of Vila and Ben Moussa [21,24,26]. This approach was further explored by Molteni & Colagrossi [57] and Antuono et al. [50,58] formally establishing the δ-SPH scheme.

$$< \frac{\mathrm{d}\rho}{\mathrm{d}t} >_a \;= -\rho_a \;< \nabla\cdot\boldsymbol{u} >_a + \left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH}\,. \tag{20}$$

We use as starting point in this work the formulation proposed by Cercos-Pita et al. [59,60] based on its good consistency and conservation properties, and the fact that it does not require tuning parameters. We include an additional term to remove the expected pressure difference from the hydrostatic forces. Furthermore, the original formulation includes a correction coefficient to account for boundaries that can be dropped since no particle in the SPH sub-domain interact with a boundary. As a result,

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-SPH} = -\sum_b \frac{m_b}{\rho_b}\,\frac{\Delta t\,\rho_a}{\rho_0}\left(p_b + \overline{\rho}_{ab}\,\boldsymbol{g}\cdot\boldsymbol{r}_{ab} - p_a\right) F_{ab}\,. \tag{21}$$

Physically, Eq. (21) smooths the natural density field from the pressure jump,[1] effectively suppressing the induced sound wave.

3.1.4. δ-SPH scheme used

The evolution equations of the final δ-SPH scheme used, following Antuono et al. [50,58], are for an a-th particle:

$$\begin{cases} \dfrac{\mathrm{d}\rho_a}{\mathrm{d}t} = \displaystyle\sum_b m_b\,\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}\,F_{ab} \;-\sum_b \frac{m_b}{\rho_b}\frac{\Delta t\,\rho_a}{\rho_0}\left(p_b + \overline{\rho}_{ab}\,\boldsymbol{g}\cdot\boldsymbol{r}_{ab} - p_a\right) F_{ab}\,, \\ \dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = \boldsymbol{g}_a - \dfrac{1}{\rho_a}\displaystyle\sum_b m_b\left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2}\right)\boldsymbol{r}_{ab}\,F_{ab} \;+\sum_b\left(\nu\,2(D+2) + \alpha\,\overline{h}_{ab}\,c\right) m_b\frac{\overline{\rho}_{ab}}{\rho_a\,\rho_b}\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{r}_{ab}}{\|\boldsymbol{r}_{ab}\|^2}\boldsymbol{r}_{ab}\,F_{ab}\,, \\ \dfrac{\mathrm{d}\boldsymbol{r}_a}{\mathrm{d}t} = \boldsymbol{u}_a\,. \end{cases} \tag{22}$$

The artificial viscosity component is only applied to inviscid flows, otherwise it will unnecessarily modify the real fluid viscosity ($\nu \neq 0 \implies \alpha = 0$). The smoothing length may vary slightly from the reference value for each particle so that the number of neighbors remains approximately constant.

$$\left(\frac{h_a}{h_0}\right)^D = \frac{\rho_0}{\rho_a}\,, \tag{23}$$

where h_0 is the reference smoothing length, which depends on the particle size, Δx, and is defined as

$$h_0 = 2\,\Delta x_{a,0} = 2\left(\frac{m_a}{\rho_0}\right)^{1/D}\,. \tag{24}$$

[1] $p_b > p_a \implies \rho_b > \rho_a \rightarrow \rho_a\uparrow,\ \rho_b\downarrow$ to reach equilibrium. Since the kernel function must decrease monotonically, $F_{ab} < 0$ (see Eq. (12)), which yields $(\mathrm{d}\rho/\mathrm{d}t)_a^{\delta-SPH} > 0$ as desired.

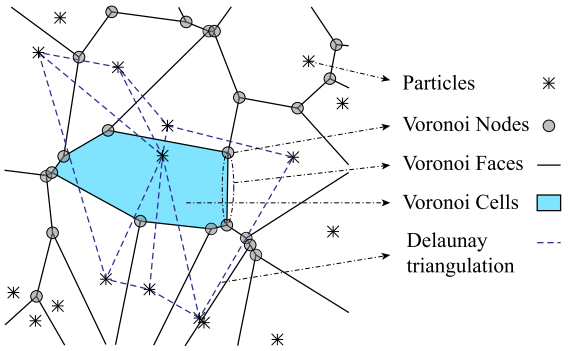


Fig. 3. Voronoi tessellation components.

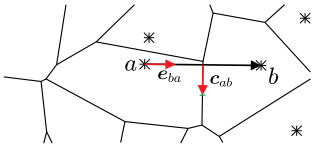


Fig. 4. Section of a Voronoi diagram, showing the vectors $\boldsymbol{e}_{ba}$ and $\boldsymbol{c}_{ab}$.

3.2. VPH

3.2.1. General

In a Voronoi tessellation (dual construction of a Delaunay triangulation), a polyhedral volume is assigned to each particle, which encompasses the space closer to that particle than to any other. It is a pure geometrical construction based only on the location of the particles. The resulting Voronoi tessellation for a D-dimensional space is characterized by its nodes, faces,[2] and cells,[3] as graphically shown in Fig. 3.

Based on this partition of the fluid domain, Serrano & Español [44] devised a methodology to calculate the spatial derivatives, with many similarities to SPH. Later, Hess & Springel [43] applied this concept to solve for fluid flows, formally establishing the basis for *Voronoi particle hydrodynamics* (VPH). Conceptually, it starts by approximating any field value at a particle a location as a volume average on its associated cell, $[f]_a$, such that

$$[f(\boldsymbol{r})]_a = \frac{1}{\mathcal{V}_a} \int_{\mathcal{V}_a} f(\boldsymbol{r}^*)\, \mathrm{d}\mathcal{V}^* , \tag{25}$$

where $\mathcal{V}_a$ is determined from the volume of the Voronoi cell. Similar to SPH, Eq. (25) provides a starting point to evaluate the spatial derivatives. The formal derivation is not trivial, and we refer the reader to the work by Hess & Springel [43]. By applying the divergence theorem to $(\mathbf{1}\cdot\boldsymbol{r})\nabla f(\boldsymbol{r})$, where $\mathbf{1}$ indicates an arbitrary constant unit vector, and assuming sufficiently smooth fields and particles close to the centroid of the cells, they arrive at

$$\nabla f_a = \frac{1}{\mathcal{V}_a} \sum_{b\neq a} A_{ab}\, f_{ab} \left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right] , \tag{26}$$

where the neighboring particles b in VPH are defined as those with common cell faces with particle a, A_{ab} is the area of cell face between particles a and b, $\boldsymbol{e}_{ab}$ is the unit vector pointing from b to a, and $\boldsymbol{c}_{ab}$ is a vector going from the midpoint between a and b to the centroid of their associated face, as shown in Fig. 4. It is stressed here that we do not follow the notation used by Hess & Springel [43]. Instead, for consistency, we follow the same simplified convention of SPH (Sec. 3.1.1), with $f_a = f(\boldsymbol{r}_a) = [f(\boldsymbol{r})]_a$ and $\nabla f_a = \nabla f(\boldsymbol{r}_a) = [\nabla f(\boldsymbol{r})]_a$ in this case.

We must highlight that Serrano & Español [44] demonstrated that Eq. (26) is exact for constant and linear fields, independently of the particle locations, which is a significant improvement from the SPH approach.

As in SPH, the Laplacian operator of a vector field is approximated following an alternative approach. Starting with the general volume averaged expression and applying directly the divergence theorem in this case,

[2] Each face represents the contact surface between two neighbor particles. It is orthogonal to the vector connecting both particles, and placed half-way between them. Geometrically, it is a simplex, *i.e.* a D-dimensional triangle, and therefore consists always of a set of D nodes.

[3] Each cell can have different number of associated faces delimiting it.

$$\left[\nabla^2 \boldsymbol{f}(\boldsymbol{r})\right]_a = \frac{1}{\mathcal{V}_a}\int_{\mathcal{V}_a} \nabla^2 \boldsymbol{f}\, \mathrm{d}\mathcal{V}^* = \frac{1}{\mathcal{V}_a}\sum_{b\neq a}\int_{\mathcal{S}_{ab}} \nabla \boldsymbol{f}\cdot \boldsymbol{e}_{ba}\, \mathrm{d}\mathcal{S}^*\,, \tag{27}$$

where $\mathcal{S}$ indicates the face surfaces that enclose the cell. We can interpret the dot product inside the integral as the directional derivative of f. Using a linear approximation, we obtain the following estimate for the Laplacian operator:

$$\nabla^2 \boldsymbol{f}_a = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\boldsymbol{f}_{ab}}{\|\boldsymbol{r}_{ab}\|}\,. \tag{28}$$

The order of this formula is not clearly established in literature. Serrano [45] used it to simulate a shear stationary flow, showing that the schemes display extra dissipation when particles are in disordered configurations while it renders accurate results for orderly ones. For an extensive discussion on the properties of the Laplacian operator in Voronoi and Delaunay tessellations, the reader is referred to Duque et al. [61] and references therein.

3.2.2. Artificial viscosity

Same as in the SPH scheme, an artificial viscosity term is necessary for inviscid problems to achieve stability. Starting from Eq. (17), we only need to determine an expression for the $\nabla(\nabla\cdot\boldsymbol{u})$ component to be consistent with SPH. We can apply directly the divergence theorem again leading to

$$[\nabla(\nabla\cdot\boldsymbol{u})]_a = \frac{1}{\mathcal{V}_a}\int_{\mathcal{V}_a}\nabla\left(\nabla\cdot\boldsymbol{u}\right)\,\mathrm{d}\mathcal{V}^* = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a}\int_{\mathcal{S}_{ab}}\left(\nabla\cdot\boldsymbol{u}\right)\boldsymbol{e}_{ab}\,\mathrm{d}\mathcal{S}^*\,. \tag{29}$$

Physically, we can interpret the divergence of the velocity on the face $\mathcal{S}_{ab}$ as its contribution to the change of volume of the cell and approximate it as

$$\left(\nabla\cdot\boldsymbol{u}\right)_{\mathcal{S}_{ab}} = \left(-\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_{ab} = \left(\frac{1}{\mathcal{V}}\frac{\mathrm{d}\mathcal{V}}{\mathrm{d}t}\right)_{ab} \approx \frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{\|\boldsymbol{r}_{ab}\|}\,, \tag{30}$$

which yields the following expression for the artificial viscosity term under the VPH scheme:

$$\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}\right)_a^{AV,VPH} = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\alpha\,\overline{h}_{ab}\,c}{2(D+2)}\left[\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} + 2\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{\|\boldsymbol{r}_{ab}\|}\,\boldsymbol{e}_{ab}\right]. \tag{31}$$

3.2.3. Diffusive terms

The density can be estimated from the Voronoi cell. However, it is more beneficial to use the continuity equation to evolve it in time, so we can include a diffusive term to reduce the high-order pressure oscillations as discussed in SPH (Sec. 3.1.3). The same formulation used in the δ-SPH scheme can be extended to the VPH, although no previous reference in the literature was found. Comparing the SPH and VPH formulations, we can extract the following analogy:

$$-\sum_b \frac{m_b}{\rho_b}\left(\ldots\right)\nabla_a W_{ab} \qquad \leftrightarrow \qquad \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\left(\ldots\right)\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right], \tag{32}$$

which applied to the diffusive term used in SPH (Eq. (21)) leads to

$$\left(\frac{\mathrm{d}\rho}{\mathrm{d}t}\right)_a^{\delta-VPH} = -\frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\Delta t\,\rho_a}{\rho_0}\frac{\left(p_b + \overline{\rho}_{ab}\,\boldsymbol{g}\cdot\boldsymbol{r}_{ab} - p_a\right)}{2\,\|\boldsymbol{r}_{ab}\|}\,. \tag{33}$$

3.2.4. δ-VPH scheme used

The evolution equations of the final δ-VPH scheme, equivalent to the δ-SPH from Sec. 3.1.4 and following Serrano & Español [44] and Hess & Springel [43] in this case, are for an a-th particle:

$$\begin{cases} \dfrac{\mathrm{d}\rho_a}{\mathrm{d}t} = -\dfrac{\rho_a}{\mathcal{V}_a}\displaystyle\sum_{b\neq a} A_{ab}\boldsymbol{u}_{ab}\cdot\left[\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right] + \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\Delta t\,\rho_a}{\rho_0}\frac{\left(p_b + \overline{\rho}_{ab}\boldsymbol{g}\cdot\boldsymbol{r}_{ab} - p_a\right)}{2\,\|\boldsymbol{r}_{ab}\|}, \\ \dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t} = \boldsymbol{g}_a + \dfrac{1}{\rho_a\,\mathcal{V}_a}\displaystyle\sum_{b\neq a} A_{ab}\left[\left(p_a + p_b\right)\frac{\boldsymbol{e}_{ab}}{2} + \left(p_a - p_b\right)\frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right] \\ \qquad\qquad -\dfrac{\nu}{\mathcal{V}_a}\displaystyle\sum_{b\neq a} A_{ab}\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} - \frac{1}{\mathcal{V}_a}\sum_{b\neq a} A_{ab}\frac{\alpha\,\overline{h}_{ab}\,c}{2(D+2)}\left[\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} + 2\,\frac{\boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}}{\|\boldsymbol{r}_{ab}\|}\,\boldsymbol{e}_{ab}\right], \\ \dfrac{\mathrm{d}\boldsymbol{r}_a}{\mathrm{d}t} = \boldsymbol{u}_a\,, \end{cases} \tag{34}$$

with $\nu \neq 0 \implies \alpha = 0$ as before, and $\mathcal{V}_a$ and A_{ab} computed from the Voronoi tessellation.

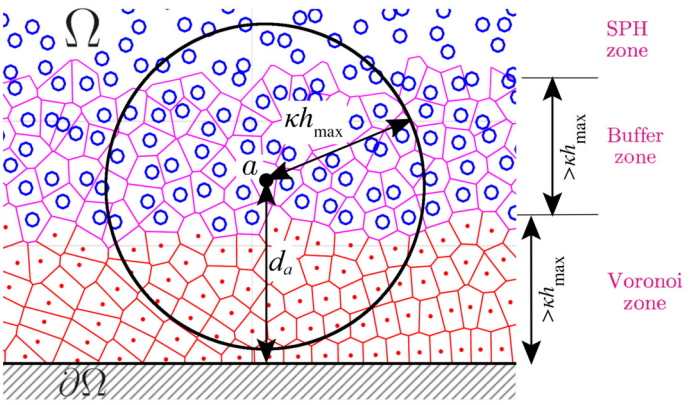


Fig. 5. Sketch with definition of Voronoi, SPH and buffer zones.

3.3. Coupling

3.3.1. SPH & VPH sub-domains

The SPH and VPH schemes described in the previous sections are very similar, and their coupling arises as a natural way of capturing the advantages of each one. VPH reduces the interaction between particles to the immediate neighbors with shared faces. This is beneficial to enforce the boundary conditions if we are able to map exactly the boundaries with cell faces. On the other hand, SPH manages efficiently free-surfaces with large deformations. Thus, we define a VPH and an SPH sub-domain based on the distance to the closest boundary, d, with some overlapping between both as a buffer zone to allow for a smooth transition between methods as shown in Fig. 5.

