

## VALIDATION OF A NEW FULLY-EXPLICIT INCOMPRESSIBLE SMOOTHED PARTICLE HYDRODYNAMICS METHOD

D. A. Barcarolo<sup>1</sup>, D. le Touzé<sup>1</sup>, F. de Vuyst<sup>2</sup>

<sup>1</sup> LUNAM Université, Ecole Centrale de Nantes, LHEEA Lab. (UMR CNRS), Nantes, France  
([daniel.barcarolo@ec-nantes.fr](mailto:daniel.barcarolo@ec-nantes.fr))

<sup>2</sup> Ecole Normale Supérieure de Cachan, CMLA Lab. (UMR CNRS), Cachan, France

**Abstract.** *The smoothed particle hydrodynamics method has been deeply studied and validated for the past decades on a weakly-compressible fluid discretization for the Euler equations. Recently several authors have proposed the use of SPH truly-incompressible algorithms in the most varied domains of computational fluid dynamics. Within this technique, based on the projection method, a pressure Poisson equation is solved to obtain an incompressible pressure profile. Even though good results have been achieved, the method lacks of CPU performance as an equation has to be solved implicitly. In this article we will present a new fully explicit Incompressible SPH. We will show that nevertheless the explicit resolution of the Euler equations, very good results are achieved both for viscous flows and for free-surface flows. We are also going to assess the use of a more precise free-surface detection algorithm and a more simple and as performing stabilizing technique than the ones existing on the literature.* **Keywords:** *Smoothed Particle Hydrodynamics, Incompressible Flow, Explicit Formulation.*

### 1. INTRODUCTION

Smoothed particle hydrodynamics (SPH) method is a fully Lagrangian, mesh-free method where the fluid domain is discretized by a set of calculation points called particles, where differential operators are evaluated through the use of a kernel function which determines a range of interaction with the neighbor particles. It has firstly been introduced by Lucy [1] and Gingold & Monaghan [2] and has since been applied to a great variety of flow problems. In order to simulate incompressible flow, there are two main solutions available in the literature: (1) the method called Weakly-Compressible SPH (WCSPH) where the fluid is represented by the compressible Navier-Stokes equations and incompressibility is represented as a pseudo-compressibility where the sound speed is artificially set in a way to keep the density variation below a certain level ( $\pm 1\% \rho_0$ ); or (2) the Incompressible SPH (ISPH) method where the incompressibility of the flow is assured at each time step by solving a pressure Poisson equation to an arbitrary accuracy. Variations of the first approach exist like Riemann-

SPH [3] and  $\delta$ -SPH [4]. The ISPH mentioned in this paper is the one based on the projection method, as originally presented by [5], although some other alternatives exist [6].

Since the introduction of the ISPH method by Cummins & Rudman [5], it has been used to model free-surface flows [7], wave breaking problems with the inclusion of a turbulence model [8], viscous flows [9], and multiphase flows [10], [11]. The method can be divided into three main variants: the first, presented in [5], involves a dependence on the velocity divergence in the right hand side (RHS) of the Poisson equation. This variant is referred to as ISPH\_DF in this article (DF standing for Divergence Free). In the second variant, the RHS of the Poisson equation depends on a density prediction as outlined by Shao & Lo [7], referred to as ISPH\_DI (DI standing for Density Invariant). The third variant is a hybridization of ISPH\_DF and ISPH\_DI, solving two Poisson equations (PPE) as described by Hu & Adams [10], and referred to as ISPH\_DFDI (Divergence Free and Density Invariant).

A detailed comparison of these three variants has been performed by Xu et al. [12]. They showed that ISPH lacks of accuracy for certain flow problems (free-surface flow mainly), due to errors associated with the truncated kernels. In order to prevent this loss of accuracy, they have proposed a FVPM-like (Finite Volume Particle Method) shifting algorithm [13] applied to the divergence free variant. This technique showed to perform well both for viscous and free-surface flows. The same author has proposed a Peclet number-based free-surface stabilization technique [14] where the viscosity of the particles close to the free-surface is artificially increased. It has shown good results, mainly for wave propagation phenomena. Recently, another approach has been developed by Lind et al. [15] based on Fick's law of diffusion and good results have been achieved for a variety of flows. In [16] a repulsive component of Lennard-Jones potential in the advection equation is used to prevent particles fracturing and therefore stabilize the method.

Other authors proposed to use some alternative corrections to increase the accuracy of the method, involving a greater effort to get in turn a more precise solution of the pressure Poisson equation. Khayyer et al. [17] proposed to correct the kernel gradient by renormalizing it. Large effort has also been made to improve the MPS (Moving Particle Semi-Implicit) method by Khayyer & Gotohin [18] where they used a higher order Laplacian operator and a renormalization-like operator to improve the PPE precision. The MPS method is very similar to Incompressible SPH, having the same solving procedure (projection method) and it is also a particle-based Lagrangian technique.

Unlike for the WCSPH method where free-surface conditions are intrinsic provided a globally consistent discretization is used, see Colagrossi et al. [19], when ISPH is applied to free-surface flow problems one needs to impose free-surface conditions to be able to solve the PPE. The kinematic condition is still intrinsic thanks to the Lagrangian formalism adopted, but the dynamic condition has to be imposed. It was found by different authors that the more accurately the free surface is detected, the more precise and less noisy the pressure field is. Therefore we propose here to use of a more accurate free-surface detection algorithm developed by Marrone et al. [20] instead of the traditionally used techniques.

