

A geometric formulation of the Shepard renormalization factor

J. Calderon-Sanchez^{a,c}, J.L. Cercos-Pita^b, D. Duque^{a,c}

^aCEHINAV, ETS Ingenieros Navales, Universidad Politécnica de Madrid, Avd. de la Memoria 4, 28040 Madrid, Spain

^bResearch department, NASAL Systems SL, General Orgaz 23, 28020 Madrid, Spain

^cDep. FAIAN, ETSIAE, Pza. de Cardenal Cisneros, 3 Universidad Politécnica de Madrid, 28040 Madrid, Spain

Abstract

The correct treatment of boundary conditions is a key step in the development of the SPH method. The SPH community has to face several challenges in this regard — in particular, a primordial aspect for any boundary formulation is to ensure the consistency of the operators in presence of boundaries and free surfaces. A new implementation is proposed, based on the existing numerical boundary integrals formulation. A new kernel expression is developed to compute the Shepard renormalization factor at the boundary purely as a function of the geometry. In order to evaluate this factor, the resulting expression is split into numerical and analytical parts, which allows accurately computing the Shepard factor. The new expression is satisfactorily tested for different planar geometries, showing that problems featuring free surfaces and boundaries are solved. The methodology is also extended to 3-D geometries without great increase in computational cost.

Keywords: Particle methods, Meshless methods, Smoothed Particle Hydrodynamics, Boundary Integrals

1. Introduction.

The treatment of boundary conditions is an important step in the development of any numerical method. During the past few years, SPH methodology regarding boundary conditions has improved considerably, thanks in part to the community addressing the SPHERIC Grand Challenges, proposed by the SPHERIC SPH Numerical Development Working Group. Boundary conditions issues are specially relevant when dealing with free-surface flows, as discussed in detail in Ref. [10].

The truncation of the kernel at the ends of the domain makes SPH interpolation process inaccurate, unless consistency of the operators is recovered. In this regard, several approaches to model boundary conditions exist, although two main groups can be distinguished among all the options available: fluid extensions and contour closure.

Email addresses: javier.calderon@upm.es (J. Calderon-Sanchez), jl.cercos@upm.es (J.L. Cercos-Pita), daniel.duque@upm.es (D. Duque)

Fluid extensions are based on the expansion of the system beyond the boundaries, in order to complete the kernel support with mirror images of the fluid particles, which are added to the SPH governing equations in order to restore the consistency of the differential operators. Along this line, the ghost particles approach [4, 23] has become the most popular option. Nevertheless, some non-trivial challenges should be addressed, such as the choices for the particle locations, which can turn to a hard task for relatively complex geometries, or the procedure to assign field values to the new particles — see e.g. Refs. [11, 25].

Alternatively, wall-like boundary conditions to model boundaries may be used, by the closure of the domain through actual surface patches. The original approach, by Campbell [5], was further developed in Refs. [13, 19]. However, the first consistent formulation is found in Ref. [17], even though some aspects of the consistency of such a method were first analyzed in Ref. [22]. The boundary integrals formulation of Ref. [17] considers an analytical kernel factor close to the boundaries, modeled as elements with associated areas and normal vectors.

Later on, a purely numerical boundary integrals methodology was developed in Ref. [9], which may be easily extended to 3-D simulations. To this end, the boundary is again discretized into elements which represent an actual area and have a normal associated to the wall they are modeling.

The boundary integrals methodology, though still in development, is a powerful choice — especially for complex geometries and 3-D applications, where the ghost particle approach becomes too complex. Nevertheless, the main setback in Ferrand’s formulation [17] is the high computational cost introduced by the connectivity between surface elements. There is therefore a need for a faster boundary integrals methodology. On the other hand, the numerical boundary integrals methodology is not as accurate as the semi-analytical formulation, as a truncation error is assumed and the symmetry of the operators is broken due to Shepard renormalization. Effort is hence needed in this regard.

The aim of this work is to improve the accuracy of the numerical boundary integrals formulation, while avoiding a high computational cost. In order to achieve this, an alternative formulation of the Shepard renormalization factor for the boundary integrals methodology, close to boundaries, will be introduced in Section 2. Afterwards, an analysis to properly assess the new formulation will be carried out in Section 2.2. Finally, the new formulation will be tested against different geometries in Section 3, where results for 2-D case and a 3-D dam break will be presented.

2. The role of the Shepard renormalization factor in the Boundary Integrals formulation

Figure 1 shows the schematic view of a point convolution \boldsymbol{x} in the presence of a boundary. In such situation, the compact support of the kernel, $\Omega(\boldsymbol{x})$, can be divided into two different subdomains. The fluid subdomain is denoted by $\bar{\Omega}(\boldsymbol{x})$ and the subdomain across the boundary, $\Omega^*(\boldsymbol{x})$. For the sake of simplicity, from now on, kernel function notation will be Ω , $\bar{\Omega}$ and Ω^* respectively, following the notation of Ref. [2].

Although fluid information in $\bar{\Omega}$ is always known, information beyond the wall (Ω^*) is not known *a priori*, hence different techniques, as the ones mentioned in Section 1, have been developed in this regard. In this work, we will focus on the gradient operator, which plays an important role in SPH simulations. Expanding the continuous convolution of the first order differential operator, for a generic field f and considering both subdomains, yields:

$$\begin{aligned} \langle \mathcal{D}f(\mathbf{x}) \rangle &= \int_{\bar{\Omega}} f(\mathbf{y}) \nabla W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y} \\ &+ \int_{\Omega^*} f(\mathbf{y}) \nabla W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y}, \end{aligned} \quad (1)$$

where the bracket operator $\langle \rangle$ represents SPH interpolation, as in Ref. [13], and \mathbf{y} is a generic point inside the kernel support. The kernel $W_h(\mathbf{y} - \mathbf{x})$ is supposed to depend only on the distance between points, divided by a characteristic constant length h . The kernel has compact support, i.e. it vanishes for $|\mathbf{y} - \mathbf{x}| > s h$, where s is an integer greater than 0. Applying the divergence theorem, and the kernel properties, a popular expression for the boundary integrals formulation [17] is found:

$$\begin{aligned} \langle \mathcal{D}f(\mathbf{x}) \rangle &= \frac{1}{\gamma(\mathbf{x})} \left(\int_{\bar{\Omega}} f(\mathbf{y}) \nabla W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y} \right. \\ &\quad \left. + \int_{\partial\bar{\Omega}} f(\mathbf{y}) \cdot \mathbf{n}(\mathbf{y}) W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y} \right) \end{aligned} \quad (2)$$

where the Shepard renormalization factor $\gamma(\mathbf{x})$ is introduced in order to keep consistency. It is defined, in continuous form, as:

$$\gamma(\mathbf{x}) := \int_{\bar{\Omega}} W_h(\mathbf{x} - \mathbf{y}) d\mathbf{y}. \quad (3)$$

Or as the discretized version:

$$\gamma(\mathbf{x}_i) = \sum_{j \in Fluid} W_h(\mathbf{x}_j - \mathbf{x}_i) V_j, \quad (4)$$

where \mathbf{x}_i and \mathbf{x}_j are the positions of a generic particle and its neighbors, respectively, and V_j their volumes.

Regarding the applicability of the boundary integrals formulation, the main setback comes from the fact that the Shepard renormalization factor, $\gamma(\mathbf{x})$, breaks the symmetry form of the operators, and therefore momentum and energy conservation cannot be assured anymore. This is because, at the boundary, $\gamma(\mathbf{x}) \neq \gamma(\mathbf{y})$ when $\mathbf{x} \neq \mathbf{y}$. Additionally, in multiphase flows, the Shepard renormalization factor integral domain near the free surface boundary should consider all fluid domains in order to be consistent with boundary integrals formulation. However, it is common practice to only model the heavier phase, which is generally the most relevant. This reduces computational effort and has indeed become one of the greatest advantages of SPH for free surface flow simulations. On the other hand, the domain at the free surface is then incomplete,

which causes a wrong computation of the Shepard renormalization factor. In order to avoid this issue, the correct procedure, as argued in Ref. [10], would be to explicitly model the lighter phase.

Another problem of the methodology occurs when the boundary integrals formulation moves on to the discrete level, i.e. when Eq. (4) is applied. In this context, even far from the boundary, $\gamma(\mathbf{x}_i) \neq \gamma(\mathbf{y}_j)$ when $\mathbf{x}_i \neq \mathbf{y}_j$, which may lead to values of the Shepard renormalization factor bigger than one. It might also happen that tensile instabilities appear because of the nature of the SPH interpolating kernels close to boundaries. Issues derived from these instabilities, such as sensitivity to particle disorder, clamping, or clustering clearly affect the computation of the Shepard factor, and might end up in unacceptable Shepard values.

Given all these facts, this work aims to develop a new boundary integrals formulation that overcomes these disadvantages by defining a new integral at the boundaries that allows to transform the whole volume integral into a surface boundary integral.

2.1. Alternative geometrical formulation of the Shepard renormalization factor

A methodology to transform the volume integral to compute the Shepard renormalization factor (4) into a surface integral was described in Ref. [16], extended later as a semi-analytical method for a generic 2-D profile in Ref. [21] and finally incorporated in Ref. [29] to 3-D applications. However, both approaches can be unified as described in this Section.

Notice that the kernel is normalized, and we may integrate it by splitting its support as

$$\int_{\bar{\Omega}} W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y} + \int_{\Omega^*} W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y} = 1. \quad (5)$$

From the definition of the Shepard factor, Eq. (3):

$$\gamma(\mathbf{x}) + \int_{\Omega^*} W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y} = 1. \quad (6)$$

Our goal is to express the Shepard renormalization factor purely as a surface integral. If we can find a function $F(\mathbf{y} - \mathbf{x})$ such that

$$\int_{\Omega^*} \nabla \cdot [(\mathbf{y} - \mathbf{x}) F(\mathbf{y} - \mathbf{x})] = \int_{\Omega^*} W_h(\mathbf{y} - \mathbf{x}) d\mathbf{y}, \quad (7)$$

then, $\gamma(\mathbf{x})$ may be obtained from a boundary integral through the application of the divergence theorem:

$$\gamma(\mathbf{x}) = 1 + \int_{\partial\bar{\Omega}} \mathbf{n}(\mathbf{y}) \cdot (\mathbf{y} - \mathbf{x}) F(\mathbf{y} - \mathbf{x}) d\mathbf{y}. \quad (8)$$

We may therefore formulate $\gamma(\mathbf{x})$ in terms of surface integrals, which are purely geometrical, unlike volume integrals, which may change in time since the particles will move. They will also be clearly independent on any free surfaces that may appear, whilst volume integrals would be affected by the integration over an area with no particles.