The width of the Voronoi zone, l_{Vor}, is such that all particles outside it are at a distance to the closest solid boundary larger than $\kappa h_{\max}$, with κ being the factor that characterizes the support radius of the kernel ($\kappa = 2$ for the C2 Wendland kernel) and $h_{\max}$ the largest smoothing length. Consequently, **we do not need to account for solid boundary conditions in the SPH formulation**. The thickness of the buffer zone, l_{buffer}, is also at least $\kappa h_{\max}$, so no particle in the SPH zone interacts with a particle in the Voronoi zone. Finally, the SPH zone is simply the rest of the fluid domain.

At each time step, we identify each particle as Voronoi, buffer, or SPH based on the zone where it lays, which depends on the position of the boundaries at that time step. We should stress that the particles themselves are just interpolation nodes where physical variables are evaluated (p, ρ, $\boldsymbol{u}$, $\mathrm{d}\rho/\mathrm{d}t$, etc.), and are not linked to any formulation. In this way, a "Voronoi particle" can become a "buffer particle" as it moves away from the boundary, and once it crosses the buffer become an "SPH particle", and vice versa.

3.3.2. Fields in the buffer zone

In the buffer zone, fields are evaluated using both SPH and VPH and need to be combined. Therefore, the value of a certain field A at a particle a when it is in the buffer is defined as:

$$A_a := \omega_a \, A_a^{SPH} + (1 - \omega_a) \, A_a^{VPH}, \tag{35}$$

where A_a^{SPH} and A_a^{VPH} are the field values obtained through the SPH and VPH interpolations, respectively, and ω is a weight defined so that the coupling is $\mathcal{C}^1$. We must stress that this condition is a first best guess, inspired by the h^2 order of the SPH interpolation studied in detailed by Quinlan et al. [62]. However, other alternatives are of course possible. To this aim, the value A_a should tend to A_a^{VPH} or A_a^{SPH} as the particle approaches to the Voronoi or SPH zones, respectively. A factor q_a is defined using the distance d_a between the particle a and the closest solid wall:

$$q_a := \frac{d_a - \kappa h_{\max}}{l_{\text{buffer}}}, \tag{36}$$

and ω_a is the output of a third-grade polynomial in q_a, which allows us to have the required $\mathcal{C}^1$ class for the coupling scheme in Eq. (35):

$$\omega_a := q_a^2 \, (3 - 2\, q_a) \, . \tag{37}$$

We have seen in the SPH and VPH expressions for the differential operators how we are looking at the interaction of a particle a with its neighbors b. Since each particle has a different weight, the remaining question is which one to use. Two possible approaches are considered:

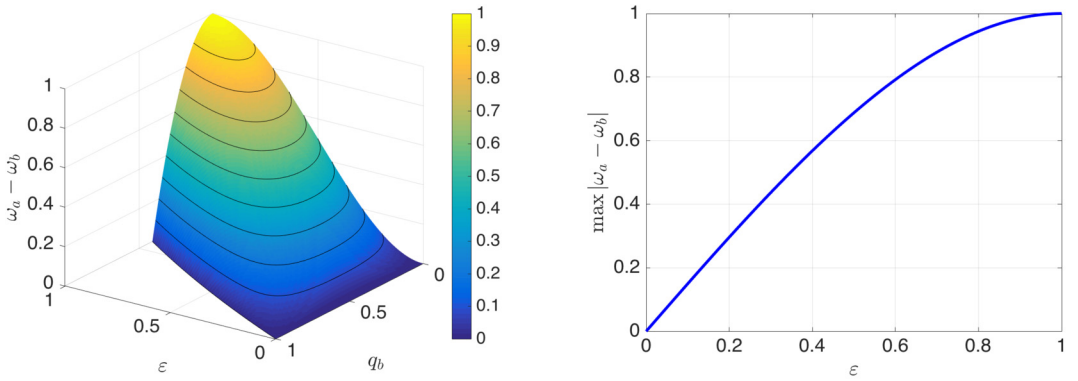


Fig. 6. Upper bound of weight differences between neighbor particles.

1. Calculate the interaction with all neighbors and apply the weight in the final value of the derivative,

$$\nabla A_a = \omega_a \nabla A_a^{SPH} + (1-\omega_a)\nabla A_a^{VPH}, \tag{38}$$

2. Average weights in the local interaction between particles a and b,

$$\nabla A_a = \sum_{b\neq a} \nabla A_{ab} = \sum_{b\neq a} \overline{\omega}_{ab} \nabla A_{b\to a}^{SPH} + (1-\overline{\omega}_{ab})\nabla A_{b\to a}^{VPH}. \tag{39}$$

The first option seems simpler, but is not conservative since the force on a due to b is different than on b due to a, even if the formulation is pair-wise antisymmetric in SPH and VPH. Looking at the pressure force as an example, $\boldsymbol{f}_{p,\,b\to a}$ and $\boldsymbol{f}_{p,\,a\to b}$ should cancel to conserve momentum, but due to the different weights it is not the case,

$$\Delta \boldsymbol{f}_{ab} = \boldsymbol{f}_{p,\,b\to a} + \boldsymbol{f}_{p,\,a\to b} = \frac{m_a}{\rho_a}(\omega_a - \omega_b)\left(\nabla p_{b\to a}^{SPH} - \nabla p_{b\to a}^{VPH}\right) \neq 0. \tag{40}$$

The second option is fully pair-wise antisymmetric, and therefore, conservative overall. However, the individual components from SPH and VPH are unbalanced due to the different weight for each neighbor. We can interpret this fact in SPH as a modification of the kernel which is no longer a symmetric function. Neighbors further from the boundary have a larger impact with no physical reason. Likewise, the neighbor contribution in VPH is not just linked to the face area, becoming more relevant those neighbors closer to the boundary.

Ideally, these deviations should cancel each other. However, this is not the case due to the different characteristic lengths of VPH (Δx) and SPH (κh). Consequently, the number of neighbors involved is different, with different individual contributions in each method. We can quantify the order of this deviation, $\mathcal{O}(\Delta\nabla A_a)$, by subtracting both approaches

$$\mathcal{O}(\Delta\nabla A_a) = \left| \sum_{b:\mathrm{SPH}} \frac{\omega_a - \omega_b}{2} \nabla A_{b\to a}^{SPH} - \sum_{b:\mathrm{VPH}} \frac{\omega_a - \omega_b}{2} \nabla A_{b\to a}^{VPH} \right|. \tag{41}$$

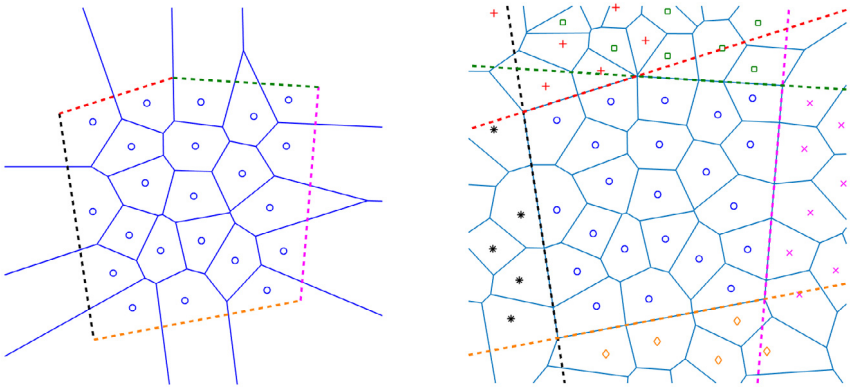
Based on the formulation used for the weight, $\max|\omega_a - \omega_b|_{\mathrm{VPH}} \leq |\omega_a - \omega_b|_{\mathrm{SPH}}$, since the possible maximum distances between SPH neighbors are equal or larger than in VPH ($\kappa h_{\max} > \Delta x$). We can, therefore, determine an upper bound to this deviation as

$$\mathcal{O}(\Delta\nabla A_a) \leq \frac{\max|\omega_a - \omega_b|_{\mathrm{SPH}}}{2} \overbrace{\left[\sum_{b:\mathrm{SPH}} \nabla A_{b\to a}^{SPH} - \sum_{b:\mathrm{VPH}} \nabla A_{b\to a}^{VPH} \right]}^{\mathcal{O}((\kappa h)^2)}. \tag{42}$$

We can take the kernel compact support radius as an upper bound for the maximum difference in the distance to the boundary between a and b, such that $|q_b - q_a| \leq \kappa h_{\max}/l_{\mathrm{buffer}}$. Naming $\varepsilon = \kappa h_{\max}/l_{\mathrm{buffer}}$, we can use Eqs. (36)-(37) to determine the maximum expected weight difference for the coupling scheme for a given ε as

$$\max|\omega_a - \omega_b|_{\mathrm{SPH}} \leq \frac{\varepsilon}{2}\left(3 - \varepsilon^2\right), \tag{43}$$

with $\varepsilon \in [0,1]$, $q_b \in [0,1]$, and $q_b + \varepsilon \leq 1$ as shown in Fig. 6. If this approach is followed to compute the derivatives, we should limit this deviation to at least an order of magnitude lower than the order of the method, $\mathcal{O}\left((\kappa h)^2\right)$, to avoid loosing accuracy. From Eq. (42), this condition implies $\max|\omega_a - \omega_b|_{\mathrm{SPH}} \leq 0.2$. We achieve it by adjusting the buffer length to limit the maximum weight variations, which, to the leading order, yields $l_{\mathrm{buffer}} \geq 7.5\,\kappa h_{\max}$ as the minimum buffer length.



(a) Unbounded cells (b) Bounded cells

Fig. 7. Voronoi diagrams in the presence of convex boundaries.

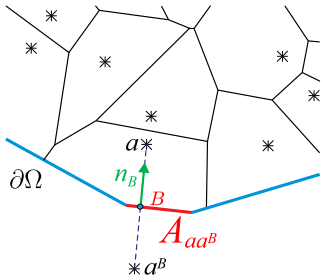


Fig. 8. Section of a Voronoi diagram close to a solid boundary.

3.4. Boundary conditions (BCs)

3.4.1. Convex solid boundaries

Under the coupled scheme, only Voronoi particles interact with solid boundaries. Consequently, the solid BCs need to be imposed only within the VPH formulation. The main challenge lies on achieving that the boundaries become the actual faces of well-defined Voronoi cells. First, all surfaces bounding the fluid domain are discretized into a set of $D-1$ dimensional triangles so they are always flat surfaces. The main topological limitation is that Voronoi cells are always convex. We must then distinguish between convex and concave adjacent boundary faces, since only the convex ones can be overlapped with "pure" Voronoi cell faces. Concave adjacent faces require a different treatment, discussed in Sec. 3.4.2.

Voronoi diagrams are unbounded by definition, representing a partition of the complete spatial domain. Those particles physically in contact with the boundary will result in Voronoi cells that extend beyond the fluid domain. We can bound them simply by noticing that in a symmetric configuration of particles, the Voronoi cells should be symmetric also. Therefore, the symmetry plane should be mapped with cell faces since it cannot be crossed. Taking advantage of this fact, the boundary faces are converted into symmetry planes, and effectively become part of the tessellation, by creating a set of points that reflect the particles in the boundary nearby, as shown in Fig. 7.

Unlike ghost particles in SPH, these reflected points are used only to create the Voronoi tessellation. We will refer to them as boundary virtual points, and identify them with a superscript ^{B}, as displayed in Fig. 8. Each boundary virtual point is associated with one particle and one boundary face, so when it is identified as a neighbor we know the type of solid BC to be applied.

Once the solid boundaries become part of the perimeter of the Voronoi cells, the BCs can be enforced on the related terms in the VPH formulation (Sec. 3.2.4). This is done by expressing the required field values at a^B as functions of the known physical properties at the boundaries. We recall that linear approximations were used to derive the VPH differential operators under the assumption of smooth fields (Sec. 3.2.1). Under the same assumption, we arrive at the following relations:

Boundary velocity. $\boldsymbol{u}_B$. Two alternatives can be applied:

1. No-slip BC: the flow velocity tends to the boundary velocity as a fluid element approaches to the boundary. To achieve this:

$$\boldsymbol{u}_{a^B} = 2\,\boldsymbol{u}_B - \boldsymbol{u}_a\,, \tag{44}$$

$$\boldsymbol{u}_{aa^B} = \boldsymbol{u}_a - \boldsymbol{u}_{a^B} = 2(\boldsymbol{u}_a - \boldsymbol{u}_B) = 2\,\boldsymbol{u}_{aB}\,. \tag{45}$$

2. Free-slip BC: the normal velocity tends to the boundary normal velocity as a fluid element approaches the boundary. Eq. (44) is then projected on the boundary normal $\boldsymbol{n}_B = \boldsymbol{e}_{aa^B}$, while the tangential component remains unaltered. $\boldsymbol{u}_{aa^B}$ should then be aligned with the normal vector, so

$$\boldsymbol{u}_{aa^B} = (2\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B)\,\boldsymbol{n}_B \implies \begin{cases} \boldsymbol{u}_{aa^B}\cdot\boldsymbol{e}_{aa^B} = 2\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B\,, \\ \boldsymbol{u}_{aa^B}\cdot\boldsymbol{c}_{aa^B} = 0\,. \end{cases} \tag{46}$$

Boundary pressure. p_B. The pressure gradient is obtained by projecting the momentum equation on the boundary normal to enforce the following Neumann condition [14]:

$$\frac{\partial p}{\partial \boldsymbol{n}_B} = \rho\left[\boldsymbol{g}\cdot\boldsymbol{n}_B - \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\boldsymbol{n}_B + \nu\,\nabla^2\boldsymbol{u}|_B\cdot\boldsymbol{n}_B\right], \tag{47}$$

with the last term being generally negligible. Noting that $\boldsymbol{n}_B$ is oriented towards the fluid, and using the centroid of the cell face as the representative point to account for hydrostatic correction, we approximate the constant p_B at the face by linear extrapolation:

$$p_B = p_a - \frac{\partial p}{\partial \boldsymbol{n}_B}\frac{\|\boldsymbol{r}_{aa^B}\|}{2} + \rho_a \boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\,, \tag{48}$$

which leads to express the pressure at the boundary virtual point as

$$p_{a^B} = 2p_B - p_a = p_a - \rho_a\left[\left(\boldsymbol{g} - \frac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right)\cdot\boldsymbol{r}_{aa^B} - 2\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\right]. \tag{49}$$

Using these relations on the corresponding elements in the VPH evolution equations, given in Eq. (34), we arrive at the following contributions from the solid boundaries to the time derivatives:

No-slip BC. $\boldsymbol{u}_{aa^B} = 2(\boldsymbol{u}_a - \boldsymbol{u}_B) = 2\,\boldsymbol{u}_{aB}$,

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = -\dfrac{\rho_a}{\mathcal{V}_a}A_{aa^B}\,\boldsymbol{u}_{aa^B}\cdot\dfrac{\boldsymbol{n}_B}{2} + \dfrac{1}{\mathcal{V}_a}A_{aa^B}\,\dfrac{\Delta t\,\rho_a}{\rho_0}\,\dfrac{\rho_a}{2}\,\dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\boldsymbol{n}_B\,, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = \dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aa^B}\left[\rho_a\left(\left(\boldsymbol{g} - \dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right)\cdot\boldsymbol{r}_{aa^B} - 2\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\right)\left(\dfrac{\boldsymbol{n}_B}{2} - \dfrac{\boldsymbol{c}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\right) + p_a\,\boldsymbol{n}_B\right] \\ \qquad\qquad - \dfrac{\nu}{\mathcal{V}_a}A_{aa^B}\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|} - \dfrac{1}{\mathcal{V}_a}A_{aa^B}\dfrac{\alpha\,h_a\,c}{2(D+2)}\left[\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|} + 2\left(\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\cdot\boldsymbol{n}_B\right)\boldsymbol{n}_B\right]. \end{cases} \tag{50}$$

Free-slip BC. $\boldsymbol{u}_{aa^B} = (2\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B)\,\boldsymbol{n}_B$,

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = -\dfrac{\rho_a}{\mathcal{V}_a}A_{aa^B}\,\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B + \dfrac{1}{\mathcal{V}_a}A_{aa^B}\,\dfrac{\Delta t\,\rho_a}{\rho_0}\,\dfrac{\rho_a}{2}\,\dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\cdot\boldsymbol{n}_B\,, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{a^B\to a} = \dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aa^B}\left[\rho_a\left(\left(\boldsymbol{g} - \dfrac{\mathrm{d}\boldsymbol{u}_{\boldsymbol{B}}}{\mathrm{d}t}\right)\cdot\boldsymbol{r}_{aa^B} - 2\boldsymbol{g}\cdot\boldsymbol{c}_{aa^B}\right)\left(\dfrac{\boldsymbol{n}_B}{2} - \dfrac{\boldsymbol{c}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\right) + p_a\,\boldsymbol{n}_B\right] \\ \qquad\qquad - \dfrac{\nu}{\mathcal{V}_a}A_{aa^B}\dfrac{\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|} - \dfrac{1}{\mathcal{V}_a}A_{aa^B}\dfrac{\alpha\,h_a\,c}{2(D+2)}\,\dfrac{3\,\boldsymbol{u}_{aa^B}}{\|\boldsymbol{r}_{aa^B}\|}\,. \end{cases} \tag{51}$$

Note that no $\boldsymbol{c}_{aB}$ term appears on the components linked to the $\nabla\cdot\boldsymbol{u}$ in the continuity equation under the no-slip condition. Even if the boundary and particle velocities are different, the virtual boundary particle created moves tangentially with the particle as otherwise will erroneously deform the boundary and lead to instabilities. Thus, the boundary tangential velocity does not induce any volume change.