Since the definition of Eq. (8) must apply for every integral, the general solution is given by the equality of the integrands. Moreover, as the kernel W_h depends only on $\rho := |\mathbf{y} - \mathbf{x}|$, the distance between \mathbf{x} and \mathbf{y} , we may seek a F that also depends only on ρ , which leads to

$$\frac{1}{\rho^{d-1}} \frac{d(\rho^d F(\rho))}{d\rho} = W_h(\rho), \quad (9)$$

where d is the spatial dimension.

The solution to Eq. (9) depends, of course, on the choice of kernel. For the Wendland kernel [30], and $s = 2$, expressions for $F(\rho)$ in reduced (non-dimensional) form are given by:

$$F(\rho) = \frac{1}{h^d} \tilde{F}\left(\frac{\rho}{h}\right), \quad (10)$$

with the function of the reduced distance $q := \rho/h$, expressed in 2-D as

$$\begin{aligned} \tilde{F}(q) = & -\frac{7}{64\pi} \left[\frac{2}{7}q^5 - \frac{5}{2}q^4 + 8q^3 - 10q^2 + 8 \right] \\ & - \frac{1}{2\pi q^2}, \end{aligned} \quad (11)$$

and in 3-D:

$$\begin{aligned} \tilde{F}(q) = & -\frac{21}{256\pi} \left[\frac{1}{4}q^5 - \frac{15}{7}q^4 + \frac{20}{3}q^3 - 8q^2 + \frac{16}{13} \right] \\ & - \frac{1}{4\pi q^3}. \end{aligned} \quad (12)$$

Both expressions in Eqs. (11) and (12) can in fact be separated into a polynomial part and a divergent part, that we will call from now on \tilde{F}_P and \tilde{F}_D respectively. Therefore, in order to be consistent with the notation $F(\rho) = F_P(\rho) + F_D(\rho)$. This implies a set of restraints and a different treatment of the kernel function that will be discussed in next section.

2.2. Efficient evaluation of γ

The effect of having a singular kernel – such as $F(q)$ – is especially significant near a boundary, where it has a greatest influence on the particle interpolation. Nonetheless, there exist several alternatives to deal with this fact. From all the options that could be considered, increasing the resolution might be the more straight-forward approach. However, results are still not assured, while moving on to an increase above 50 times the fluid particle resolution results in an unacceptable increase of the computational cost, keeping in mind that a fast and reasonably accurate solution is being sought. To show it, Figure 2 illustrates how a resolution increase affects the Shepard renormalization factor $\gamma(\mathbf{x})$ compared to the theoretical solution for a fluid particle approaching a 90° corner along one of the boundaries. The divergent term of the kernel has an influence on the last particle interpolation, giving a wrong value of the Shepard renormalization factor which is independent of the resolution. The dashed line in Figure 2 represents the expected error derived

from this last particle interpolation. As it can be appreciated, the error is nearly constant, in this case with a value of about $C = 0.021$. Since the value is the same for different resolutions, it can be guessed that the error follows the Shepard curvature.

Alternatively, Eq. (8) could be solved analytically. Such approach has been already considered in [29], leading however to involved expressions that, again, would be too costly from the computational point of view. Finally, another possibility, based on the previous one, would consist on solving once for multiple situations and tabulate the results. Unfortunately, a very high resolution would be needed again to obtain an accurate enough solution. It seems that none of these options are able to improve the solution. There is, however, another one, to be analyzed. Re-writing the expression in Eq. (8) for a boundary that is split into different S_j patches yields:

$$\gamma(\mathbf{x}) = 1 + \sum_j \int_{S_j} \mathbf{n}(\mathbf{y}) \cdot (\mathbf{y} - \mathbf{x}) F(\mathbf{y} - \mathbf{x}) d\mathbf{y}. \quad (13)$$

If the curvature of the patches is neglected, they may be approximated by flat segments, so that normals are constant, as depicted in Figure 3. In this case,

$$\gamma(\mathbf{x}) - 1 \simeq \sum_j \mathbf{n}_j \cdot (\mathbf{y}_j - \mathbf{x}) \int_{S_j} F(\mathbf{y} - \mathbf{x}) d\mathbf{y}. \quad (14)$$

The expression $\mathbf{n}_j \cdot (\mathbf{y}_j - \mathbf{x})$, with $\mathbf{y}_j \in S_j$, is taken out of the integral since, for a flat patch, it equals r_j , the distance between \mathbf{x} and the patch. The expression above is in fact similar to the ones proposed in Ref. [17].

Finally, going back to Eqs. (11, 12), it has already been noticed that the kernel can be decomposed into a polynomial term, F_P , and a divergent term F_D .

The integral of the polynomial term will still be solved numerically within the traditional boundary integrals formulation —either the semi-analytical [17] or the purely numerical [9] one, whereas the divergent term is solved analytically. In Appendix A it is shown that, based on the previous approximations, one can get in 2-D to:

$$\mathbf{n}_j \cdot (\mathbf{y}_j - \mathbf{x}) \int_{S_j} F_D(\mathbf{y} - \mathbf{x}) d\mathbf{y} = -\frac{1}{2\pi} \Delta\theta_j, \quad (15)$$

and in 3-D to:

$$\mathbf{n}_j \cdot (\mathbf{y}_j - \mathbf{x}) \int_{S_j} F_D(\mathbf{y} - \mathbf{x}) d\mathbf{y} = -\frac{1}{4\pi} \Delta\Omega_j. \quad (16)$$

In 2-D, $\Delta\theta_j$ is the angle subtended by segment S_j at point \mathbf{x} , which can be easily computed. Similarly, in 3-D $\Delta\Omega_j$ is the solid angle subtended by patch S_j at point \mathbf{x} .

The approach described in [24], where Eq. (14) is directly addressed, yields a complex and costly formulation. That is partially solved in [29], where the same expression (14) is addressed, applying this time

the Gauss’ theorem to transform it in a line integral, which fits the semi-analytical formulation described in [17], but has a poor performance in purely numerical approaches [9]. In Appendix B efficient ways to evaluate the subtended angle from Eqs. (15, 16), both in semi-analytical and purely numerical contexts, are described. Incidentally, we may emphasize that different expressions can be found, depending on the arbitrary discretization patch shape selected, in a similar fashion of mesh-based methods [27]. In these algorithms it is assumed that the boundary is discretized in straight line segments for 2-D applications, while for 3-D applications different discretisations have been considered: triangles for the semi-analytical approach, and squares for the purely numerical one. Indeed, in the semi-analytical approach the computational overhead of the connectivities is generally accepted, taking advantage of the well-known surface triangulation properties [14]. Conversely, in the purely numerical approach the boundary is discretized in square patches, as a natural 2-D extension of the volumetric discretization usually applied in SPH. Comparing the computational performance and results quality of both approaches is out of the scope of this work. Nevertheless, an attempt to compare the performance of the new formulation and the traditional approach within the methodology of numerical boundary integrals is carried out in Section 3.2.

It is noteworthy that Eq. (15) in 2-D correctly predicts that for a point, \boldsymbol{x} , on one of the boundary patches,

$$\gamma(\boldsymbol{x}) - 1 \rightarrow -\frac{1}{2\pi}\pi = -\frac{1}{2}, \quad (17)$$

as long as only the patch containing x contributes to the summation (because all the others are either far away, or aligned with the patch). This is the expected result: $\gamma(\boldsymbol{x}) = 1/2$ close to a flat boundary. The same holds true in 3-D applying Eq. (16).

Therefore, if a point is detected at the contour, then it is just necessary to add $-\frac{1}{2}$ to the integration, such that the point itself will be ignored in the numerical integration. Performance of Eq. (14) has been tested and the result is shown in Figure 4, which shows the Shepard renormalization factor value for a fluid particle approaching a horizontal wall, from both the monolithic expression of the new kernel ($F(\boldsymbol{q})$) —i.e. the straight forward approach—, and from expressions (15) and (16) —in which the divergent part of the integral is solved analytically—. Both are seen to perform well far enough of the wall, whilst their behavior starts to differ when they are very close to the wall. Here is where the divergent part weights more and therefore, having an exact expression for this part makes the solution to be accurate near the wall. Otherwise, the increment in the particle resolution would need to be huge to diminish the error near the wall.

This derivation is extensible to any kind of planar boundary, including any angled corner. Figure 5 shows the results for a 90° corner. Performance is similar to the one shown in Figure 4, where a straight boundary is considered. In fact, results in Figure 5 (left) are equivalent to the test results shown in Figure 2. It can be appreciated that the correct value of $\gamma(\boldsymbol{x}) = 0.25$ is now reached at the corner when the divergent part

of the kernel is integrated exactly with the new formulation.

The new formulation is not only suitable for planar boundaries but for general boundaries indeed. Figure 6 represents the Shepard value at a circular boundary for a particle approaching radially from the center of the domain. Performance is seen to be similar to the previous tests shown. The new formulation predicts correctly the value when approaching the boundary even having considered flat segments. Again, the monolithic formulation is not able to capture the correct value of the Shepard renormalization factor close to the boundary due to the effect of the singularity in the kernel.

In order to assess the accuracy of the method as a function of the $h/(\Delta r)$ ratio, which controls the number of neighbors in an SPH calculation, we select the point closest to the corner of Figure 5 (the one with the worst results), and evaluate the relative error in the Shepard value, as this ratio is decreased. In Figure 7 the monolithic expression is seen to improve slightly as fewer neighbors are considered. Meanwhile, the semianalytic method is seen to result in much lower errors. Both methods are affected by fluctuations as the number of neighbors becomes low. It is obvious that these are caused by the part that is common to both methods: the polynomial. Also, the monolithic part is well fit by the expression with a $C = 0.021$ displacement mentioned above and in Figure 2.

3. Applications

3.1. Hydrostatic 2-D case

With the results shown in Section 2.2, it is expected that some current drawbacks of the boundary integrals formulation are avoided and results improve for any kind of geometry that is tested. In order to carry out a more relevant test, a still liquid test case has been carried out with the open-source free tool AQUAgnusph [9], with both the standard formulation and the new formulation.

A weakly-compressible δ -SPH model is solved. The δ -SPH scheme [1, 3, 7] is considered in order to avoid numerical instabilities. The standard Navier-Stokes continuity and momentum equations for a barotropic fluid are solved including diffusive terms for both equations. See for instance Refs. [1, 6]. A stiffened linear Equation of State, as in Refs. [2, 7] is used.

The case consists of a tank set at rest for a certain period of time. The geometry is shown in Figure 8. The aim is to assess the performance of the new formulation introduced in previous section, to check whether the new boundary term properly computes the integrals. This is a relatively simple test that nevertheless allows an easy assessment of the features that should be improved. In particular, errors in the consistency of the operators, which may lead to spurious velocities and nonphysical pressure values, and to the wrong computation of forces in the tank.

Moreover, problems mainly associated to the escape of particles from the boundary domain, which can occur with the current formulation, are expected to be avoided. In addition to that, the traditional

formulation makes it impossible to properly correct terms involving the divergence of the velocity, due to inconsistencies that arise in the presence of a free surface. This issue is now fixed, since with the new formulation the divergence of the velocity term is appropriately renormalised with the Shepard factor term.

Density is set as $\rho = 998\text{kg/m}^3$ and dynamic viscosity μ as the standard value $\mu = 8.94 \times 10^{-4}\text{Pa} \cdot \text{s}$. There is therefore no artificial viscosity in the simulation. Filling level is chosen as $H = 92\text{mm}$. Three different resolutions have been tested, with simulations with 10000, 50000 and 100000 particles. Also, different supports ($h/dr = 2, 3$ and 4) have been tried for the finest resolution. No boundary forces have been implemented to stop fluid particles from penetrating the solid walls.

Figure 9 shows the tank at rest for the same simulation time, $t\sqrt{g/H} = 9.32$. The traditional formulation of the Shepard renormalization factor leads to numerical errors in the interpolation. As can be clearly appreciated, these errors make the particles at the free surface move, with some of them jumping out of the tank. The simulation then turns unstable, even at early stages. Conversely, the new formulation computes the value of the Shepard renormalization factor exactly at the boundary, and therefore, particles remain without spurious movement and the simulation remains stable in time.

Besides such gross scale instability-related issues, close to the contact line, Shepard renormalization factor errors can be appreciated all along the free surface, when the original expression from Eq. (4) is considered. Furthermore, it can be appreciated that the original Shepard formulation is prone to produce wrong values, $\gamma(\mathbf{x}) > 1$, when tensile instabilities occur.

In order to deeper analyze the differences between both formulations, Figure 10 presents the kinetic energy evolution for three different resolutions. As can be clearly appreciated, in the traditional formulation kinetic energy increases until the simulation fails. This time is shortened if the number of particles increases. Regarding the new formulation, the maximum value slightly increases as the number of particles increase, however its value is negligible and constant in time.

Figures 11 and 12 show the error in the computation of the hydrostatic pressure according to the L_2 error of the pressure field (compared with the correct one), normalized by the maximum hydrostatic pressure $\rho g H$. Figure 11 compared the traditional and the new formulations for three different resolutions, whilst Figure 12 represents the same two formulations for the biggest resolution and three different supports.

In Figure 11 results are seen to follow the same trends as the kinetic energy. Errors in the pressure field start to rise until the simulation fails. The failure occurs earlier for finer resolutions. Conversely, the new formulation has a relatively constant error that diminishes as the resolution increases, being almost the same for the two finer resolutions tested.

The same behavior is found when testing different supports. Whilst the traditional formulation tends to increase the error when support increases, in the new formulation this change is almost not noticeable, with a reasonable and constant error in time being obtained.

3.2. 3-D dam break

In the previous application the capability of the new formulation to significantly improve the results, specially close to the contact line, has been checked. However, in order to assess the performance of the new method in a more general context, the SPHERIC validation test number 2, consisting on a 3-D dam break flow, is here considered. This validation test has already been considered in a similar context in Refs. [24, 29].

The 3-D dam break initial condition is schematically depicted in Fig. 13. A reservoir of water with the shape of a rectangular box, with dimensions $1.228 \text{ m} \times 1 \text{ m} \times 0.55 \text{ m}$, is initially set in hydrostatic equilibrium at one side of the tank, which has dimensions $3.22 \text{ m} \times 1 \text{ m} \times 1 \text{ m}$. A fixed box of dimensions $0.161 \text{ m} \times 0.403 \text{ m} \times 0.161 \text{ m}$ is also placed inside the tank.

Experiments on this test case have been described in Ref. [18]. In such experiments the wave height is quantitatively measured in the vertical probes labeled H_1 - H_4 . An additional set of pressure sensors are distributed along the fixed inner box, as schematically depicted in Fig. 14.

For the sake of simplicity, herein we focus in the first impact of the flow against the inner box, and more specifically in the pressure records measured in the sensors P_1 - P_4 .

To carry out the simulations, a numerical sound speed of $c_0 = 40 \text{ m/s}$ is considered, as well as the typical water density and viscosity values, $\rho_0 = 998 \text{ kg/m}^3$ and $\mu = 10^{-6} \text{ m}^2 / \text{s}$. The δ -SPH formulation [1] is applied for stability, following Ref. [15] (see also Ref. [7], where several δ -SPH term alternatives are discussed.) This is in contrast with the simulations carried out in Refs. [29] and [24], where a large artificial viscosity was considered in order to preserve the stability. In these works, experimental records were not included, and SPH results were compared against a Volume-of-Fluid Finite Volume simulation.

No-slip boundary conditions have been imposed along the solid walls. However, in this practical application a boundary force is added as well, to avoid walls penetration. It is triggered when a fluid particle moves closer than $0.1 \Delta r$ to the wall. Even though such a boundary force has a detrimental effect in the pressure record noise, it is nevertheless required to avoid dramatically small time steps.

The air phase is neglected, for optimization purposes, assuming the inconsistencies close to the free surface discussed in Refs. [10] and [11] which have been also discussed previously in Section 2. Besides the consistency issues close to the free surface, neglecting the air phase may induce some other errors in the pressure records, as already mentioned in Ref. [24].

In order to analyze the effect of the discretization on the results, several initial particle spacings have been considered, $\Delta r = 0.55/30 \text{ m}$, $0.55/45 \text{ m}$, and $0.55/60 \text{ m}$, with a constant kernel length ratio, $h/\Delta r = 4$. Along the same line, several kernel lengths have been simulated, with a spacing of $\Delta r = 0.55/45 \text{ m}$, namely $h/\Delta r = 2, 3$ and 4 .

In Fig. 15, the pressure records computed by AQUA_gpusph [9] are compared with the experimental ones, for different initial particle spacing values.

Pressure records are obtained with a 1000 FPS sampling rate. On top of that, high frequency noise has been filtered out by means of a Savitsky-Golay filter [26].

After time $t = 0.5$ s, the pressure record associated to spacing $\Delta r = 0.55/45$ is significantly affected by a ~ 100 Hz frequency noise signal, whose source is not clear.

Nevertheless, a good agreement in the pressure measured at the pressure probes P_1 and P_2 is achieved for all the simulations, considering that SPH tends to overestimate the pressure peak at sensor P_1 . The same cannot be said for the pressure sensors P_3 and P_4 , where the phenomenon is poorly captured in the simulation.

The overestimated pressure peaks, as well as the errors at P_3 , have been attributed in the past to the absence of the air phase, see e.g. [24]. Unfortunately, the P_4 pressure record, which is consistently the one most affected by the neglect of the air phase, has been circumvented in the SPH literature [12, 20, 24, 29]. Anyway, given the accuracy of the records at probes P_1 and P_2 , and the large errors at P_3 and P_4 , it is plausible to conclude that the air phase plays an important role in the experimental pressure.

In Fig. 16 similar pressure validations are shown for different kernel length ratios and constant initial particle space, $\Delta r = 0.55/45$.

The number of neighbors plays so far a secondary roll in the computed pressure at sensors P_1 - P_2 . However, the reduction of the kernel length, and therefore the smoothing radius, improves the pressure at P_4 , specially close to the pressure peak, even if it always poorly captured in SPH.

Again, the simulation associated to the parameters $\Delta r = 0.55/45$ m and $h/\Delta r = 4$ is significantly affected by a ~ 100 Hz noise signal, which is not seen for the other lengths at all.

From the results discussed above, it is reasonable to suggest that simulations are correctly converging, both when the particles spacing is decreased and when the number of neighbors is increased.

As stated above, the computational performance can be a critical factor, as already stressed in Ref. [29], where the computation of the analytical value of γ took 40% of the overall computation time. In Table 3.2, the time required to compute a single time step, in a NVIDIA GeForce GTX 750 Ti graphics device, is presented for all the simulations described above, for the traditional and the new γ formulations. Additionally, the performance for $\gamma = 1$, i.e. when the factor is not computed at all, has been added in order to be able to evaluate the overall impact.

The traditional γ formulation takes $\sim 10\%$ of the overall computational time when large number of neighbors is considered, growing up to $\sim 16\%$ for the lowest number of neighbors, $h/\Delta r = 2$. The new formulation requires almost half the time for the largest number of neighbors, i.e. $\sim 5\%$ of the overall computation time. Such number rises to $\sim 10\%$ for the lowest number of neighbors. The general reduction of the computational cost, when the new formulation is applied, reflects the benefit of moving from a volume integral to a surface one.

For completeness, it is interesting to compare the results obtained with the new formulation, with the

Table 1: Time in seconds to compute a single time step, averaged along the first 100 time steps.

Δr (m)	$h/\Delta r$	Former formulation	New formulation	$\gamma = 1$
0.55/30	4	0.89	0.84	0.80
0.55/45	4	2.95	2.82	2.68
0.55/60	4	7.24	6.93	6.62
0.55/45	2	0.50	0.48	0.43
0.55/45	3	1.37	1.28	1.18

ones obtained in [20], where incompressible-SPH (ISPH) was applied. In Fig. 17 the pressure records at the sensors P_1 and P_3 , for the ISPH simulation and the WCSPH simulation with the new formulation are depicted, and compared with the experimental results. In this particular case, WCSPH has been able to capture significantly better the flow impact phenomena.

4. Conclusions

In the present work, a new kernel formulation is presented with the aim of reducing consistency issues for the boundary integrals formulation when free surfaces and boundaries are present in the interpolation. The kernel is defined in such a way that the whole volume integral is cast into a surface integral when boundaries are present, thus formulating the Shepard renormalization factor as a purely geometrical factor.

The expression which makes possible to write the Shepard factor as a function only of the geometry has a disadvantage: the kernel expression includes a divergent term which might cause problems when dealing with particles very close to the boundary. However, it is possible to split the expression into a numerical part, which is formed by a polynomial expression, and solve the divergent term analytically. The analytical term consists basically on a function of the subtended angle, which can be easily computed both in 2-D and 3-D

The new formulation has been tested against different planar boundaries, including corners, showing that the Shepard is computed correctly at the boundary. The new geometrical factor has been included into the tool AQUA_gpusph and tested for two different applications. First, a 2-D hydrostatic stationary tank simulation is carried out in order to assess the performance of the new formulation and see if the boundary term computes properly. Errors in the consistency of the operators that lead to spurious velocities and wrong pressure values in the traditional formulation are now minimized. Different resolutions and compact supports have been tried and compared to the results obtained with the traditional formulation. Second, a 3-D case is performed to show the applicability of the new formulation in a more general context, consisting on a 3-D dam break. The pressure value at four different points is presented and compared to other SPH formulations and the experimental results, showing a reasonable behavior for two of the probes and not as

good for the other two, although this is in accordance with previous SPH simulations carrying out this test. Also, computational performance is studied by comparing the time to compute a single time step that both formulations employ. Results show a general reduction in the computation time, reflecting the benefits of moving from a volume integral to a surface integral approach.