3.4.2. Concave solid boundaries

When a particle is in the proximity of two or more adjacent concave faces, the previous approach of simply adding extra points is not feasible. In fact, if we add the boundary virtual points, the resulting symmetry planes would intersect the cell as shown in Fig. 9, which is clearly erroneous.

Therefore, when a particle is near two or more concave boundary faces, it does not generate any boundary virtual points. Instead, the resulting convex Voronoi cell is directly trimmed with the boundary after the tessellation is generated. The intersection algorithm implemented proceeds as follows:

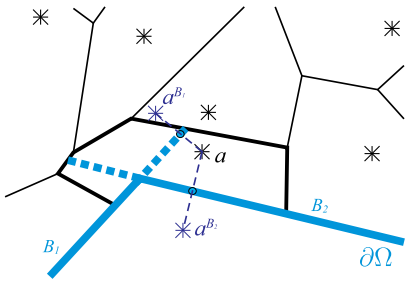
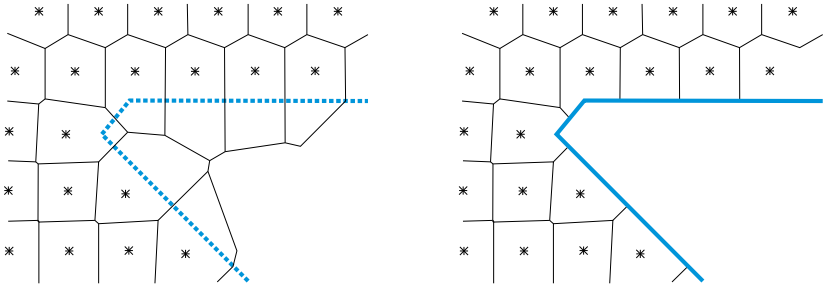


Fig. 9. Concave cell with boundary virtual nodes.



(a) Untrimmed cells (b) Trimmed cells

Fig. 10. Voronoi diagram trimmed with concave boundary.

1. Checks if a particle is in the proximity of boundaries but has no boundary virtual points to identify it as a *concave boundary particle*. If not, no trimming is required and goes to the next particle.
2. Extracts the faces shared with neighbors that are also concave boundary particles, as those are the ones that need to be trimmed.
3. Faces linked to neighbors with lower index have been trimmed already, so directly imports these faces.
4. Determines which cell nodes are within the fluid domain, and if a boundary node falls within the cell. The maximum particle size is limited to ensure that only one boundary node can be covered by a cell.
5. Removes cell faces with all nodes outside the fluid domain.
6. With the remaining cell faces, it determines the intersection points between cell edges and boundary faces. We define edges as segments connecting two nodes in one face.
7. Based on the intersection points found and the possible boundary node within the cell, it creates the new faces on the boundary that close the cell. Each resulting face should still have only D nodes, so more than one face can be generated on the original cell face/boundary planes.
8. Updates the trimmed faces in the tessellation data, and adds the new faces and nodes found.
9. Updates the linked faces to the trimmed cell, and begins in step 1 with the next particle.

The resulting tessellation includes degenerate cells with concave faces, but maps correctly boundaries as shown in Fig. 10.

The next step is enforcing properly the BCs on the concave faces. First, we should acknowledge that the fluid particle within the cell can be on either side of the plane containing the concave boundary. In fact, as the particle moves, it can cross this plane without crossing or even being close to the boundary itself (Fig. 11). Mathematically, this implies that the formulation used for convex boundaries cannot be used directly, since it is possible that $\|\boldsymbol{r}_{aa^B}\| \to 0$, which would lead to a singularity.

Using the minimum distance to the surrounding boundaries, d_a, and the orthogonal distance to the boundary B, $d_{a\perp B} = (\boldsymbol{r}_a - \boldsymbol{x}_1)\cdot\boldsymbol{n}_B$, where $\boldsymbol{x}_1$ is the position vector of the closest node of the face, we can group the possible locations of the particle within the cell into three main categories (Fig. 12). Based on this subdivision, the following generalized formulation is proposed, tailored for each category:

$\underline{d_{a\perp B} \geq d_a}$ Particle on the side of the boundary towards the fluid, since the nodes of the face are ordered so $\boldsymbol{n}_B$ is oriented always towards the fluid. Furthermore, it falls within the orthogonal projection of the boundary so it is feasible to use the convex formulation already derived by simply creating a boundary virtual point at

$$\boldsymbol{r}_{a^B} = \boldsymbol{r}_a - 2\,d_{a\perp B}\,\boldsymbol{n}_B \implies \begin{cases} \boldsymbol{r}_{aa^B} = 2\,d_{a\perp B}\,\boldsymbol{n}_B\,, \\ \|\boldsymbol{r}_{aa^B}\| = 2\,d_{a\perp B}\,. \end{cases} \tag{52}$$

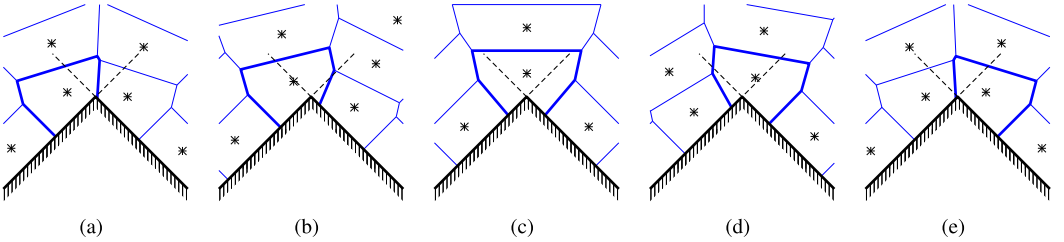


Fig. 11. Possible scenarios of a fluid particle next to a concave boundary.

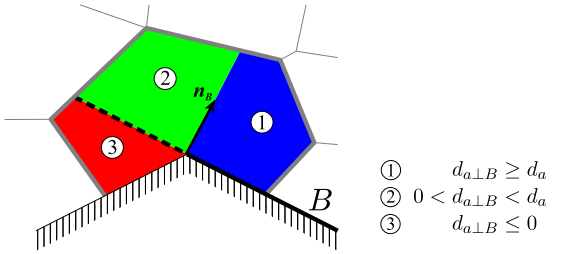


Fig. 12. Zones within the cell into which we can categorize the position of the particle, based on d_a and $d_{a\perp B}$.

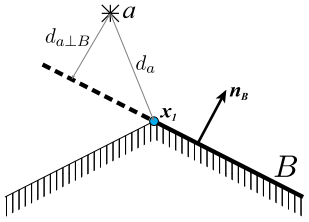


Fig. 13. Representative point in concave faces.

$\underline{0 < d_{a\perp B} < d_a}$ Still on the fluid side, but at risk of approaching the concave boundary plane without approaching the boundary face. We can revise the formulation for concave boundaries by realizing that the mid-point between particles, used for the general VPH scheme, is not the representative point in this case. A more logical option is choosing the closest boundary node to the particle, $\boldsymbol{x}_1$, as shown in Fig. 13, to get a better estimate of the derivative:

$$\nabla f(\boldsymbol{r}^*) \approx \frac{f(\boldsymbol{r}_a) - f(\boldsymbol{x}_1)}{\|\boldsymbol{r}_a - \boldsymbol{x}_1\|} \frac{\boldsymbol{r}_a - \boldsymbol{x}_1}{\|\boldsymbol{r}_a - \boldsymbol{x}_1\|} = \frac{f_{ax_1}}{d_a} \frac{\boldsymbol{r}_{ax_1}}{d_a} . \tag{53}$$

Note that if we reached this case, the fluid particle falls outside of the orthogonal projection of the boundary. The distance from the particle to $\boldsymbol{x}_1$ is precisely the minimum distance from the particle to the boundary, d_a. Furthermore, its projection to $\boldsymbol{n}_B$ should be precisely the orthogonal distance, $d_{a\perp B}$. Using this new approach in the expressions used to derive the general VPH differential operators, we observe how

$$f_{ab} \left(\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|} \right) \quad \rightarrow \quad f_{ax_1} \frac{d_{a\perp B}}{d_a^2} \left(d_{a\perp B} \boldsymbol{n}_B - \boldsymbol{c}_{aB} \right) . \tag{54}$$

Note that both formulations converge as $d_{a\perp B} \rightarrow d_a$, since $f_{ax_1} \leftrightarrow f_{ab}/2$ and $d_{a\perp B} \leftrightarrow \|\boldsymbol{r}_{ab}\|/2$, but now does not become singular as $d_{a\perp B} \rightarrow 0$. In fact, since this formulation is more conservative, it is feasible to extend it to convex boundaries and have a unified formulation for all solid boundaries. When applied to the directional derivative used to evaluate the second order differential operators,

$$\frac{\boldsymbol{u}_{ab}}{\|\boldsymbol{r}_{ab}\|} = \nabla \boldsymbol{u} \cdot \boldsymbol{e}_{ab} \rightarrow \frac{\boldsymbol{u}_{ax_1}}{d_a} \frac{\boldsymbol{r}_{ax_1}}{d_a} \cdot \boldsymbol{n}_B = \frac{d_{a\perp B}}{d_a^2} \boldsymbol{u}_{ax_1} , \tag{55}$$

which is directly applied to the Laplacian and the gradient of the divergence operators in Eq. (28) and Eq. (30), respectively. Finally, the pressure at the boundary in Eqs. (47)-(49) and the diffusive term from the δ-VPH correction in Eq. (33) need to be modified:

$$p_B = p_a - \rho_a\, \boldsymbol{g}\cdot(d_{a\perp B}\,\boldsymbol{n}_B - \boldsymbol{c}_{aB}) + d_{a\perp B}\,\frac{\mathrm{d}\boldsymbol{u_B}}{\mathrm{d}t}\cdot\boldsymbol{n}_B\,, \tag{56}$$

$$\frac{\left(p_b + \overline{\rho}_{ab}\,\boldsymbol{g}\cdot\boldsymbol{r}_{ab} - p_a\right)}{\|\boldsymbol{r}_{ab}\|}\,\boldsymbol{e}_{ab}\cdot\left(\frac{\boldsymbol{e}_{ab}}{2} - \frac{\boldsymbol{c}_{ab}}{\|\boldsymbol{r}_{ab}\|}\right) \to \rho_a\,\frac{d_{a\perp B}^3}{d_a^3}\,\frac{\mathrm{d}\boldsymbol{u_B}}{\mathrm{d}t}\cdot\boldsymbol{n}_B\,. \tag{57}$$

Again, we can use these new expressions in Eq. (34) to obtain the final contributions from the concave boundaries to the time derivatives:

No-slip BC. $\boldsymbol{u}_{ax_1} = \boldsymbol{u}_a - \boldsymbol{u}_B = \boldsymbol{u}_{aB}$,

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} = -\dfrac{\rho_a}{\mathcal{V}_a}A_{aB}\,\dfrac{d_{a\perp B}^2}{d_a^2}\,\boldsymbol{u}_{ax_1}\cdot\boldsymbol{n}_B + \dfrac{1}{\mathcal{V}_a}A_{aB}\,\dfrac{\Delta t\,\rho_a^2}{\rho_0}\,\dfrac{d_{a\perp B}^3}{d_a^3}\,\dfrac{\mathrm{d}\boldsymbol{u_B}}{\mathrm{d}t}\cdot\boldsymbol{n}_B\,, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} = \dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aB}\left[\rho_a\left(d_{a\perp B}\left(\boldsymbol{g} - \dfrac{\mathrm{d}\boldsymbol{u_B}}{\mathrm{d}t}\right)\cdot\boldsymbol{n}_B - \boldsymbol{g}\cdot\boldsymbol{c}_{aB}\right)\dfrac{d_{a\perp B}}{d_a^2}\,(d_{a\perp B}\,\boldsymbol{n}_B - \boldsymbol{c}_{aB}) + p_a\,\boldsymbol{n}_B\right] \\ \qquad\qquad - \dfrac{\nu}{\mathcal{V}_a}A_{aB}\,\dfrac{d_{a\perp B}}{d_a^2}\,\boldsymbol{u}_{ax_1} - \dfrac{1}{\mathcal{V}_a}\,A_{aB}\,\dfrac{\alpha\,h_a\,c}{2(D+2)}\,\dfrac{d_{a\perp B}}{d_a^2}\left[\boldsymbol{u}_{ax_1} + 2\left(\boldsymbol{u}_{ax_1}\cdot\boldsymbol{n}_B\right)\boldsymbol{n}_B\right]. \end{cases} \tag{58}$$

Free-slip BC. $\boldsymbol{u}_{ax_1} = (\boldsymbol{u}_{aB}\cdot\boldsymbol{n}_B)\,\boldsymbol{n}_B$,

$$\begin{cases} \left(\dfrac{\mathrm{d}\rho_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} = -\dfrac{\rho_a}{\mathcal{V}_a}A_{aB}\,\dfrac{d_{a\perp B}^2}{d_a^2}\,\boldsymbol{u}_{ax_1}\cdot\boldsymbol{n}_B + \dfrac{1}{\mathcal{V}_a}A_{aB}\,\dfrac{\Delta t\,\rho_a}{\rho_0}\,\dfrac{\rho_a\,d_{a\perp B}^2}{d_a^2}\,\dfrac{\mathrm{d}\boldsymbol{u_B}}{\mathrm{d}t}\cdot\dfrac{\boldsymbol{r}_{ax_1}}{d_a}\,, \\ \left(\dfrac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right)^{VPH}_{B\to a} = \dfrac{1}{\rho_a\,\mathcal{V}_a}\,A_{aB}\left[\rho_a\left(d_{a\perp B}\left(\boldsymbol{g} - \dfrac{\mathrm{d}\boldsymbol{u_B}}{\mathrm{d}t}\right)\cdot\boldsymbol{n}_B - \boldsymbol{g}\cdot\boldsymbol{c}_{aB}\right)\dfrac{d_{a\perp B}}{d_a^2}\,(d_{a\perp B}\,\boldsymbol{n}_B - \boldsymbol{c}_{aB}) + p_a\,\boldsymbol{n}_B\right] \\ \qquad\qquad - \dfrac{\nu}{\mathcal{V}_a}A_{aB}\,\dfrac{d_{a\perp B}}{d_a^2}\,\boldsymbol{u}_{ax_1} - \dfrac{1}{\mathcal{V}_a}\,A_{aB}\,\dfrac{\alpha\,h_a\,c}{2(D+2)}\,\dfrac{d_{a\perp B}}{d_a^2}\,3\,\boldsymbol{u}_{ax_1}\,. \end{cases} \tag{59}$$

$\underline{d_{a\perp B} \le 0}$ In this category, the particle is located behind the fluid side of the boundary. Evaluating it would require some kind of field interpolation within the cell to link the particle with the fluid area next to the boundary face. Nevertheless, since the face should fall further from the particle than other faces where $d_{a\perp B} > 0$, it should not have a significant impact. As a first approximation, we neglect its effect and assume $d_{a\perp B} < 0 \equiv d_{a\perp B} = 0$, both for no-slip and free-slip BCs.