Overall, a potentially powerful formulation for computing the Shepard renormalization factor with application to the boundary integrals formulation is presented, showing that it can be easily implemented both in 2-D and 3-D and that is applicable to a wide range of engineering problems.

Acknowledgments

The authors would like to thank Prof. Antonio Souto-Iglesias for his enthusiasm and the interesting discussions, and for his valuable help pushing this work forward.

This research has received funding from Universidad Politécnica de Madrid under a pre-doctoral scholarship.

Appendix A. Expressions for the divergent part of F

It is worth mentioning that the expressions for the divergent part of F (F_D), i.e. the last terms in Eqs. (11, 12) are not limited to a particular kernel choice. Indeed, by its definition in Eq. (9), F is found by solving

$$[\rho'^d F(\rho')]_{\rho^d F(\rho)}^0 = \int_{\rho}^{sh} \rho'^{d-1} W_h(\rho') d\rho',$$

where the integration limits guarantee that F will have the same support as W_h .

If we consider the $\rho \rightarrow 0$ limit, this yields

$$-\rho^d F_D(\rho) = \int_0^{sh} \rho'^{d-1} W_h(\rho') d\rho'.$$

But, due to the kernel normalization, the right-hand side integral is $1/2\pi$ in 2-D, or $1/4\pi$ in 3-D. From this fact, the same expressions for F_D will arise in general. Notice that $\rho F_D(\rho)$ is basically the Coulomb field of a point particle, hence what is derived here closely follows the demonstration of Gauss' Law in electrostatics.

In Figure A.18 (left) the 2-D situation is sketched for a generic patch (index “ j ” will be dropped, for the sake of readability). The distance $\mathbf{y} - \mathbf{x}$ can be decomposed in a constant normal distance to the patch (r), and a component tangent to the wall (t), which is the integration variable.

The integrals that appear in (14) are of the form

$$\begin{aligned} \mathbf{n} \cdot (\mathbf{y}_j - \mathbf{x}) \int_S F(\mathbf{y} - \mathbf{x}) d\mathbf{y} &= r \int_S F(\rho) dt = \\ &= -\frac{r}{2\pi} \int_S \frac{1}{\rho^2} dt. \end{aligned}$$

It is straightforward to demonstrate this equality,

$$\frac{r}{\rho^2} dt = d\theta$$

(an expression which also appears in the calculations of the electrostatic field of a charged thin rod) from which Eq. (15) results.

Figure A.18 (right) sketches the 3-D situation. In this case, the integrals have this form

$$\begin{aligned} \mathbf{n} \cdot (\mathbf{y}_j - \mathbf{x}) \int_S F(\mathbf{y} - \mathbf{x}) d\mathbf{y} &= r \int_S F(\rho) dS = \\ &= -\frac{r}{4\pi} \int_S \frac{1}{\rho^3} dt. \end{aligned}$$

In the same manner, we may demonstrate

$$\frac{r}{\rho^3} dS = d\Omega,$$

this is, the differential of a solid angle. This way, Eq. (16) is obtained.

Appendix B. Efficient evaluation of the angle subtended by a patch

As discussed above, the alternative Shepard renormalization factor boils down to the computation of a convolution boundary term and an angle subtended by a boundary patch. The former convolution boundary term, associated to the non-singular part of kernels in Eqs. (11, 12), can be in fact computed applying the boundary integrals approach described in Refs. [17] and [8]. However, this is clearly not the case of the angle subtended by boundary patches.

As the boundary integrals methodology itself, the angle subtended by a boundary patch computation admits both a semi-analytical approach [17] and a purely numerical one [8], being the latter more efficient but less precise than the former. Both approaches will be discussed here, in order to highlight their similarities and differences.

Appendix B.1. Semi-analytical methodology

When the semi-analytical boundary integrals methodology is applied, as described in [17], the boundary is discretized in planar patches, each defined by its vertices and normal vector, as it is schematically depicted in Fig. B.19.

Indeed, in 2-D simulations the boundary is discretized in line segments, such that the tangent distance of a generic j^{th} vertex,

$$r_j^t = |(\mathbf{y}_j - \mathbf{x}) - \mathbf{n}_j (\mathbf{n}_j \cdot (\mathbf{y}_j - \mathbf{x}))|,$$

can be used to compute the angle subtended by the generic j^{th} line segment:

$$\Delta\theta_j = \tan^{-1} \left(\frac{r_{j+1}^t}{r} \right) + s_t \tan^{-1} \left(\frac{r_j^t}{r} \right),$$

with

$$s_t = \text{sign}(|\mathbf{y}_j - \mathbf{x}| - (r_{j+1}^t - r_j^t)).$$

Even though 3-D boundaries can be discretized by a wide variety of planar patches, only triangulations will be discussed here. The expression to compute angles subtended by triangular plates, discussed in [28], can be applied:

$$\begin{cases} \mathbf{a} = \mathbf{y}_j - \mathbf{x}, \\ \mathbf{b} = \mathbf{y}_{j+1} - \mathbf{x}, \\ \mathbf{c} = \mathbf{y}_{j+2} - \mathbf{x}, \\ \Delta\Omega_j = \frac{\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})}{abc + a(\mathbf{b} \cdot \mathbf{c}) + b(\mathbf{a} \cdot \mathbf{c}) + c(\mathbf{a} \cdot \mathbf{b})} \end{cases}$$

Appendix B.2. Purely numerical methodology

In the purely numerical boundary integrals methodology (details of the implementation can be found in Refs. [8, 9]), the same discretization applied to the volume particle is naturally extended to the boundary, i.e. the boundary is sampled by boundary elements associated to a boundary area portion. Indeed, mesh connectivity between elements is not required any more, which permits a substantial improvement of computational performance. On the other hand, truncation errors are introduced, meaning lower quality results.

In fact, the 2-D implementation is rather similar to the semi-analytical one, discussed in Appendix B.1. The boundary is discretized in line segments that, in contrast to the semi-analytical approach, are defined by their center \mathbf{y}_j , area S_j and normal \mathbf{n}_j , as schematically depicted in Fig. B.20 (left). Analogously, we can compute the tangential distance to the generic j^{th} boundary element center,

$$r_j^t = |(\mathbf{y}_j - \mathbf{x}) - \mathbf{n}_j (\mathbf{n}_j \cdot (\mathbf{y}_j - \mathbf{x}))|,$$

with an expression for the angle subtended by the boundary element similar to the one already found for the semi-analytical methodology,

$$\Delta\theta_j = \tan^{-1}\left(\frac{r_j^t + S_j/2}{r}\right) - \tan^{-1}\left(\frac{r_j^t - S_j/2}{r}\right).$$

The 3-D case is much more complex. The boundary is then discretized in square area elements, defined by their center \mathbf{y}_j , area S_j , normal \mathbf{n}_j as schematically depicted in Fig. B.20 (right). A tangent vector \mathbf{t}_j is considered, whose choice does not affect the results, as long as it is contained in the plane of the square. To compute the solid angle subtended by the boundary element we can make use of the well known expression of the solid angle subtended by a rectangular patch in which one corner coincides with the projection of the origin, \mathbf{x} , onto the plane:

$$\Omega(a, b, r) = \cos^{-1}\left(\sqrt{\frac{1 + (a/r)^2 + (b/r)^2}{(1 + (a/r)^2)(1 + (b/r)^2)}}\right),$$

with a and b being the width and height of the rectangular patch respectively.

To this end, we can compute the tangential distances,

$$\begin{aligned} r_j^t &= |\mathbf{t}_j \cdot (\mathbf{y}_j - \mathbf{x})|, \\ r_j^b &= |\mathbf{b}_j \cdot (\mathbf{y}_j - \mathbf{x})|, \end{aligned}$$

where \mathbf{b}_j is the binormal vector, $\mathbf{n}_j \times \mathbf{t}_j$. The subtended solid angle is subsequently computed as follows:

$$\begin{aligned} \Delta\Omega_j &= \Omega \left(r_j^t + \frac{\sqrt{S_j}}{2}, r_j^b + \frac{\sqrt{S_j}}{2}, r \right) \\ &- s_t \Omega \left(r_j^t - \frac{\sqrt{S_j}}{2}, r_j^b + \frac{\sqrt{S_j}}{2}, r \right) \\ &- s_b \Omega \left(r_j^t + \frac{\sqrt{S_j}}{2}, r_j^b - \frac{\sqrt{S_j}}{2}, r \right) \\ &+ s_t s_b \Omega \left(r_j^t - \frac{\sqrt{S_j}}{2}, r_j^b - \frac{\sqrt{S_j}}{2}, r \right), \end{aligned}$$

with

$$\begin{aligned} s_t &= \text{sign} \left(r_j^t - \frac{\sqrt{S_j}}{2} \right), \\ s_b &= \text{sign} \left(r_j^b - \frac{\sqrt{S_j}}{2} \right). \end{aligned}$$

References

References

- [1] Antuono M, Colagrossi A, Marrone S. Numerical diffusive terms in weakly-compressible SPH schemes. *Computer Physics Communications* 2012; 183(12):2570–2580.
- [2] Antuono M, Colagrossi A, Marrone S, Molteni D. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Computer Physics Communications* 2010; 181(3):532–549.
- [3] Antuono M, Marrone S, Colagrossi A, Bouscasse B. Energy balance in the δ -SPH scheme. *Computer methods in Applied Mechanics and Engineering* 2015; .
- [4] Bouscasse B, Colagrossi A, Marrone S, Antuono M. Nonlinear water wave interaction with floating bodies in SPH. *Journal of Fluids and Structures* 2013; 42:112–129.
- [5] Campbell PM. Some new algorithms for boundary value problems in smooth particle hydrodynamics. Technical report, DTIC Document, 1989.
- [6] Cercos-Pita J, Antuono M, Colagrossi A, Souto-Iglesias A. SPH energy conservation for fluid–solid interactions. *Computer Methods in Applied Mechanics and Engineering* 2017; .
- [7] Cercos-Pita J, Dalrymple R, Heralut A. Diffusive terms for the conservation of mass equation in sph. *Applied Mathematical Modelling* 2016; 40(19-20):8722–8736.
- [8] Cercos-Pita JL. AQUAgpusph, a new free 3D SPH solver accelerated with openCL. In *8th international SPHERIC workshop*. 2013; .
- [9] Cercos-Pita JL. Aquagpusph, a new free 3D SPH solver accelerated with OpenCL. *Computer Physics Communications* 2015; 192:295–312.
- [10] Colagrossi A, Antuono M, Le Touzé D. Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model. *Physical Review E* 2009; 79/5:1–13.

- [11] Colagrossi A, Antuono M, Souto-Iglesias A, Le Touzé D. Theoretical analysis and numerical verification of the consistency of viscous smoothed-particle-hydrodynamics formulations in simulating free-surface flows. *Physical Review E* 2011; 84(2):026705.
- [12] Crespo AC, Dominguez JM, Barreiro A, Gómez-Gesteira M, Rogers BD. GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods. *PloS one* 2011; 6(6):e20685.
- [13] De Lefte M, Le Touzé D, Alessandrini B. Normal flux method at the boundary for SPH. In *4th Int. SPHERIC Workshop (SPHERIC 2009)*. 2009; .
- [14] Delaunay B, et al. Sur la sphere vide. *Izv Akad Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 1934; 7(793-800):1–2.
- [15] Fatehi R, Manzari M. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Computers & Mathematics with Applications* 2011; 61(2):482–498.
- [16] Feldman J, Bonet J. Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering* 2007; 72(3):295–324.
- [17] Ferrand M, Laurence D, Rogers B, Violeau D, Kassiotis C. Unified semi-analytical wall boundary conditions for inviscid laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids* 2013; .
- [18] Kleefsman K, Fekken G, Veldman A, Iwanowski B, Buchner B. A volume-of-fluid based simulation method for wave impact problems. *Journal of computational physics* 2005; 206(1):363–393.
- [19] Kulasegaram S, Bonet J, Lewis R, Profit M. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechanics* 2004; 33(4):316–325.
- [20] Lee ES, Violeau D, Issa R, Ploix S. Application of weakly compressible and truly incompressible SPH to 3-D water collapse in waterworks. *Journal of Hydraulic Research* 2010; 48(S1):50–60.
- [21] Leroy A, Violeau D, Ferrand M, Kassiotis C. Unified semi-analytical wall boundary conditions applied to 2D incompressible SPH. *Journal of Computational Physics* 2014; 261:106–129.
- [22] Macia F, González LM, Cercos-Pita JL, Souto-Iglesias A. A boundary integral SPH formulation consistency and applications to ISPH and WCSPH. *Progress of Theoretical Physics* 2012; 128(3):439–462.
- [23] Marrone S, Colagrossi A, Antuono M, Colicchio G, Graziani G. An accurate SPH modeling of viscous flows around bodies at low and moderate reynolds numbers. *Physical Review E* 2013; 245.
- [24] Mayrhofer A, Ferrand M, Kassiotis C, Violeau D, Morel FX. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D. *Numerical Algorithms* 2015; 68(1):15–34.
- [25] Merino-Alonso P, Macia F, Souto-Iglesias A, Colagrossi A. Consistency analysis of flow field extension models into ghost fluid regions for SPH solid body boundary condition implementations. In *Proceedings of the 8th International SPHERIC Workshop, Trondheim, Norway*. 2013; .
- [26] Savitzky A, Golay MJ. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 1964; 36(8):1627–1639.
- [27] Sozer E, Brehm C, Kiris CC. Gradient calculation methods on arbitrary polyhedral unstructured meshes for cell-centered cfd solvers. In *52nd Aerospace Sciences Meeting*. 2014; page 1440.
- [28] Van Oosterom A, Strackee J. The solid angle of a plane triangle. *IEEE transactions on Biomedical Engineering* 1983; (2):125–126.
- [29] Violeau D, Leroy A, Mayrhofer A. Exact computation of SPH wall renormalising integrals in 3-D. In *Proc. 9th International SPHERIC Workshop*. 2014; .
- [30] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics* 1995; 4(1):389–396.

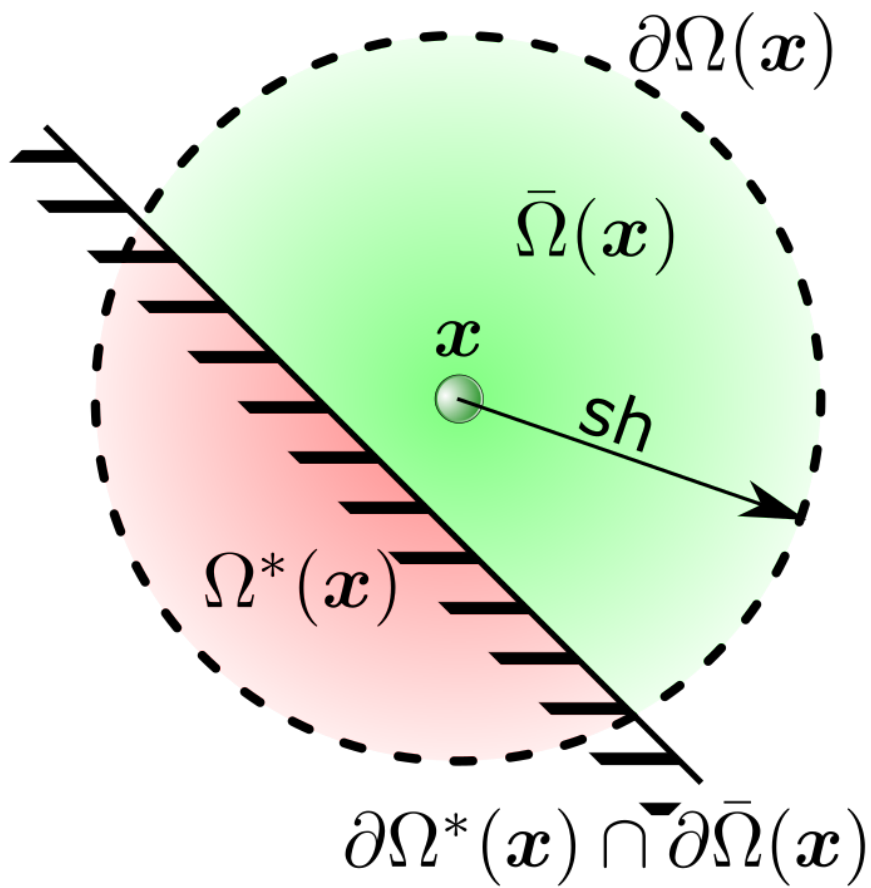


Figure 1: Schematic view of a point convolution in the presence of a boundary.

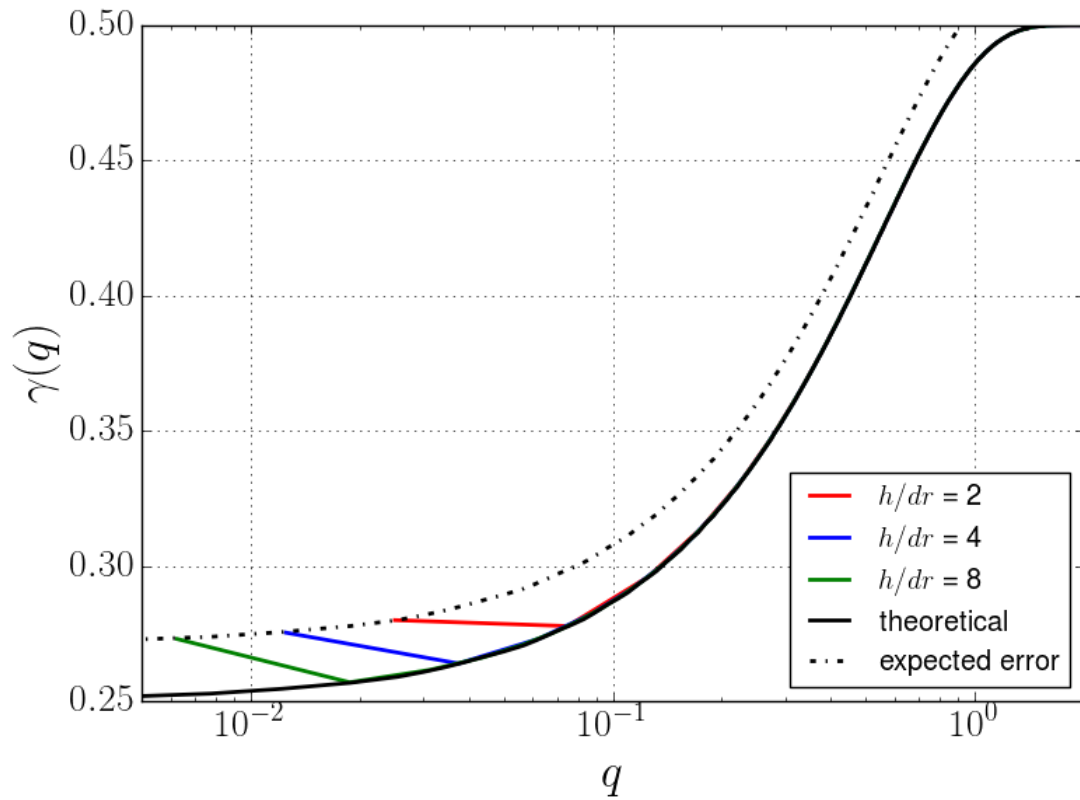


Figure 2: Shepard value at the contour near a 90° corner for different kernel supports computed with $F(q)$ and compared to the analytical value. Dashed line is the black solid line displaced upwards $C = 0.021$, and represents the expected error. Horizontal axis: normalized distance q to the corner, vertical axis: Shepard value $\gamma(\mathbf{x})$.