3.4.3. Free-surface

There is no special treatment for SPH regarding the free-surface (FS). The free-surface boundary conditions (kinematic and free-stress) are a direct consequence of the Lagrangian and explicit nature of the scheme.

Regarding VPH, the free-surface shape needs to be identified from the particle distribution in order to bound properly the Voronoi cells.[4] No previous VPH simulations dealing with FS were found, so we chose to start with a simplified approach described below. The exploration of more advanced methods proposed in the literature for SPH [23,63–65] is left for future-work.

Prior to the generation of the Voronoi tessellation, the angles between the lines connecting a particle with the neighbor elements is explored. Under a flat free-surface, particles should be located parallel to it. We can then place free-surface virtual points (FSVP) to map the free-surface with Voronoi faces. Furthermore, if we assign a distinct angular region for each neighbor without prioritizing any direction, the portion assigned to the FSVP will cover a range of 90° in any angular dimension. Inspired by this fact, the algorithm proposed arranges the neighbors by their angular location with respect to a. If the angle between two adjacent neighbors is greater than 90°, one or more FSVP directions (if $> 180°$) are assigned, as shown in Fig. 14. If the particle has no neighbors, 2^D FSVP directions aligned with the Cartesian coordinate axis are automatically assigned. One free-surface virtual point is created for each valid direction found, and placed at a distance $l_{a,0}$ defined as

$$l_{a,0} = \Delta x_{a,0} = V_{a,0}^{1/D} = \left(\frac{m_a}{\rho_0}\right)^{1/D}. \tag{60}$$

Defining the FSVP this way induces the creation of a face between a and a^{FS}, where the location of the FS is assumed and on which the dynamic and kinematic free-surface BCs are applied:

[4] While SPH is advantageous to treat FS boundary conditions, the free-surface shape can be close to solid boundaries and, therefore, fall within the VPH sub-domain that guarantees that no SPH particle is in contact with these solid boundaries.

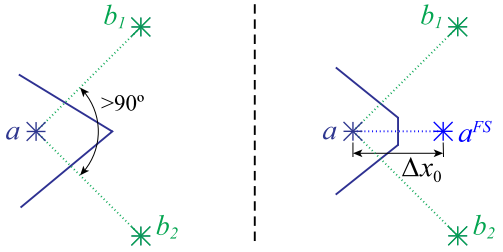


Fig. 14. Voronoi free surface particle detection.

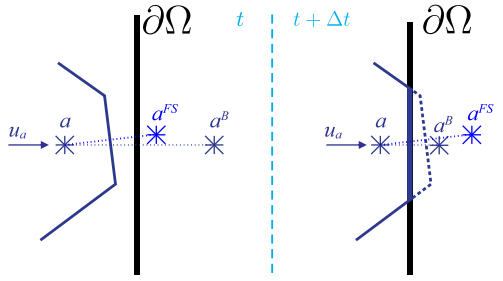


Fig. 15. Voronoi free surface particle approaching a solid boundary.

$$p_{FS} = p_0 \implies p_{a^{FS}} = 2p_0 - p_a\,, \tag{61}$$

$$\boldsymbol{u}_{FS} = \boldsymbol{u}_a \implies \boldsymbol{u}_{aa^{FS}} = \boldsymbol{0}\,. \tag{62}$$

Furthermore, this choice for $l_{a,0}$ leads to the volume of particle a being close to $\mathcal{V}_{a,0}$. Since mass is conserved, the density of such particle is close to ρ_0 and therefore the pressure at the free-surface elements is around p_0. In addition, since the movement of the FSVP is not predicted between time steps, we limit the VPH differential operators to the first order components if a particle is in contact with a FSVP by imposing $\boldsymbol{c}_{ab} = \boldsymbol{0}$ for all its neighbors.

3.4.4. Voronoi free-surface particle close a solid boundary

The same criteria to detect possible free-surface virtual point directions described in Sec. 3.4.3 is followed in the proximity of boundaries. In order to include the angle covered by the boundary, we also include as part of the particle neighbors the linked reflected point a^B and the common boundary nodes if more than one face is near the particle.

However, particles already on the free surface are free to collide with or detach from a boundary. In this case, no reflected point a^B is included in the free-surface criterion, leaving that region open to place a free-surface virtual point a^{FS}. While the relative distance to a^{FS} is fixed at Δx_0, the reflected point a^B approaches as the particle a moves towards the boundary. Both virtual points are used for the Voronoi tessellation, so the algorithm that generates it will implicitly select which one (or maybe both if they are not aligned) is required to close the cell (Fig. 15). This approach provides a smooth transition between free surface and solid boundary conditions.

3.4.5. Periodic

When the flow has spatial periodicity given by the vector $\Delta\boldsymbol{r}$ as shown in Fig. 16, we only need to solve for the flow within a pair of planes spaced $\|\Delta\boldsymbol{r}\|$. As a result, boundary faces are placed on these planes to limit the computational domain, although we cannot use them to bound the Voronoi cells like solid walls because physically they do not exist. Instead, we use the periodic BC to place additional layers of particles beyond the periodic faces, named a^P since each one is linked to a fluid particle located in the nearby of the linked periodic plane: $\boldsymbol{r}_{a^P} = \boldsymbol{r}_a \pm \Delta\boldsymbol{r}$. Periodic particles ensure that the fluid particles remain bounded and are properly surrounded by neighbors. The fluid fields on them are assigned based on their linked particle: $\boldsymbol{u}_{a^P} = \boldsymbol{u}_a$, $p_{a^P} = p_a$.

3.4.6. Inlets/outlets

In a Lagrangian scheme, particles can cross this type of boundaries. Moreover, since SPH and VPH rely on the interaction with neighbor particles, we need to include particles beyond inlet/outlets to ensure that the particles within the fluid domain are well surrounded [66]. In the proposed coupled scheme, only Voronoi particles are in contact with the inlets/outlets to avoid mixing different particle characteristic lengths as happens in the buffer zone. This is enforced by including the distance to inlet/outlets boundaries when evaluating the criteria to determine the VPH and SPH sub-domains (Sec. 3.3.1).

Like periodic faces, inlet/outlet boundaries do not exist physically and cannot be used to bound Voronoi cells. This closure is achieved by a layer of inlet/outlet particles, identified as b^{IO}. In order to ensure that they cover properly the fluid domain,

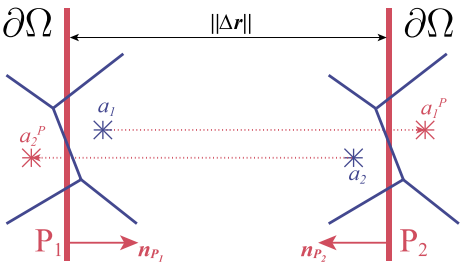


Fig. 16. Periodic particles.

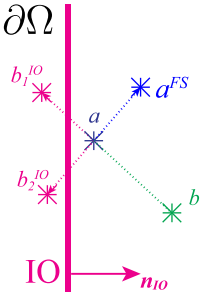


Fig. 17. Inlet/Outlet particles generation. Particle a only detects one neighbor b and places 3 virtual points. 2 of them are beyond the I/O boundary and become I/O particles.

the same algorithm that detects FSVP directions described in Sec. 3.4.3 is used on particles next to inlet/outlet faces. Every virtual point found is evaluated to check if it falls in the orthogonal projection of the inlet/outlet face. If so, it is added as an inlet/outlet particle instead, as shown in Fig. 17.

Inlet/outlet particles are still fluid particles. However, the BCs are used to determine the field values on them and evaluate their movement, instead of relying on the dynamics from their neighbors. Still, their neighbors are used as a reference to extrapolate the field values using linear approximations where Neumann type BCs are applied. If an inlet/outlet particle is linked to multiple regular fluid particles, the contribution from each is determined using a weight value based on the relative contact area, $\omega_{ab^{IO}} = A_{ab^{IO}}/A_{b^{IO}}$, where a $A_{ab^{IO}}$ is the area of the shared face with the linked particle a, and $A_{b^{IO}} = \sum_a A_{ab^{IO}}$ is the total area shared with regular fluid particles. As a result, the following formulations are considered depending on the type of inlet/outlet:

Velocity I/O. $\boldsymbol{u}_{b^{IO}} = \boldsymbol{u}_{IO},\ p_{b^{IO}} = \sum_a \omega_{ab^{IO}}\, p_a.$

Pressure I/O. $\boldsymbol{u}_{b^{IO}} = \sum_a \omega_{ab^{IO}}\, \boldsymbol{u}_a,\ p_{b^{IO}} = p_{IO}.$

3.5. Density re-initialization

Mass is automatically conserved in this Lagrangian scheme as it is lumped in the particles. Volume is obtained from the density that is evolved using the continuity equation, with no explicit conservation requirements under the weakly compressible assumption. The numerical corrections and errors in the integration of the continuity equation allow the particles to "expand" without actually pushing away their neighbors accordingly. As a result, nonphysical high-frequency pressure oscillations arise causing instabilities in the simulation.

We could use the cell volume to recover the consistency of density, mass, and volume within the VPH sub-domain. However, we must keep in mind that the Voronoi diagram is an abstraction used to approximate the differential operators, but does not represent exactly how the original mass associated to the particle is distributed around it at a given time step. Consequently, this mismatch would produce nonphysical density jumps between neighboring cells, leading again to a noisy field. Instead, the density field is periodically re-initialized following Colagrossi & Landrini [12], using a first-order accurate moving-least-squares (MLS) kernel to guarantee its consistency at the particle level, *i.e.* satisfy $\sum_b W_{ab} = 1$. This re-initialization is applied to all fluid particles, including those in the Voronoi zone. In fact, mass does not appear explicitly in the VPH formulation (Sec. 3.2) and we could argue that the smoothed density and volume fields lead to a more consistent mass distribution. However, mass is not re-distributed among the particles since it would conflict with the SPH formulation and progressively scatter the particle sizes, eventually requiring a splitting and merging algorithm not considered at this stage of the project.

To avoid excessive computations, the frequency at which this density re-initialization is applied, Δn_ρ, can be adjusted as an external parameter. Colagrossi & Landrini [12] showed substantial improvements in the flow fields by performing the density re-initialization every 20 steps.

3.6. Shifting algorithm

As particles move, the resulting Voronoi cells tend to become distorted. Furthermore, a Voronoi diagram can have multiple generating particle configurations, which reflects the existence of glass modes that results in a progressive drift of the particles away from the centroid of their cell. In order to balance these effects that reduce the precision of the differential operators, a shifting algorithm inspired by the Lloyd's method [67] is used in the Voronoi sub-domain, similar to Springel [42]. When applied, the distance d_g between the particle and the cell centroid $\boldsymbol{x}_g$ is monitored, shifting the particle if it exceeds a certain given threshold according to Eq. (63).

$$\boldsymbol{r}_a \rightarrow \boldsymbol{r}_a + \begin{cases} 0 & \text{for} \quad \dfrac{d_g}{\zeta\,\Delta x_a} < 0.9, \\ (\boldsymbol{x}_g - \boldsymbol{r}_a)\,\dfrac{d_g - 0.9\zeta\,\Delta x_a}{0.2\zeta\,\Delta x_a} & \text{for } 0.9 \leq \dfrac{d_g}{\zeta\,\Delta x_a} < 1.1, \\ \boldsymbol{x}_g - \boldsymbol{r}_a & \text{for } 1.1 \leq \dfrac{d_g}{\zeta\,\Delta x_a}, \end{cases} \tag{63}$$

where ζ is the coefficient that defines the reference threshold, typically set to $\zeta = 0.25$.

4. Implementation

4.1. Code

The code key routines are written in C++, and the main time-stepping program is implemented in Matlab. MEX functions are used to connect both languages. The code has been tested in 2D but is designed to work also in 3D. Parallelization has not yet been undertaken. The Voronoi tessellation requires around 20% of the computation time. Apart from it, the performance of the code is similar to that of any standard serial SPH model.

Regarding external libraries, *Eigen* C++ library (release 3.3.2) is used for matrix calculations. The generation of the Voronoi diagram is performed using the *qHull* C++ library (release 2005.2). Output data is saved as an HDF5 file (release 1.8.12), whose C++ library is already integrated in Matlab.

4.2. Neighbor boundaries and particles searching algorithm

The range of interaction of a particle is finite in both SPH and VPH schemes. For this reason, an efficient way to identify the neighbors is using a hierarchical tree [68]. We start by creating a uniform grid at each time step with the cell size equal to the maximum smoothing length, $\max(\kappa h_a)$, or particle-neighbor distance, $\max(\|\boldsymbol{r}_{ab}\|)$, whichever is larger. As such, the interaction range of particles belonging to one grid cell is restricted to that cell plus the neighboring ones, either in SPH or VPH.

Next, we determine which boundary faces should be linked to each grid cell, if any. We recall that the domain of influence of the boundaries is characterized by the thickness of the Voronoi and buffer zones (Sec. 3.3.1). Particles within a cell will only evaluate the distance to its associated boundaries to determine which zone they fall in and their coupling weight value, ω.

Finally, a connectivity list is created to determine efficiently which particles belong to a given grid cell. Loops that require the interaction with neighbors go through the grid cells, limiting the interactions to the particles contained in it and its adjacent cells.

4.3. Initialization

Particles are initially placed on a Cartesian grid built over the physical domain, $\Omega_0 := \Omega(t = 0)$, based on a given reference particle size, Δx_0. Afterwards, a Voronoi tessellation is determined for all particles and the shifting algorithm in Sec. 3.6 is applied repeatedly until all particles fall close to the cell centroids. Once this is achieved, flow field variables are assigned to each particle. The volume from the Voronoi cell is used to determine the mass of each particle, $m_a = \rho(\mathbf{r}_a, 0)\,\mathcal{V}_a$, and is kept constant throughout the simulation. In simulations with a free-surface, the background pressure p_0 of the equation of state (4) is set to zero. In confined domain simulations, $p_0 := 0.05\,\rho_0\,c^2$ to make the scheme more sensitive to pressure variations while still remaining within the weakly compressible regime.