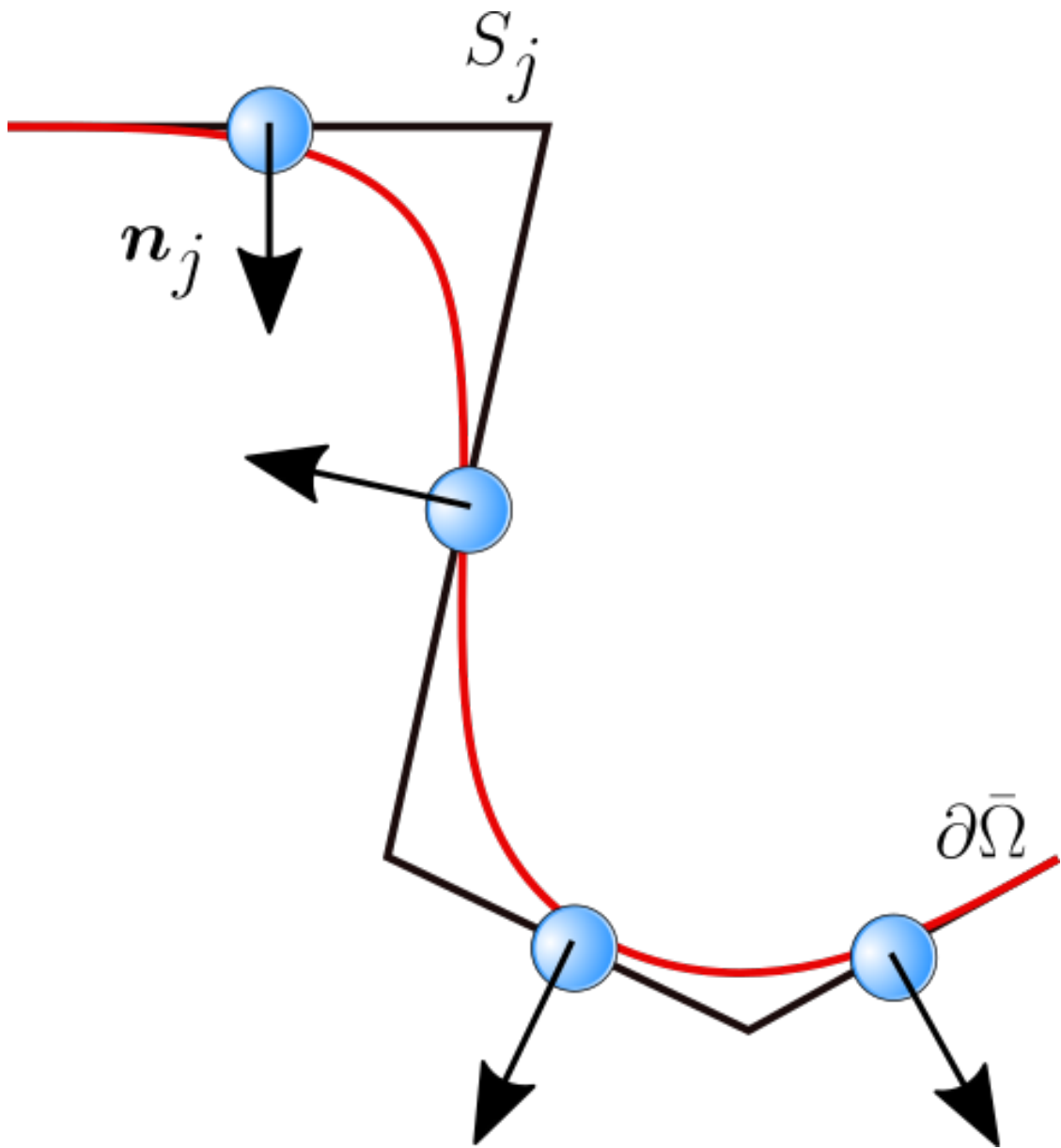


Figure 3: Discretization of the continuous boundary $\partial\bar{\Omega}$ (red line) to a set of discretized planar patches S_j (black line)

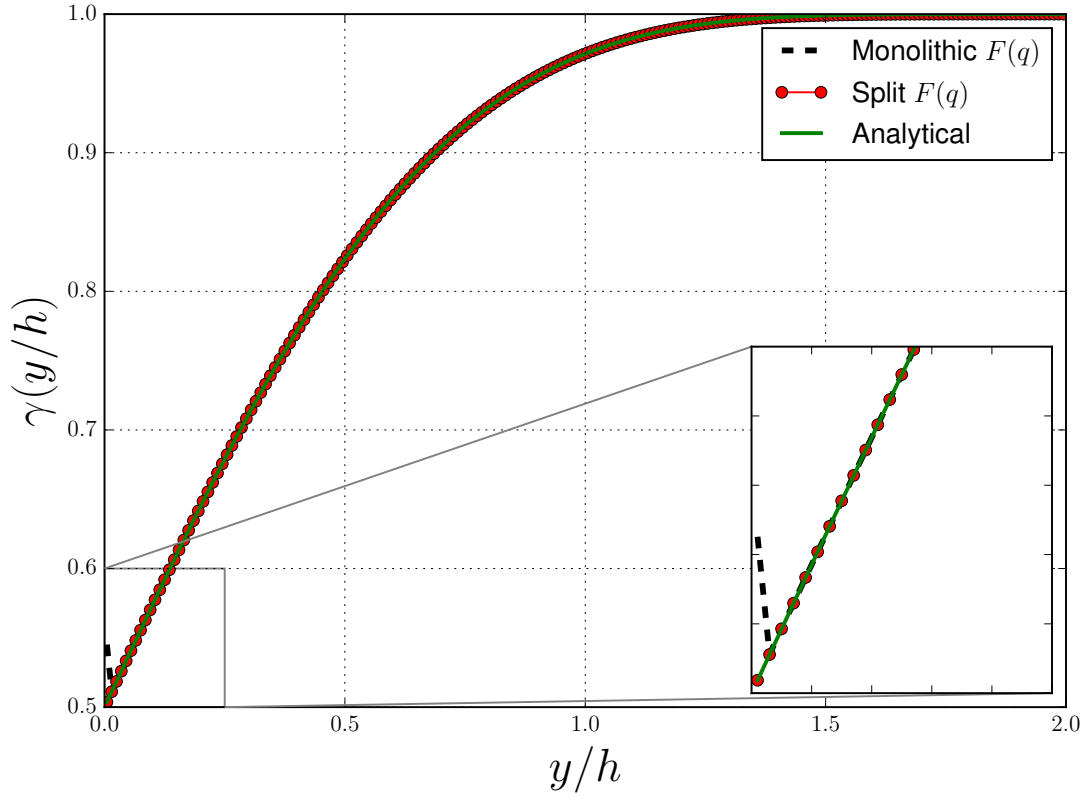


Figure 4: Shepard value for a fluid particle approaching a straight boundary. Black dashed line: monolithic expression from Eq (11), red line: the same expression, but with the semi-analytic formulation as in Eq. (15), green line: analytical value of the Shepard renormalization factor.

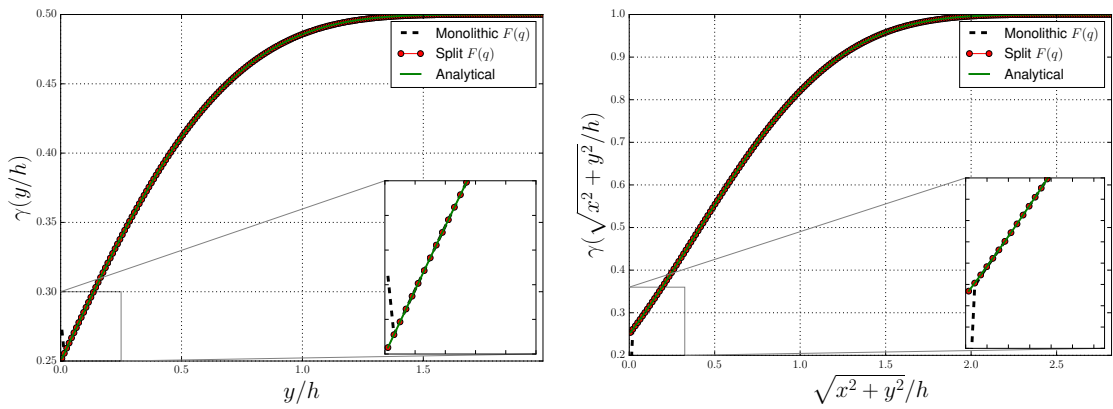


Figure 5: Shepard value for a 90° corner, for approaches along the vertical wall (left) and the diagonal (right). Legend as in Figure 4

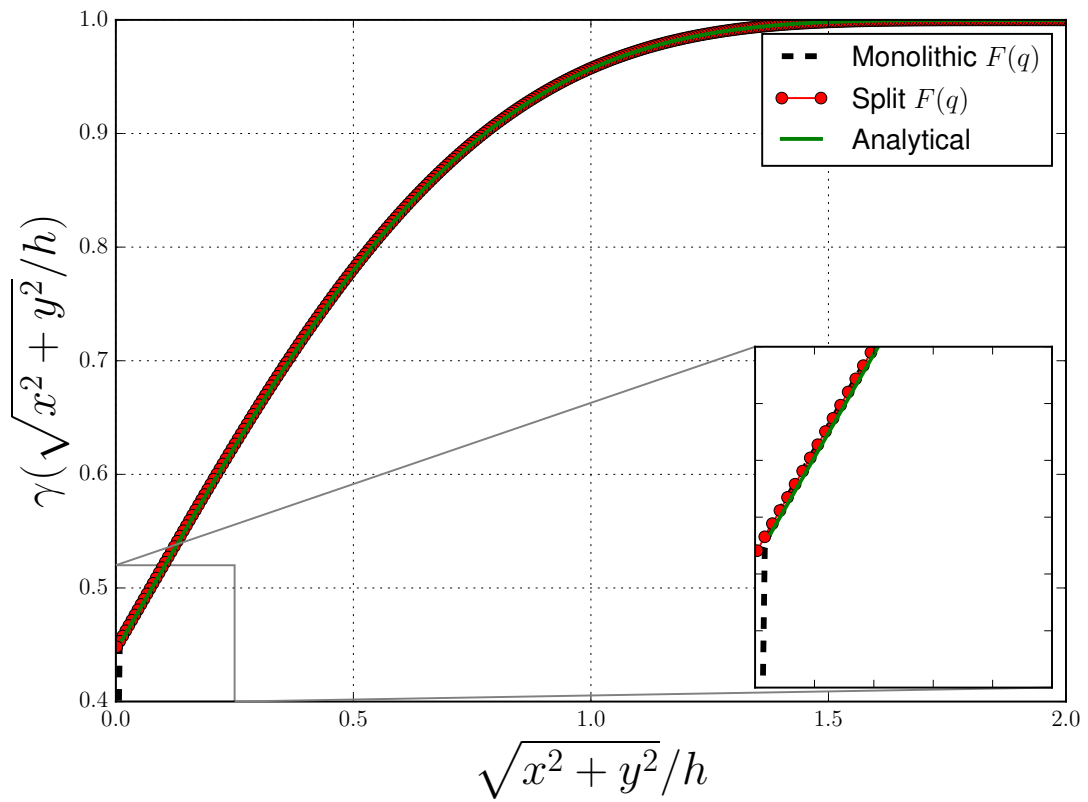


Figure 6: Shepard value for a circular boundary for a particle approaching along the radius of the circle from the center to the boundary. Legend as in Figure 4

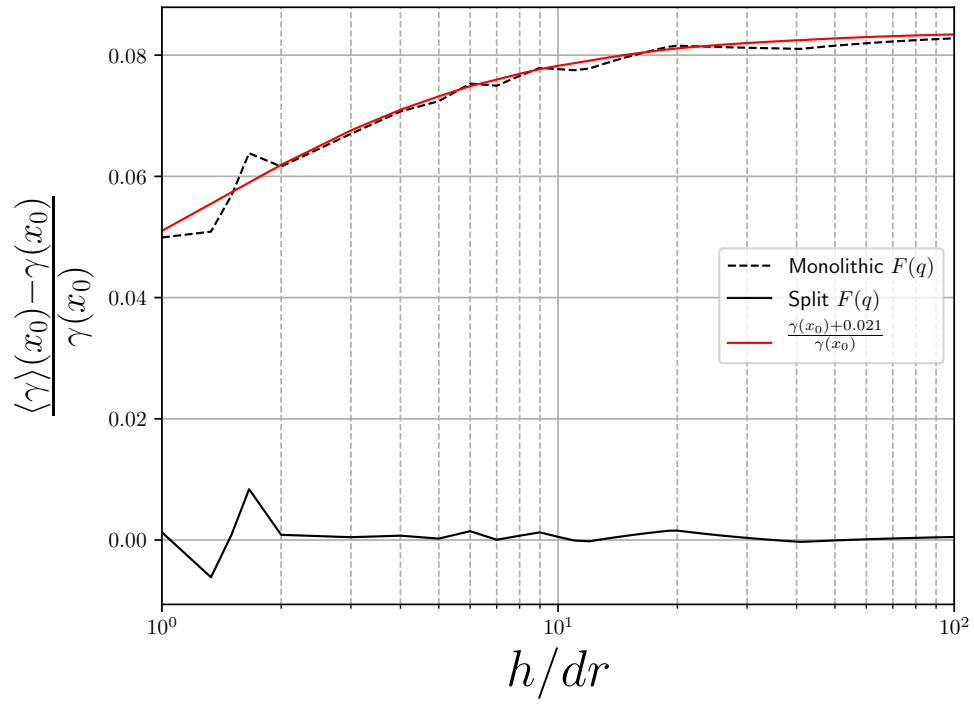


Figure 7: Relative error in the Shepard value at fixed point, close to a 90° corner as a function of $h/(\Delta r)$. Dashed line: monolithic expression; red line: approximation to the former; solid line: split expression.

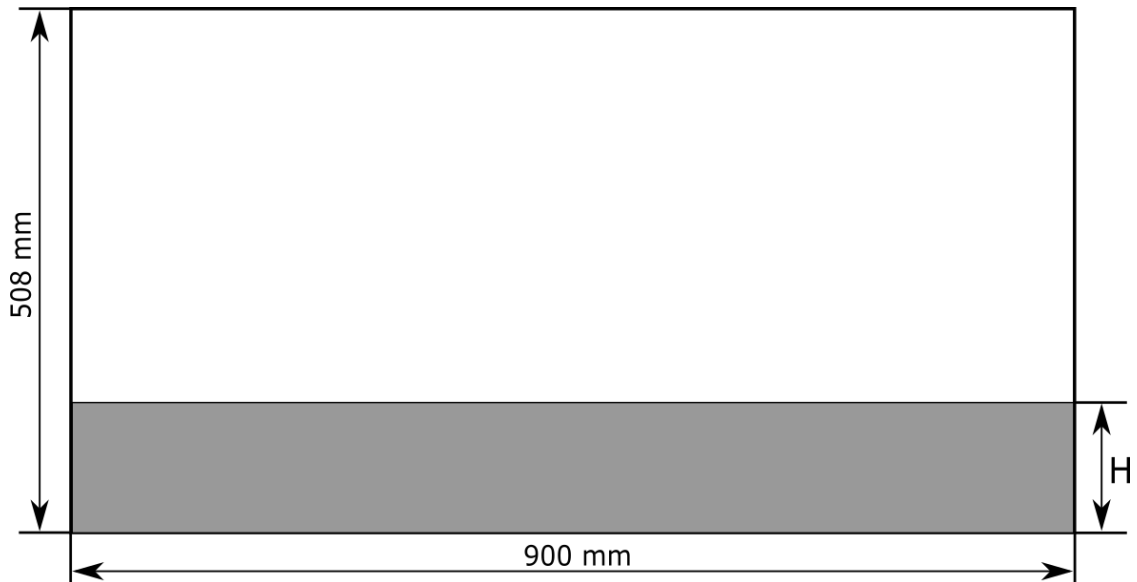


Figure 8: Geometry of the 2-D hydrostatic tank.

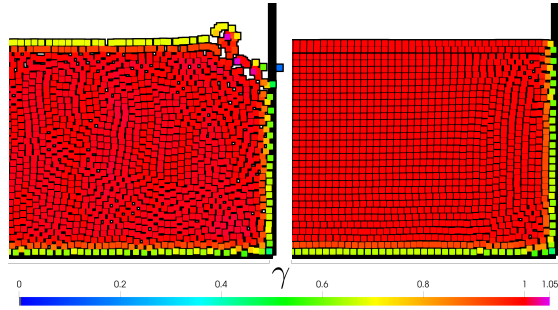


Figure 9: Detail of the Shepard renormalization factor field at the right corner of the Hydrostatic 2-D tank set at rest for the traditional Shepard formulation (left) and the new Shepard geometrical formulation (right). Compact support $h/dr = 2$ and number of particles $N = 10000$. Reduced time is $t\sqrt{g/H} = 9.32$.

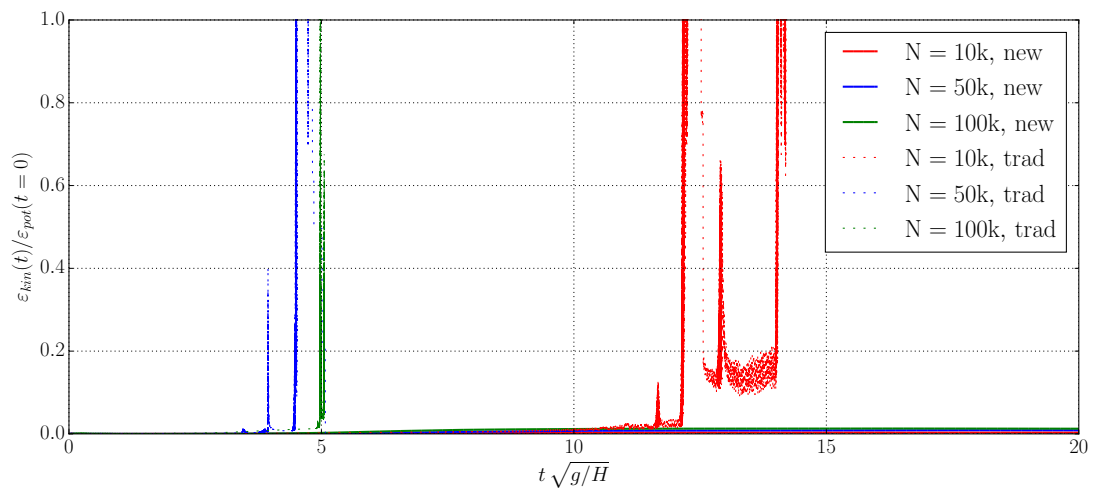


Figure 10: Evolution of the reduced kinetic energy for the hydrostatic case. Solid lines: new formulation for three different resolutions (10000, 50000 and 100000), dashed lines: traditional formulation.

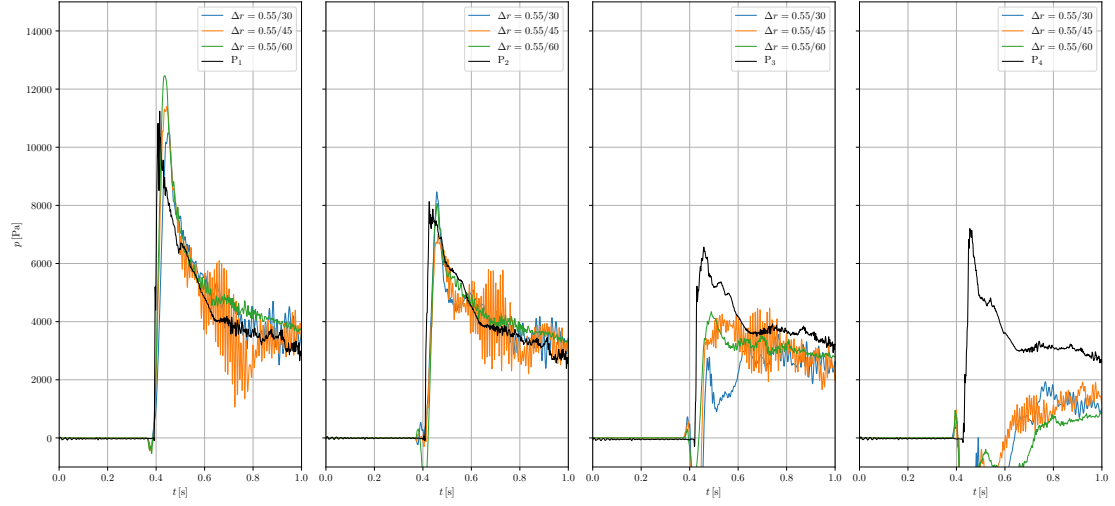


Figure 11: Evolution of the reduced L_2 hydrostatic pressure error for the hydrostatic case. Solid lines: new formulation for three different resolutions (10000, 50000 and 100000), dashed lines: traditional formulation.

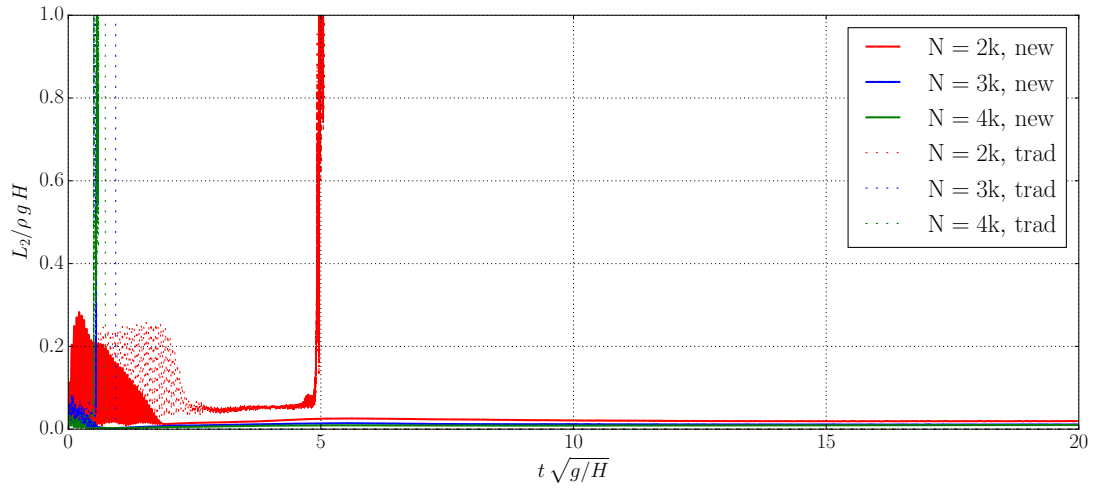


Figure 12: Evolution of the non-dimensional L_2 hydrostatic pressure error for the hydrostatic case. Solid lines: new formulation for three different supports (2, 3 and 4), dashed lines: traditional formulation for the same supports.