A stabilization scheme of the particle positions is executed when starting from static conditions, with the aim of finding a balanced initial condition. All viscous terms in the momentum Eq. (2) are replaced by a linear damping term, $-\xi\,\boldsymbol{u}$, to accelerate the process of reaching equilibrium [9]:

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{g} - \frac{\nabla p}{\rho} - \xi\,\boldsymbol{u}\,, \tag{64}$$

where ξ controls the intensity of the damping and is typically set to $\xi\,\Delta t = 0.05$, with Δt being the time step increment. Sec. 4.4 below shows how it is treated in the time stepping scheme.

During the stabilization phase, the maximum velocity $u_{\max} = \max_a \|\boldsymbol{u}_a\|$ and the total kinetic energy $\mathcal{E}_K = \sum_a 0.5 m_a \|\boldsymbol{u}_a\|^2$ are monitored. The stabilization evolves until $t/t_s \sim 0.1$, with t_s being the maximum time span allowed. From then on, it continues until $t > t_s$, $\max_a \|\boldsymbol{u}_a\|/U < 0.01$, $\mathcal{E}_K^n/\mathcal{E}_0 < 0.001$, or $\left(\mathcal{E}_K^n - \mathcal{E}_K^{n-1}\right)/\Delta t < 0.001\mathcal{E}_0/t_s$, with n being the time step index, U the characteristic velocity of the problem, and $\mathcal{E}_0$ a reference energy.

4.4. Time-stepping

The positions of the particles are advanced in time with an explicit second-order scheme, while a first-order leap frog predictor-corrector scheme is used for the flow fields, following Gray [69] and Souto-Iglesias et al. [70]. With similar computational requirements as the forward Euler method, it shows better performance because it uses information from the predicted state.

Special mention should be made to the linear damping term followed during the stabilization phase in the initialization. Instead of incorporating it as part of the time derivatives, it is treated implicitly and added as an additional component in the time-stepping scheme. Following Monaghan [9], and denoting with ** the predicted values, the leap-frog scheme for the velocity reads

$$\boldsymbol{u}_a^{**} = \frac{1}{1 + 0.5\,\xi\,\Delta t}\left(\boldsymbol{u}_a^n + \Delta t \frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\right), \tag{65}$$

$$\boldsymbol{u}_a^{n+1} = \frac{1}{1 + 0.5\,\xi\,\Delta t}\left[\boldsymbol{u}_a^{**} + \frac{\Delta t}{2}\left(\frac{\mathrm{d}\boldsymbol{u}_a^{**}}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{u}_a^n}{\mathrm{d}t}\right)\right]. \tag{66}$$

Various reference values for the time step increment are defined based on the following criteria that affect the stability of the explicit scheme. The Courant condition for the propagation of sound waves, both for the first and second order terms, yields

$$\Delta t_1 = \min_a\left(\frac{0.5\Delta x_a}{c}\right), \qquad \Delta t_2 = \min_a \sqrt{\Delta x_a \left(\left\|\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right\|\right)^{-1}}. \tag{67}$$

The inter-particle signal speed, defined as $u_{ab}^{sig} = 2c - \boldsymbol{u}_{ab}\cdot\boldsymbol{e}_{ab}$ with index b going through the neighbors of particle a, leads to

$$\Delta t_3 = \min_{ab}\left(\frac{\|\boldsymbol{r}_{ab}\|}{u_{ab}^{sig}}\right). \tag{68}$$

Finally, in order to prevent the velocity vector reversing direction when its magnitude is larger than a given threshold, typically $0.01c$,

$$\Delta t_4 = \min_a\left[\|\boldsymbol{u}_a\|\left(\left\|\frac{\mathrm{d}\boldsymbol{u}_a}{\mathrm{d}t}\right\|\right)^{-1}\right] \qquad \forall a \in \|\boldsymbol{u}_a\| > 0.01c\,. \tag{69}$$

The time step increment is obtained as the minimum of these, affected by a CFL factor. Since we use in this work a first-order time stepping scheme, a conservative $CFL = 0.2$ is chosen.

$$\Delta t = CFL\,\min\left(\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4\right). \tag{70}$$

5. Verification

This section summarizes the results obtained using the δ-VSPH scheme under a set of verification cases, arranged in increasing order of complexity. The results presented by Fernandez-Gutierrez et al. [47] are first reviewed to show how the code performs under fairly simple conditions: hydrostatic equilibrium, sound wave propagation, impulsively-started Couette flow, and Lamb-Oseen vortices. After that, two more challenging configurations are tested: an impinging jet whose analytic solution is known, and a dam break from which experimental data is available in the literature.

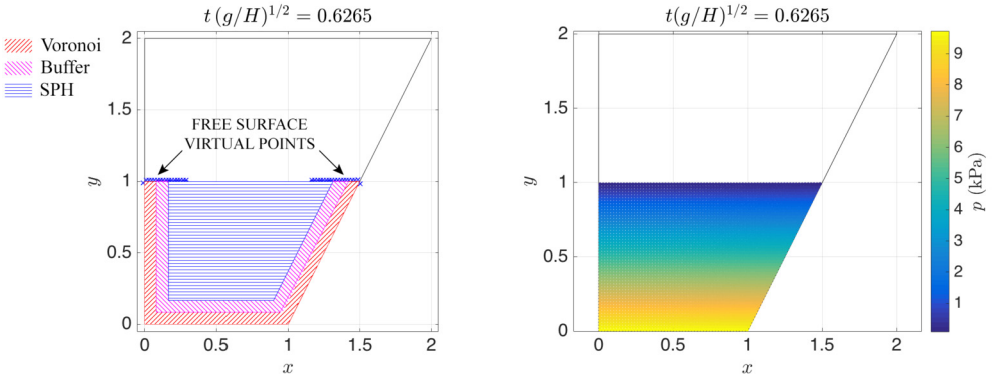


Fig. 18. Left: Zones in a hydrostatic equilibrium simulation of the trapezoidal tank. Right: pressure field at same time.

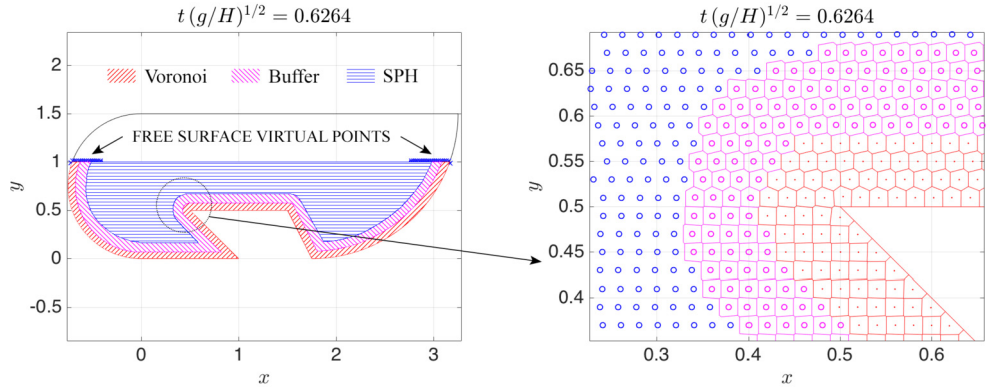


Fig. 19. Left: Zones in a hydrostatic equilibrium simulation with the complex geometry from Colagrossi et al. [71]. Right: Detail of the Voronoi cell shapes next to the concave vertex.

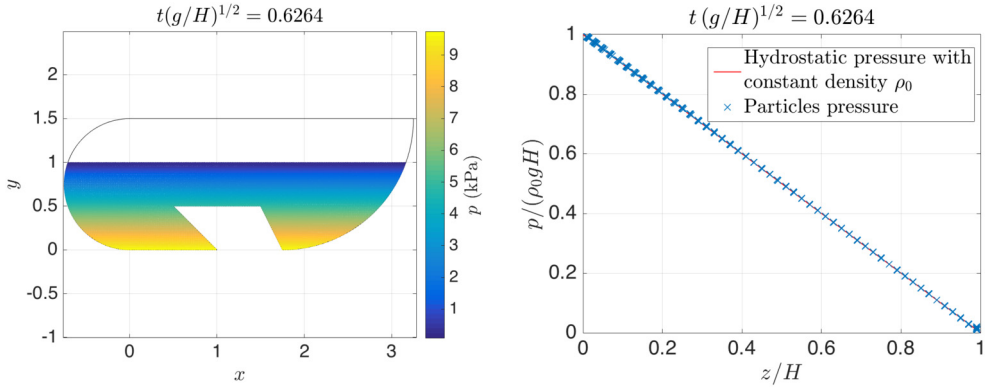


Fig. 20. Hydrostatic equilibrium with concave boundaries: Pressure field.

5.1. Hydrostatic equilibrium

The same configurations studied by Colagrossi et al. [71] are analyzed here: a trapezoidal tank (Fig. 18) and a complex geometry with concave boundaries (Fig. 19), filling in both cases the domain up to height H with 3125 and 7427 particles, respectively. The latter allows us to test the trimming algorithm and the dedicated formulation used to enforce the BC on them (Sec. 3.4.2). As shown in Figs. 18 and 20, no significant differences between the Voronoi and SPH sub-domains are noticeable in the resulting pressure distributions, which show good agreement with the hydrostatic incompressible exact solution.

All particles are initialized at the reference density ρ_0, which is an out-of-equilibrium situation under the weakly compressible regime and the presence of gravity. Thus, vertical oscillations that physically represent an exchange between kinetic

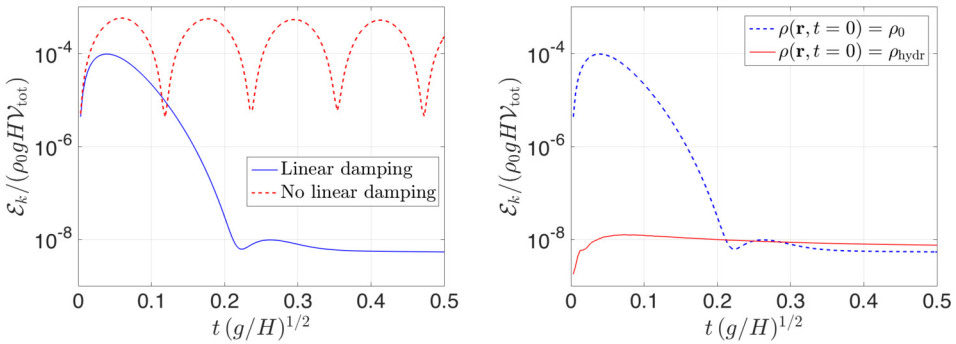


Fig. 21. Kinetic energy evolution (trapezoidal tank). Left: with and without linear damping term; Right: with uniform and hydrostatic density fields.

and elastic energy are expected. The stabilization scheme used during the initialization quickly removes these acoustic-related oscillations, as shown on the left panel of Fig. 21. Moreover, on the right panel is shown how the convergence is faster if we use as initial density field the expected values from the hydrostatic field using the weakly compressible equation of state.

5.2. Pressure driven flows: sound wave propagation and reflection

This test case is inspired by the work of Ott & Schnetter [72], using in our configuration a rectangular 2D domain periodic in y. Particles are evenly distributed and the density field is initialized with a small perturbation term, $\Delta\rho(x)$, modeling a pressure pulse:

$$\rho(\boldsymbol{r}, t=0) = \rho_0 + \Delta\rho(x) = \rho_0 + A\rho_0 x^2 \, \exp\left\{-\left(\frac{x-x_0}{W}\right)^2\right\}, \; x \in (0, L)\,, \tag{71}$$

where $A = 0.005$ m^{-2} controls the amplitude of the pressure wave, $W = 0.1L$ controls its width, $x_0 = 0.5L$ is the location of the initial peak of the pressure wave, and $L = 1$ m is the longitudinal extension of the fluid domain. The propagation direction of the sound wave is controlled by the initial velocity field, set as

$$\boldsymbol{u}^x(\boldsymbol{r}, t=0) = c\,\frac{\Delta\rho(x)}{\rho_0}\,, \quad \boldsymbol{u}^y(\boldsymbol{r}, t=0) = 0\,. \tag{72}$$

The flow is treated as inviscid, so artificial viscosity is included. However, the diffusive δ-correction term is not used because its purpose is to attenuate the pressure waves that we want to observe. 8000 particles are used.

The left panel of Fig. 22 presents longitudinal cross sections of the sound wave at various time steps, where $t^* = t\,c/L$. Given that $x_0 = 0.5\,L$, at $t^* = 1$ the sound wave should return to its initial position, propagating now in the opposite direction. The results show how the shape of the sound wave is kept reasonable well after it crosses the Voronoi zone and is reflected in the wall. The evolution of the energy components variation, $\widehat{\Delta\mathcal{E}} = (\mathcal{E} - \mathcal{E}^0)/\mathcal{E}^0_{\text{tot}}$, is shown on the right panel of Fig. 22. Superscript ^{0} denotes the initial value, while $\mathcal{E}_{\text{tot}}$ is the total energy, which in this case is the combination of kinetic energy, $\mathcal{E}_K$, and elastic energy, $\mathcal{E}_C$, defined for the weakly compressible regime adopted in Eqs. (4)-(5) as

$$\mathcal{E}_C(t) = \begin{cases} \displaystyle\sum_a \frac{m_a\,c^2}{\gamma\,(\gamma-1)}\left[\left(\frac{\rho_a(t)}{\rho_0}\right)^{\gamma-1} - 1\right] + m_a\left(p_0 - \frac{c^2\rho_0}{\gamma}\right)\frac{\rho_a(t)-\rho_0}{\rho_a(t)\,\rho_0} & \text{for } \gamma \neq 1\,, \\ \displaystyle\sum_a m_a\,c^2 \log\left(\frac{\rho_a(t)}{\rho_0}\right) + m_a\left(p_0 - c^2\rho_0\right)\frac{\rho_a(t)-\rho_0}{\rho_a(t)\,\rho_0} & \text{for } \gamma = 1\,, \end{cases} \tag{73}$$

where log is the natural logarithm. Physically, we can observe the conversion in the reflections between kinetic and elastic energy with an overall decay due to numerical dissipation.