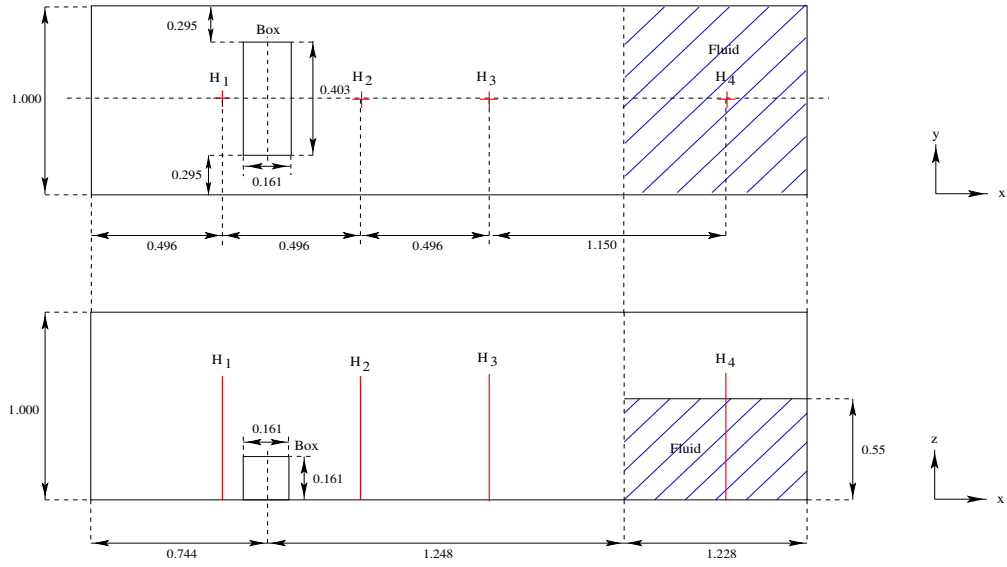


Figure 13: Schematic 3-D dam break flow initial condition [18]

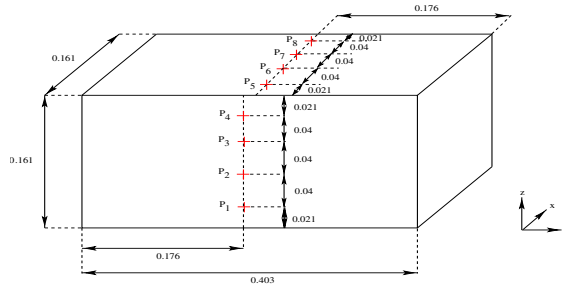


Figure 14: Schematic inner box description [18]

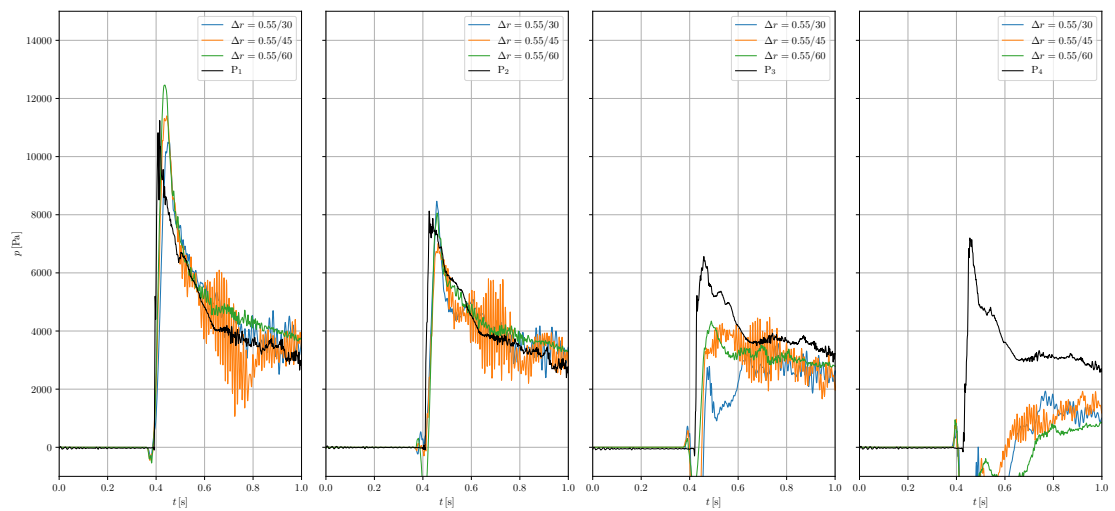


Figure 15: Pressure validation for different initial particle spacing values

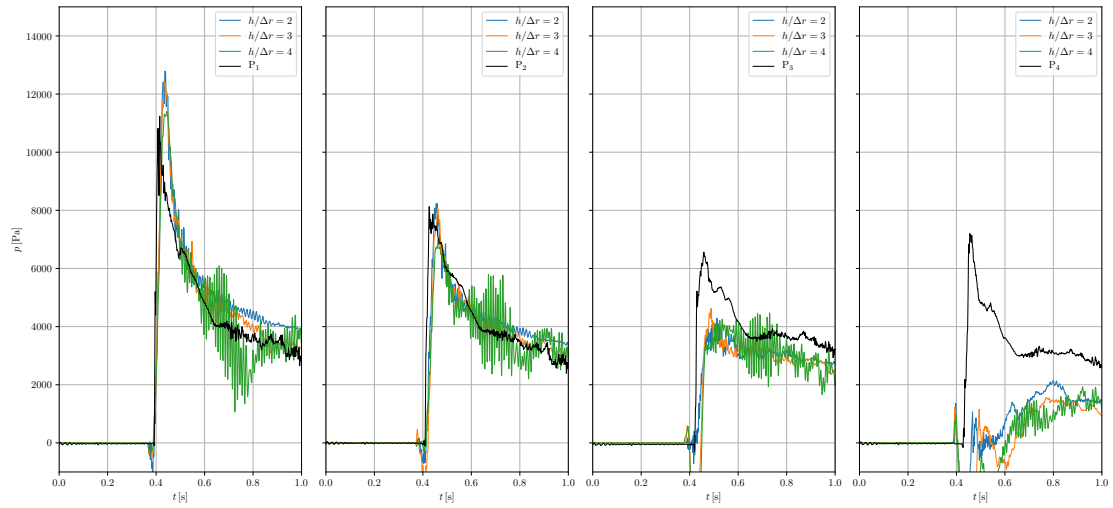


Figure 16: Pressure validation for different kernel length ratios

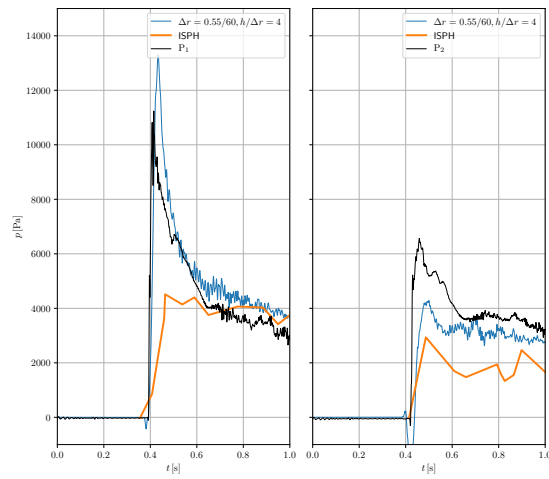


Figure 17: Pressure comparison with an ISPH solution

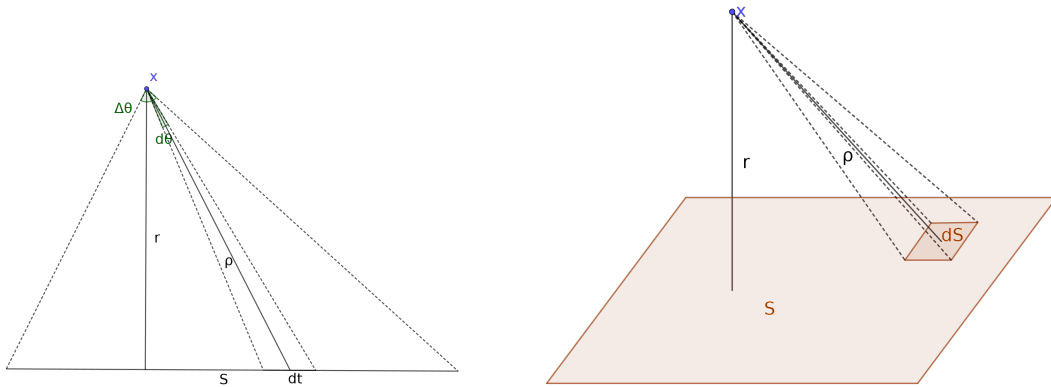


Figure A.18: Sketches of the integration over patches in 2-D (left) and 3-D (right)

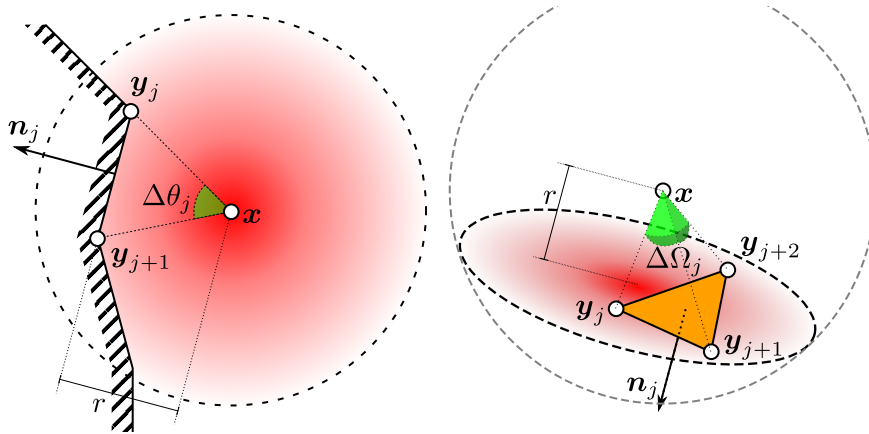


Figure B.19: Schematic view of the subtended angle computation within the semi-analytical context [17].

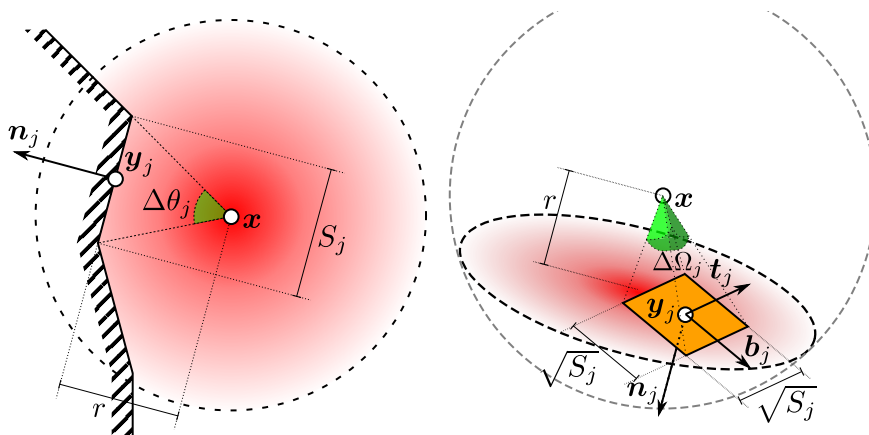


Figure B.20: Schematic view of the subtended angle computation within the purely numerical context [8].