5.3. Shear flows: 2D impulsively-started Couette flow

A rectangular domain periodic in x is considered in this case (left panel on Fig. 23), with $a = 1$ m as the height of the channel and 5600 particles used. Particles have zero initial velocity, while the upper and lower boundaries move continuously with the same velocity magnitude, U, and opposite directions, following Monaghan [73]. $\rho(\boldsymbol{r}, t=0) = \rho_0$ and viscosity is adjusted to achieve Reynolds number $Re = a\,U/\nu = 10$. The middle panel of Fig. 23 presents multiple horizontal velocity profiles along the vertical direction at various time steps. Time is made non-dimensional using the mean lifetime of the

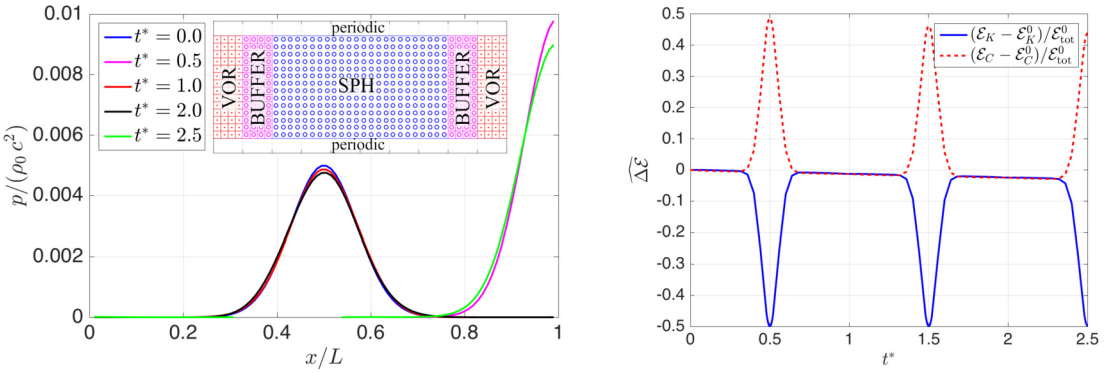


Fig. 22. Left: 2D sound wave propagation simulation. Pressure profiles at different time steps; Right: Sound wave energy decay.

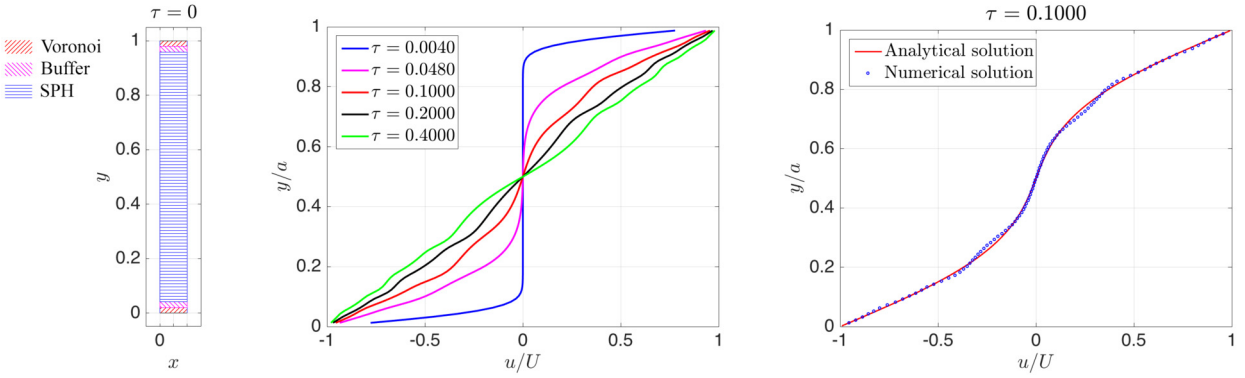


Fig. 23. Impulsive starting Couette. Left: Fluid zones; Right: velocity profiles at various τ.

transition to stationary state, $\tau = t\,\nu/(\rho_0\,(a/2)^2)$. The right panel of Fig. 23 compares the velocity profile at $\tau = 0.1$, which is a representative transition state, with the theoretical solution given by Batchelor [74] and adapted to present boundary conditions,

$$\frac{\boldsymbol{u}^x(y,t)}{U} = -\left(1 - \frac{2y}{a}\right) - \frac{2}{\pi}\sum_{j=1}^{\infty}\frac{1}{j}\sin(j\pi\,\tilde{y})\exp(-j^2\pi^2\tau)\,, \tag{74}$$

where $\tilde{y}$ indicates the distance to the closest boundary made non-dimensional with $a/2$. The agreement is reasonable but small oscillations that deviate from the exact solution are noticeable.

5.4. Lamb–Oseen vortex

The purpose of this test case is to observe the evolution of the velocity field as a result of a pure viscous diffusion process after being initialized with the theoretical values for one single vortex in an infinite domain, presented by Macià et al. [52]:

$$\boldsymbol{u}(\boldsymbol{r}, t=0) = (\boldsymbol{u}^x, \boldsymbol{u}^y) = q\,\frac{1 - \exp\left(-\frac{r^2}{a^2}\right)}{r^2}\,(-y, x)\,, \tag{75}$$

$$a(t)^2 = a_0^2 + 4\,\nu\,t\,, \tag{76}$$

where q controls the intensity of the vortex, a is the effective vortex core radius where the velocity is maximum, ν is the kinematic viscosity, and r is the magnitude of the position vector.

Both circular and square domains are considered, enforcing the no-slip boundary condition on the enclosing walls and using roughly 12k and 16k particles, respectively. The density field is initialized with $\rho(\boldsymbol{r}, t=0) = \rho_0$, and the same input values as Macia are used: $q = 0.5$ m^2/s, $a_0 = 1$ m, and $Re = 2\pi q/\nu = 10$. The simulation results show a good agreement with the theoretical values from Eq. (76), as displayed in Fig. 24.

In order to visualize better the mass transport across the buffer zone, a configuration with two symmetric Lamb–Oseen vortices has also been simulated (Fig. 25). The same values for q, a_0, and ρ are used, although the walls are closer to the

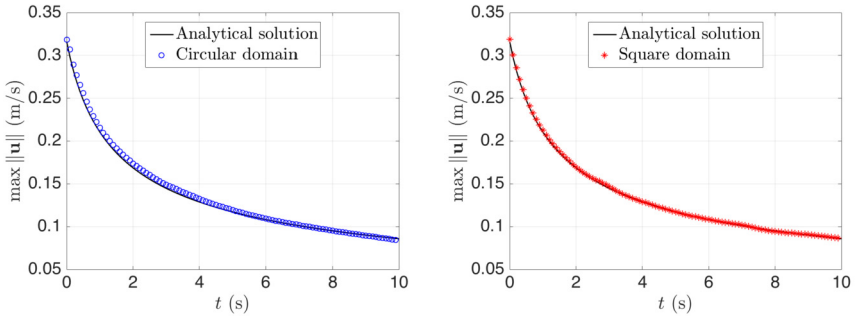


Fig. 24. Evolution of $\max \|\boldsymbol{u}\|$ for one Lamb–Oseen vortex. Left: circular domain; Right: square domain.

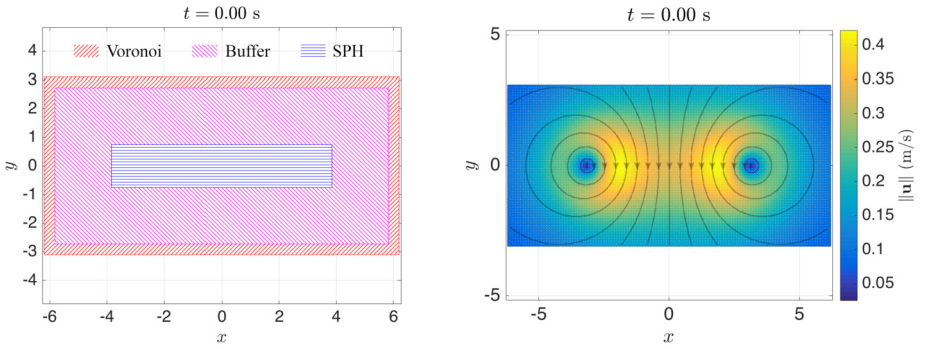


Fig. 25. Left: Initial fluid zones in the simulation with two symmetric Lamb–Oseen vortices. Right: Initial velocity field for the same configuration, with velocity streamlines superimposed.

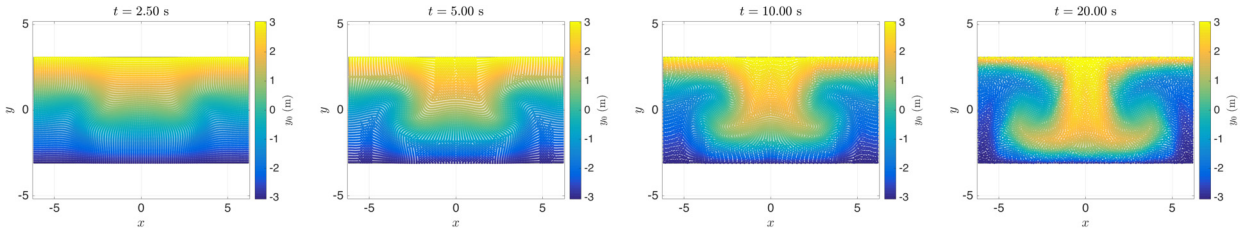


Fig. 26. Particle configuration at different time steps with the two symmetric Lamb–Oseen vortices. Particles colored with their initial vertical position, y_0.

vortex centers in this case. The free-slip BC is applied now at the walls and $Re = 2\pi q/\nu = 60$ to facilitate the movement of the particles. Fig. 26 displays the resulting particle distribution at various time steps, showing how particles move smoothly between the different fluid sub-domains.

5.5. Impinging jet

This verification case involves a velocity inlet on the top through which a column of water with width H and velocity U is injected in the fluid domain. Since gravitational forces are not included, the fluid column does not experience any stretching before approaching the bottom, when it is deflected laterally and moves sideways until it crosses the pressure outlet and leaves the domain. After the initial transient phase, the flow reaches a steady condition whose analytic solution was studied by Michell [75], and discussed further by Taylor [76]. They provide expressions for the velocity and pressure distributions over the bottom as a function of the angle between the jet and the bottom plate, which have been used in the literature as a source of verification for various numerical models [57,50,77]. Particularized for a vertical jet,

$$q = -\frac{2\tilde{u}}{\tilde{u}^2 + 1}\,, \tag{77}$$

$$\frac{\pi x}{H} = \log\left(\frac{1+q}{1-q}\right) + \arcsin q\,, \tag{78}$$

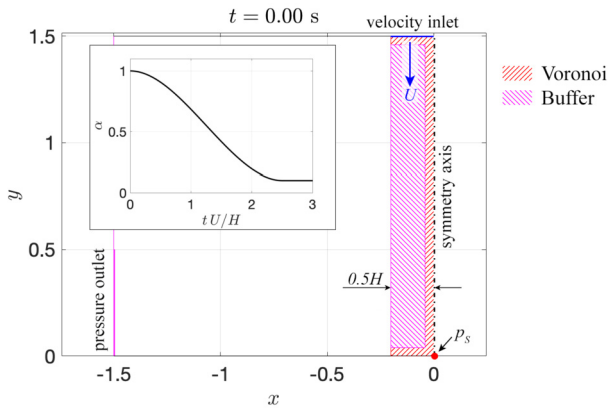


Fig. 27. Initial impinging jet set up and fluid zones, and initial ramp down of artificial viscosity coefficient.

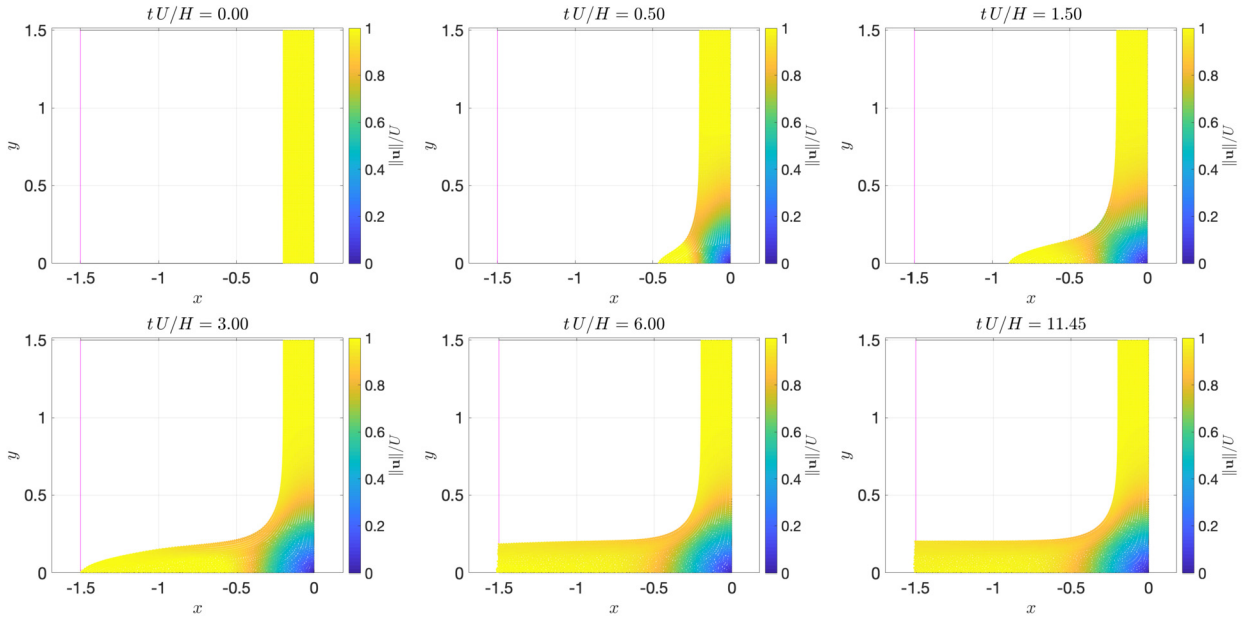


Fig. 28. Particle configuration at different time steps for the impinging jet simulation. Particles colored with their velocity magnitude.

$$\frac{p}{1/2\,\rho U^2} = 1 - \tilde{u}^2\,, \tag{79}$$

where x is the horizontal position, with the origin placed at the intersection of the bottom plane with the impinging jet central axis (see left panel of Fig. 27), $\tilde{u}$ the horizontal velocity made non-dimensional with the incident jet velocity, $\tilde{u} = u/U$, and p the pressure at the bottom. The solution of the free-surface shape was derived by Milne-Thomson & Melville [78] and expressed in terms of its Cartesian coordinates (x_{fs}, y_{fs}), using the same origin as above. For the left branch of a vertical jet,

$$\frac{\pi\, y_{fs}}{H} = \frac{\pi}{2} + \ln\left\{\coth\left[-\frac{\pi}{4}\left(\frac{2x_{fs}}{H}+1\right)\right]\right\}\,. \tag{80}$$

The flow is symmetric around the vertical axis, so only the left branch is simulated. The same parameters as Molteni & Colagrossi [57], who also analyze this configuration using an SPH model, are reproduced here: $H = 0.4$, $\alpha = 0.1$, and $c = 12U$. The flow is initialized when the water column contacts the bottom boundary, as shown in Fig. 27, with $\rho(\boldsymbol{r}, t=0) = \rho_0$ and $\boldsymbol{u}(\boldsymbol{r}, t=0) = (0, -U)$, and the particle size is chosen so $0.5H/\Delta x = 20$. In addition, the artificial viscosity coefficient is set initially to 1.0 to reduce the transient pressure oscillations and reach the steady state faster, and is progressively reduced to its assigned value as shown also in Fig. 27.

Fig. 28 shows the particle distribution at various time steps, with a maximum of 6126 particles reached throughout the simulation. The convergence to the steady state is visualized in Fig. 29, which presents the time-history of the pressure at

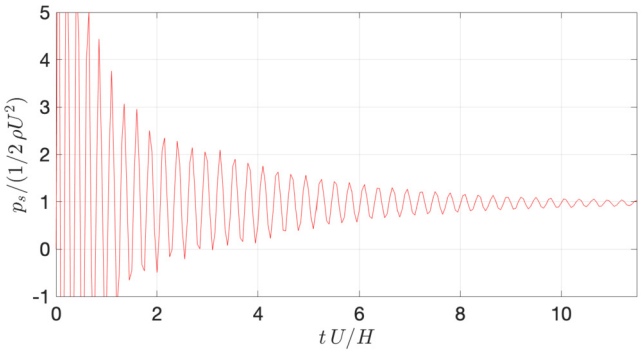


Fig. 29. Time history of the pressure at the stagnation point from the impinging jet simulation.

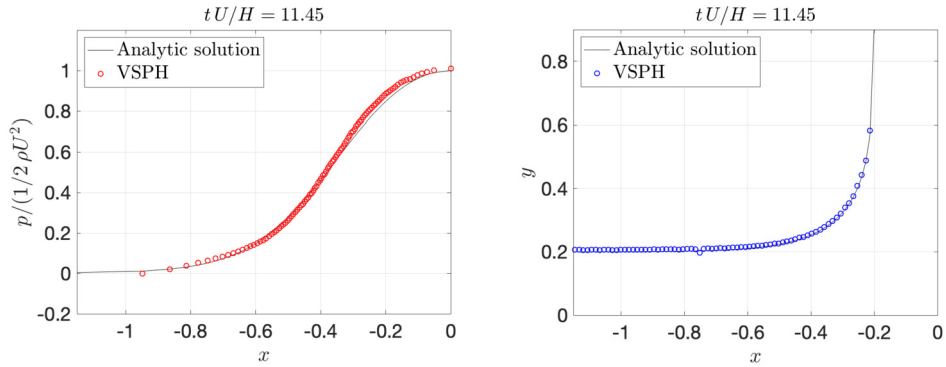


Fig. 30. Comparison between the exact solution and the numerical approximation from the VSPH simulation of the impinging jet. Left: pressure distribution at the bottom plate. Right: free surface shape.

the stagnation point, $p_s/(1/2\,\rho U^2)$, located at the origin. After the initial large oscillations following the impact with the bottom plate, the amplitude of the oscillations is reduced, converging towards the expected analytic value of 1.0. Moreover, Fig. 30 compares the analytic solutions for the pressure over the bottom plate (Eq. (79)) and the free-surface elevation (Eq. (80)) with the resulting values from the simulation, showing excellent agreement.

5.6. Dam break

This verification case has been extensively used by the SPH community to evaluate the capability to simulate breaking waves and free-surface impacts with rigid structures [79,80,14,58,81,60,82,83]. The same configuration as Colagrossi & Landrini [12], who studied this case in detail, is analyzed in this work since they use a particle resolution suitable for our serial code (Fig. 31). The fluid is initialized at rest with the density values determined from the hydrostatic field, based on the results from the hydrostatic case (Sec. 5.1). Free-slip BCs are considered on the tank walls. Under the presence of gravity, the fluid column with an initial height H begins falling and expanding laterally until it hits the side wall and is ejected upwards. Slowed by gravity, it starts running down the wall and it overturns backwards, plunging into the underlying fluid. As a result of this second impact, the fluid bounces back as a rebounding jet. The artificial viscosity coefficient is set to $\alpha = 0.03$ and the particle resolution is $\Delta x = 0.02$ m, same as Colagrossi & Landrini [12], which correspond to 5000 particles. The density field is re-initialized every 20 time steps following Sec. 3.5. Under this configuration, the VPH sub-domain covers practically the entire fluid domain. Fig. 31 shows the pressure field at various time steps.

Given that particles under the VPH formulation only interact with their immediate neighbors, they are more sensitive to their relative movement than under the SPH scheme. As a result, they show a larger cohesion than in SPH, with particles on the free-surface remaining more connected and conforming a well defined shape. The density re-initialization effectively suppress the sonic waves consequence of the weakly compressible relaxation, as found by Colagrossi & Landrini [12], yielding smooth pressure fields over the fluid domain. The resulting free-surface shape and field values match well the results available in the literature, given the particle resolution used.

Experimental data using this configuration was first presented by Zhou et al. [84], with the experimental set-up described in detail by Buchner [85] and Lee et al. [86]. Further experimental work was presented by Lobovský et al. [87], who used more accurate sensors to report pressure on the right wall and wave elevation in a larger number of locations. Fig. 32 shows the vertical position of each p_i pressure gauge and the longitudinal location of each z_i water column height probe, respectively, and Figs. 33-34 compare the values from the simulations with the reported experimental results.

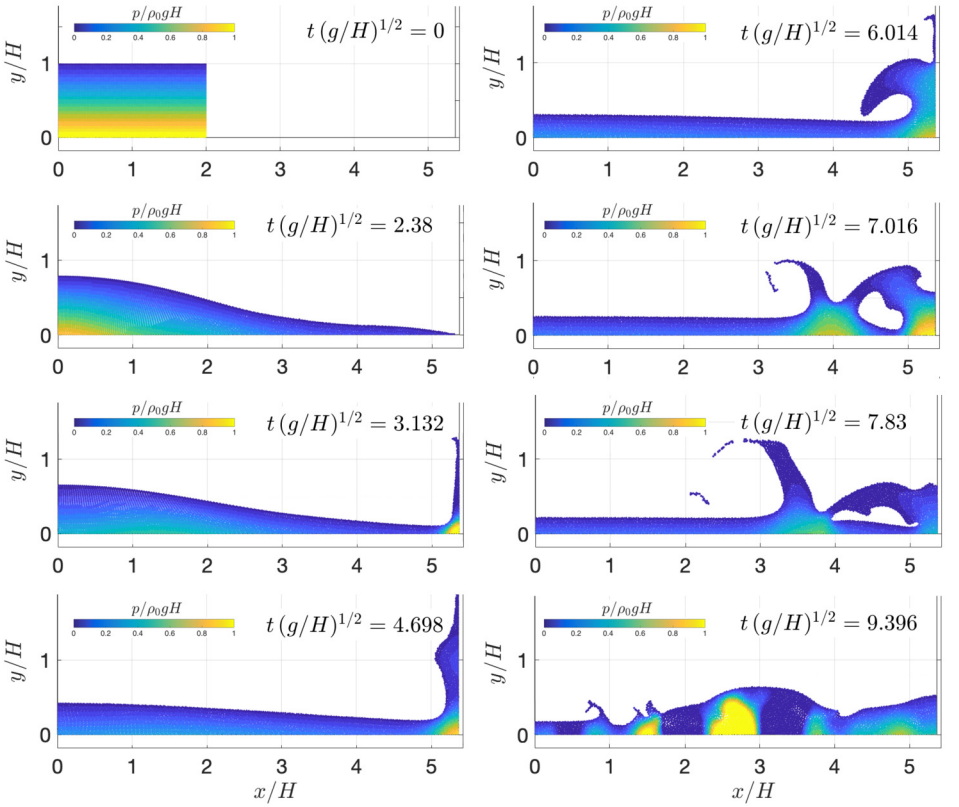


Fig. 31. Pressure field at different time steps for the dam break simulation.

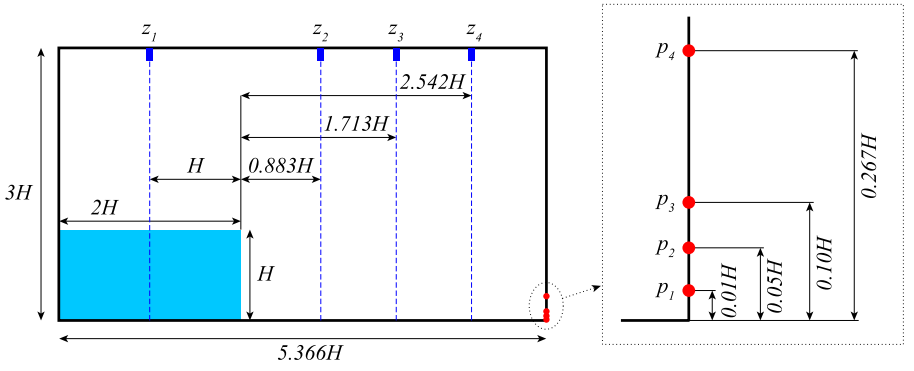


Fig. 32. Location of pressure and wave height sensors in the experimental set-up of Lobovský et al. [87].

While the overall pressure trends from the experimental values are reproduced, some differences are noticeable, with generally larger pressures computed by the numerical model. This observation suggests the use of a smaller particle size to overcome the larger dependence on the immediate neighbors under the VPH scheme when the free-surface is close to solid boundaries. On the other hand, the water column height results match well the experimental values up to the arrival of the secondary wave generated by the fluid overturning backwards, when the impact of the air-entrapment phenomena not modeled by the single-phase model becomes significant, affecting mostly z_2 and z_3.

6. Concluding remarks

The methodology and results presented in this work show the feasibility of combining SPH with Voronoi diagrams into a hybrid Lagrangian SPH-VPH scheme, with an explicit weakly compressible formulation for both methods. The fluid domain is subdivided into three separate zones: close to the boundaries where only VPH is applied, far from the boundaries where only SPH is applied, and a transition zone where all fields are computed using both methods and combined through a

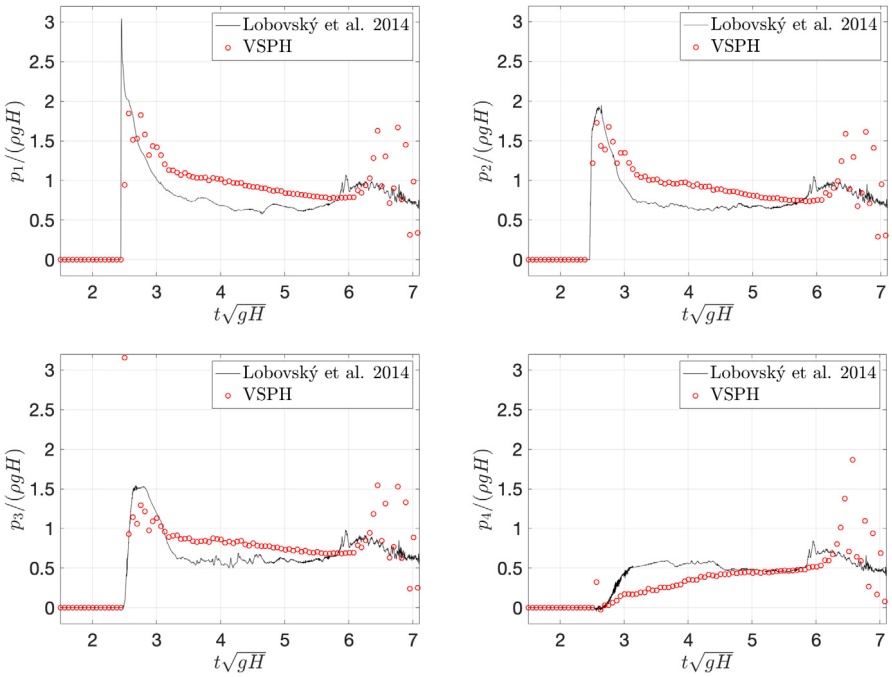


Fig. 33. Time history comparison between computed and measured values of wall pressure for the dam break.

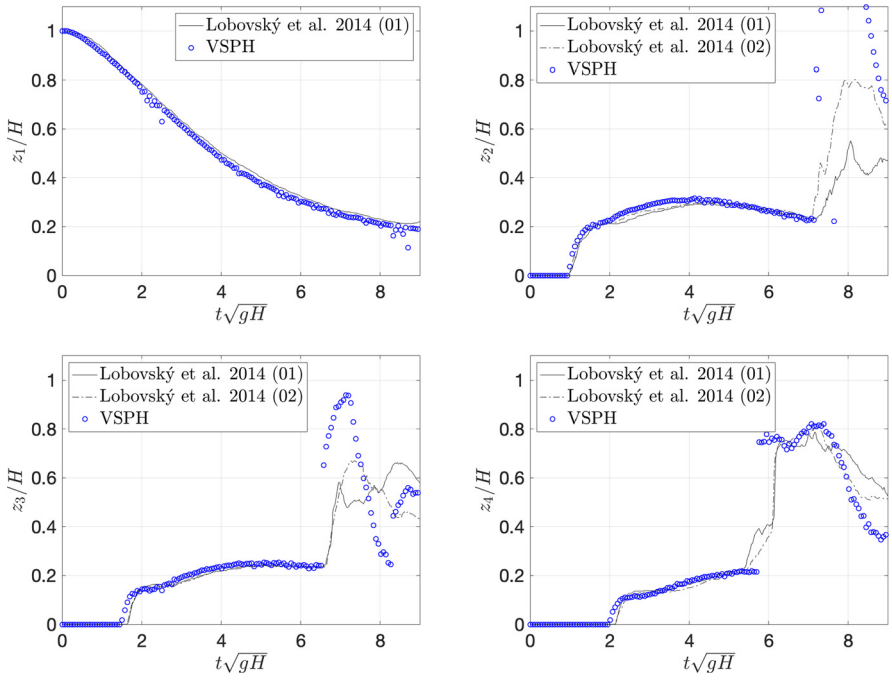


Fig. 34. Time history comparison between computed and measured values of water height characterizing the free-surface shape for the dam break.

weighting function. While SPH is based on smoothing using a kernel function, VPH uses volume averaged approximations based on a Voronoi tessellation [44,43]. However, both methods are conceptually very close, making it logical to extend to VPH the stabilization corrections proposed in the literature for SPH. As such, new artificial viscosity and diffusive terms are included in the VPH scheme inspired by the δ-SPH standard formulation [50,58,60].

The new coupled scheme proposed extends the artificial viscosity and δ-correction to the VPH formulation, incorporates free-surfaces to it for the first time, and is capable of dealing with complex geometries. The treatment of the boundary conditions is greatly simplified by using Voronoi diagrams. Under the VPH scheme, the interaction with neighbor elements

is done through the cell face shared with them. If the faces overlap the boundaries, the known physical values from the BCs can be directly used to evaluate the VPH differential operators. Special emphasis has been placed in the algorithm generating the tessellation to enforce the exact mapping of boundary faces. Moreover, it is done without imposing any restriction on the particles movement.

The accuracy of the coupled scheme is discussed by means of a set of well-known verification benchmarks. The results showed that pressure gradient dominated problems, such as hydrostatic conditions or sound wave propagations, are well simulated by the method. However, problems which are dominated by viscous diffusion render less accurate results, consistent with the increased error made in the higher order differential operators. The oscillations observed in the impulsively-started Couette flow reflect the impact of sonic effects, even in shear driven fluids. While they progressively vanish, a larger attenuation or a lower sensitivity to density variations must be considered if the transient phase is the main focus of the simulation.

While this work establishes the foundation of the δ-VSPH method, it remains to be further developed in multiple areas to demonstrate its full potential under realistic three-dimensional applications. Fortunately, the theoretical background behind the use of Voronoi diagrams is directly applicable to any dimension with no modification of the scheme. In fact, the *qHull* C++ library used to generate the Voronoi tesellation already allows its computation in any arbitrary number of dimensions [88]. Moreover, all the δ-VSPH formulation presented in Sec. 3 is applicable as presented for 3D configurations as well, with the only exception of the free-surface detection algorithm (Sec. 3.4.3). Given that multiple angular dimensions exist when moving beyond 2D, it is not possible to arrange neighbors by their angular location. Instead, the search for possible free-surface directions can be limited to a finite set of predefined values to limit the computational cost. Still, the number of particles grow significantly when considering 3D scenarios. Thus, modifying our C++/Matlab implementation to take advantage of the *Open MPI*[5] C++ library to allow parallel computing with distributed-memory systems is among the immediate next steps to be taken.

The great advantage of the coupled scheme presented is that it combines the core formulations of SPH and VPH, but does not establish any further restriction on each. The large similarities between both methods will simplify transferring the techniques proposed for the more mature SPH to VPH.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank Prof. Antonio Souto-Iglesias from the Technical University of Madrid, whose assistance during his stay at UC Berkeley from September 2016 to June 2017 was instrumental to conduct this research.

References

[1] S. Li, W.K. Liu, Meshfree Particle Methods, 1 ed., Springer-Verlag Berlin Heidelberg, 2004.
[2] J. Monaghan, Smoothed particle hydrodynamics, Rep. Prog. Phys. 68 (2005) 1703–1759.
[3] J. Monaghan, Smoothed particle hydrodynamics and its diverse applications, Annu. Rev. Fluid Mech. 44 (2012) 323–346.
[4] G. Liu, M. Liu, Smoothed Particle Hydrodynamics. A Meshfree Particle Method, World Scientific Publishing Co. Pte. Ltd., 2003.
[5] V. Springel, Smoothed particle hydrodynamics in astrophysics, Annu. Rev. Astron. Astrophys. 48 (2010) 391–430.
[6] D. Violeau, Fluid Mechanics and the SPH Method: Theory and Applications, Oxford University Press, 2012.
[7] L. Lucy, A numerical approach to the testing of the fission hypothesis, Astron. J. 82 (1977) 1013–1024.
[8] R. Gingold, J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. R. Astron. Soc. (MNRAS) 181 (1977) 375–389.
[9] J. Monaghan, Simulating free surface flows with SPH, J. Comput. Phys. 110 (1994) 39–406.
[10] L. Libersky, A. Petschek, T. Carney, J. Hipp, F. Allahdadi, High strain Lagrangian hydrodynamics a three-dimensional SPH code for dynamic material response, J. Comput. Phys. 109 (1993) 67–75.
[11] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1997) 214–226.
[12] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, J. Comput. Phys. 191 (2003) 448–475.
[13] M. Yildiz, R.A. Rook, A. Suleman, SPH with the multiple boundary tangent method, Int. J. Numer. Methods Eng. 77 (2009) 1416–1438.
[14] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, G. Graziani, Delta-SPH model for simulating violent impact flows, Comput. Methods Appl. Mech. Eng. 200 (2011) 1526–1542.
[15] M. De Leffe, D. Le Touzé, B. Alessandrini, A modified no-slip condition in weakly-compressible SPH, in: 6th ERCOFTAC SPHERIC Workshop on SPH Applications, 2011, pp. 291–297.
[16] F. Macià, M. Antuono, L.M. González, A. Colagrossi, Theoretical analysis of the no-slip boundary condition enforcement in SPH methods, Prog. Theor. Phys. 125 (2011) 1091–1121.
[17] J. Feldman, J. Bonet, Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems, Int. J. Numer. Methods Eng. 72 (2007) 295–324.
[18] S. Hieber, P. Koumoutsakos, An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers, J. Comput. Phys. 227 (2008) 8636–8654.

[5] https://www.open-mpi.org.

[19] J. Marongiu, F. Leboeuf, E. Parkinson, Riemann solvers and efficient boundary treatments: an hybrid SPH-finite volume numerical method, in: 3rd ERCOFTAC SPHERIC Workshop on SPH Applications, EPFL, 2008.
[20] M. De Leffe, D. Le Touzé, B. Alessandrini, Normal flux method at the boundary for SPH, in: 4th SPHERIC, 2009, pp. 149–156.
[21] J.P. Vila, On particle weighted methods and smooth particle hydrodynamics, Math. Models Methods Appl. Sci. 09 (1999) 161–209.
[22] G.A. Dilts, Moving-least-squares-particle hydrodynamics–I. Consistency and stability, Int. J. Numer. Methods Eng. 44 (1999) 1115–1155.
[23] G.A. Dilts, Moving least-squares particle hydrodynamics II: conservation and boundaries, Int. J. Numer. Methods Eng. 48 (2000) 1503–1524.
[24] B.B. Moussa, N. Lanson, J.P. Vila, Convergence of meshless methods for conservation laws applications to Euler equations, in: M. Fey, R. Jeltsch (Eds.), Hyperbolic Problems: Theory, Numerics, Applications: Seventh International Conference in Zürich, February 1998 Volume I, Birkhäuser, Basel, 1999, pp. 31–40.
[25] B. Moussa, J. Villa, Convergence of SPH method for scalar nonlinear conservation laws, SIAM J. Numer. Anal. (2000).
[26] B. Ben Moussa, On the convergence of sph method for scalar conservation laws with boundary conditions, Methods Appl. Anal. 13 (2006) 29–62.
[27] J.-C. Marongiu, F. Leboeuf, J. Caro, E. Parkinson, Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method, J. Hydraul. Res. 48 (2010) 40–49.
[28] S. Marrone, A.D. Mascio, D. Le Touzé, Coupling of smoothed particle hydrodynamics with finite volume method for free-surface flows, J. Comput. Phys. 310 (2016) 161–180. Available online 11 December 2015.
[29] P. Kumar, Q. Yang, V. Jones, L. McCue-Weil, Coupled SPH-FVM simulation within the OpenFOAM framework, Proc. IUTAM 18 (2015) 76–84.
[30] E. Napoli, M.D. Marchis, C. Gianguzzi, B. Milici, A. Monteleone, A coupled finite volume-smoothed particle hydrodynamics method for incompressible flows, Comput. Methods Appl. Mech. Eng. 310 (2016) 674–693.
[31] S. Idelsohn, E. Oñate, F.D. Pin, The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves, Int. J. Numer. Methods Eng. 61 (2004) 964–989.
[32] E. Oñate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows, Comput. Methods Appl. Mech. Eng. 197 (2008) 1777–1800. Computational Methods in Fluid–Structure Interaction.
[33] G. Shobeyri, R.R. Ardakani, Improving accuracy of SPH method using Voronoi diagram, Iran. J. Sci. Technol., Trans. Civ. Eng. 41 (2017) 345–350.
[34] M.A. Ghaffari, S. Xiao, Smoothed particle hydrodynamics with stress points and centroid Voronoi tessellation (CVT) topology optimization, Int. J. Comput. Methods 13 (2016) 1650031.
[35] G. Chiaki, N. Yoshida, Particle splitting in smoothed particle hydrodynamics based on Voronoi diagram, Mon. Not. R. Astron. Soc. 451 (2015) 3955–3963.
[36] G. Voronoi, Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs, J. Reine Angew. Math. 134 (1908) 198–287.
[37] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu, Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, John Wiley and Sons, Ltd, 2000.
[38] M. De Berg, O. Cheong, M. Van Kreveld, M. Overmars, Computational Geometry: Introduction, Springer, 2008.
[39] A. Okabe, Spatial tessellations, in: International Encyclopedia of Geography: People, the Earth, Environment and Technology, John Wiley and Sons, Ltd, 2016, pp. 1–11.
[40] D.A. Barcarolo, Improvement of the Precision and the Efficiency of the SPH Method: Theoretical and Numerical Study, Ph.D. thesis, Ecole Centrale de Nantes, 2013.
[41] D.A. Barcarolo, D.L. Touzé, G. Oger, F. de Vuyst, Voronoi-SPH: on the analysis of a hybrid finite volumes - smoothed particle hydrodynamics method, in: 9th Int. SPHERIC Workshop, 2014, pp. 371–378.
[42] V. Springel, E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh, Mon. Not. R. Astron. Soc. 401 (2010) 791.
[43] S. Hess, V. Springel, Particle hydrodynamics with tessellation techniques, Mon. Not. R. Astron. Soc. 406 (2010) 2289–2311.
[44] M. Serrano, P. Español, Thermodynamically consistent mesoscopic fluid particle model, Phys. Rev. E 64 (2001) 046115.
[45] M. Serrano, Comparison between smoothed dissipative particle dynamics and Voronoi fluid particle model in a shear stationary flow, Phys. A, Stat. Mech. Appl. 362 (2006) 204–209.
[46] D. Fernández-Gutiérrez, Delta Voronoi Smoothed Particle Hydrodynamics, δ-VSPH, Ph.D. thesis, University of California at Berkeley, 2011.
[47] D. Fernández-Gutiérrez, A. Souto-Iglesias, T.I. Zohdi, A hybrid Lagrangian Voronoi–SPH scheme, Comput. Part. Mech. 5 (2018) 345–354.
[48] S. Marrone, A. Colagrossi, M. Antuono, G. Colicchio, G. Graziani, An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers, J. Comput. Phys. 245 (2013) 456–475.
[49] A. Colagrossi, A. Souto-Iglesias, M. Antuono, S. Marrone, Smoothed-particle-hydrodynamics modeling of dissipation mechanisms in gravity waves, Phys. Rev. E 87 (2013) 023302.
[50] M. Antuono, A. Colagrossi, S. Marrone, D. Molteni, Free-surface flows solved by means of SPH schemes with numerical diffusive terms, Comput. Phys. Commun. 181 (2010) 532–549.
[51] F. Macià, A. Colagrossi, M. Antuono, A. Souto-Iglesias, Benefits of using a Wendland kernel for free-surface flows, in: 6th ERCOFTAC SPHERIC Workshop on SPH Applications, 2011.
[52] F. Macià, J.M. Sánchez, A. Souto-Iglesias, L.M. González, WCSPH viscosity diffusion processes in vortex flows, Int. J. Numer. Methods Fluids 69 (2012) 509–533.
[53] P. Español, M. Revenga, Smoothed dissipative particle dynamics, Phys. Rev. E 67 (2003) 026705.
[54] D. Violeau, Dissipative forces for Lagrangian models in computational fluid dynamics and application to smoothed-particle hydrodynamics, Phys. Rev. E 80 (2009) 036705.
[55] X.Y. Hu, N.A. Adams, Angular-momentum conservative smoothed particle dynamics for incompressible viscous flows, Phys. Fluids 18 (2006) 101702.
[56] A. Ferrari, M. Dumbser, E.F. Toro, A. Armanini, A new 3D parallel SPH scheme for free surface flows, Comput. Fluids 38 (2009) 1203–1217.
[57] D. Molteni, A. Colagrossi, A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH, Comput. Phys. Commun. 180 (2009) 861–872.
[58] M. Antuono, A. Colagrossi, S. Marrone, Numerical diffusive terms in weakly-compressible SPH schemes, Comput. Phys. Commun. 183 (2012) 2570–2580.
[59] J.L. Cercos-Pita, A Novel Generalized Diffusive SPH Model: Theoretical Analysis and 3D HPC Implementation, Ph.D. thesis, Technical University of Madrid (UPM), 2016.
[60] J. Cercos-Pita, R. Dalrymple, A. Herault, Diffusive terms for the conservation of mass equation in SPH, Appl. Math. Model. 40 (2016) 8722–8736.
[61] D. Duque, P. Español, J.A. de la Torre, Extending linear finite elements to quadratic precision on arbitrary meshes, Appl. Math. Comput. 301 (2017) 201–213.
[62] N.J. Quinlan, M. Lastiwka, M. Basa, Truncation error in mesh-free particle methods, Int. J. Numer. Methods Eng. 66 (2006) 2064–2085.
[63] A. Haque, G.A. Dilts, Three-dimensional boundary detection for particle methods, J. Comput. Phys. 226 (2007) 1710–1730.
[64] S. Marrone, A. Colagrossi, D. Le Touzé, G. Graziani, Fast free-surface detection and level-set function definition in SPH solvers, J. Comput. Phys. 229 (2010) 3652–3663.
[65] S. Marrone, Enhanced SPH Modeling of Free-Surface Flows with Large Deformations, Ph.D. thesis, University of Rome, La Sapienza, 2011.
[66] I. Federico, S. Marrone, A. Colagrossi, F. Aristodemo, M. Antuono, Simulating 2D open-channel flows through an SPH model, Eur. J. Mech. B, Fluids 34 (2012) 35–46.
[67] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (1982) 129–137.

[68] L. Hernquist, N. Katz TreeSPH, A unification of SPH with the hierarchical tree method, Astrophys. J. Suppl. 70 (1989) 419–446.
[69] J. Gray, Caldera Collapse and the Generation of Waves, Ph.D. thesis, Monash University, 2001.
[70] A. Souto-Iglesias, L. Delorme, L. Pérez-Rojas, S. Abril-Pérez, Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics, Ocean Eng. 33 (2006) 1462–1484.
[71] A. Colagrossi, B. Bouscasse, M. Antuono, S. Marrone, Particle packing algorithm for SPH schemes, Comput. Phys. Commun. 183 (2012) 1641–1683.
[72] F. Ott, E. Schnetter, A modified SPH approach for fluids with large density differences, arXiv:physics/0303112, 2003.
[73] J. Monaghan, Smoothed particle hydrodynamic simulations of shear flow, Mon. Not. R. Astron. Soc. 365 (2005) 199–213.
[74] G.K. Batchelor, Introduction to Fluid Dynamics, Cambridge University Press, New York, 1967.
[75] J.H. Michell, J.J. Thomson, V. On the theory of free stream lines, Philos. Trans. R. Soc. Lond. A 181 (1890) 389–431.
[76] G. Taylor, Oblique impact of a jet on a plane surface, Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Sci. 260 (1966) 96–100.
[77] M. Dumbser, A simple two-phase method for the simulation of complex free surface flows, Comput. Methods Appl. Mech. Eng. 200 (2011) 1204–1219.
[78] L.M. Milne-Thomson, Theoretical Hydrodynamics, fifth ed., The Mac-Millan Press, Ltd., 1968.
[79] G. Colicchio, A. Colagrossi, M. Greco, M. Landrini, Free-surface flow after a dam break: a comparative study, Ship Technol. Res. 49 (2002) 95–104.
[80] A. Khayyer, H. Gotoh, S. Shao, Enhanced predictions of wave impact pressure by improved incompressible sph methods, Appl. Ocean Res. 31 (2009) 111–131.
[81] S. Marrone, A. Colagrossi, A. Di Mascio, D. Le Touzé, Analysis of free-surface flows through energy considerations: single-phase versus two-phase modeling, Phys. Rev. E 93 (2016) 053113.
[82] X. Cao, F. Ming, A. Zhang, L. Tao, Multi-phase sph modelling of air effect on the dynamic flooding of a damaged cabin, Comput. Fluids 163 (2018) 7–19.
[83] D.D. Meringolo, S. Marrone, A. Colagrossi, Y. Liu, A dynamic δ-sph model: how to get rid of diffusive parameter tuning, Comput. Fluids 179 (2019) 334–355.
[84] Z. Zhou, J. De Kat, B. Buchner, A nonlinear 3-D approach to simulate green water dynamics on deck, in: Proc. 7th International Symposium on Numerical Ship Hydrodynamics, Report, vol. 7, 1999.
[85] B. Buchner, Green Water on Ship-Type Offshore Structures, Ph.D. thesis, Delft University of Technology, 2002.
[86] T.-h. Lee, Z. Zhou, Y. Cao, Numerical simulations of hydraulic jumps in water sloshing and water impacting, J. Fluids Eng. 124 (2002) 215–226.
[87] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, A. Souto-Iglesias, Experimental investigation of dynamic pressure loads during dam break, J. Fluids Struct. 48 (2014) 407–434.
[88] C.B. Barber, D.P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Trans. Math. Softw. 22 (1996) 469–483.