As stated before, the ISPH method resides in the solution of the PPE which makes the method computationally intensive. Even if within this method the time step is no longer limited by the speed of sound as in WCSPH, due to the Lagrangian semi-implicit nature of the

resolution the time step is still limited by a CFL condition where the considered velocity is the one of the flow. The increase in time step permitted is thus limited to one order of magnitude only. It has been verified by the authors of the present article that only by getting the time step to its stability limits a CPU time comparable to the one of WCSPH can be achieved. Yet, the main advantage of using an implicit algorithm to solve Navier-Stokes equations is supposed to reside in the possibility to overcome the time steps limitations of an explicit algorithm, leading in Eulerian mesh-based methods (like the standard Finite Volume Method) to time steps which can be, depending on the flow, of several orders of magnitude (100, 1000...) bigger than the ones which have to be used in an explicit solution of the same flow. Thus, in the impossibility of getting lower computational costs, one may say that the advantage of ISPH is in getting a more stable and less noisy pressure field as well on its larger applicability, but as it can be verified on the literature, this method is only able to give the same quality of results as Riemann-SPH and  $\delta$ -SPH and is applied up to now to a narrower spectrum of problems.

Hence, in order to justify better the use of the traditional Incompressible SPH algorithms, one must be able to find a manner to (1), reduce the CPU time, or (2), overcome the CFL limitation, always in a way preserving the good quality of results already achieved by the method. The projection method has been largely used and studied in Finite Element Method ([21],[22], [23] and [24]). Gresho et al. [23] showed that the semi-implicit projection algorithm must respect a convective stability limit where  $CFL \leq O(5-10)$ , which corresponds to 5-10 times the WCSPH time step. Later Christon & Carroll [24] stated that  $CFL \approx 5$  is a reasonable trade-off between accuracy and computational cost for Finite Element Method-based CFD algorithms. In ISPH this condition can also be verified, even though when increasing the CFL number the quality of the results is poorer. In the present article we propose a solution to the first restriction (reduce the CPU time) by making the solution of the PPE explicit. That way, we intend to reduce the CPU time and at the same time achieve results that are of comparable accuracy to the existing SPH algorithms. Also, in the method we are presenting, an existing technique, XSPH [25], will be used to stabilize the scheme when applied to free-surface flows.

Regarding the explicit solution of the PPE itself, it has already been used to simulate viscous flows by Rafiee et al. in [26] where focus was given to fluid structure interaction problems. Nonetheless, the latter cited article lacks of an analysis of the method as well as of a validation process to assess its applicability compared to the existing SPH techniques. Here we argue on the explicit ISPH method properties, we propose dedicated techniques necessary to its application to free-surface flows, and we validate it on a large series of benchmark test cases.

The present article is organized on the following way: first the governing equations are presented; then the SPH method is shortly exposed; right after, a review of the existing ISPH algorithms is made; then the explicit ISPH algorithm is explained together with the new stabilizing technique; then the boundary treatment is presented (including the new free-surface detection algorithm); finally, the results of the proposed method are discussed on four test cases: the lid-driven cavity flow, the flow past a cylinder and two dam-break simulations: one in 2D and another in 3D; conclusions are then drawn.

## 2. GOVERNING EQUATIONS

Within ISPH, we consider the incompressible Navier-Stokes equations in a moving Lagrangian referential:

$$\nabla \cdot u = 0 \quad (1)$$

$$\frac{Du}{Dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 u + F \quad (2)$$

$$\frac{Du}{Dt} = u \quad (3)$$

where  $\rho$ ,  $p$ ,  $u$ ,  $\nu$  and  $F$  stand for density, pressure, velocity, kinematic viscosity and body force respectively. The viscous term may be neglected when these equations are applied to flows which can be considered quasi-inviscid.

## 3. SPH METHODOLOGY

In SPH the fluid domain is discretized by a finite number of particles which represent elementary fluid volumes. Each particle has its own volume ( $V_j$ ) and some other physical properties. In order to evaluate a field  $f_i$  for a  $i$ -th particle one has to use the following convolution sum:

$$f_i = \sum_j f_j W_{ij} V_j \quad (4)$$

where  $f_j$  is the value of  $f$  for the particle  $j$ , and  $W_{ij} = W(r_i - r_j, h)$  a smoothing kernel function with  $h$  being the measure of the kernel support known as smoothing length and  $r_{ij}$  the distance between particles  $i$  and  $j$ . To evaluate the gradient of the field  $f$  one may use (5):

$$\nabla f_i = \sum_j f_j \nabla W_{ij} V_j \quad (5)$$

We may consider the evaluation of the particle density which is given by (6):

$$\rho = \sum_j m_j W_{ij} \quad (6)$$

Now we can discretize the terms on (1) and (2) using the SPH formulation.

### 3.1. Viscous Operator

The viscous operator used in this article is the one given by Morris et al [27] already used on ISPH:

$$\nu \nabla^2 u_i = 2 \sum_j V_j \frac{\nu r_{ij} \cdot \nabla W_{ij}}{(r_{ij}^2)} u_{ij} \quad (7)$$

where  $\nu$  is the kinematic viscosity.

### 3.2. Pressure Gradient

The pressure gradient is computed using the formulation already used for Standard SPH and in semi-implicit ISPH [12].

$$\nabla p_i = \sum_j V_j (p_i + p_j) \nabla W \quad (8)$$

### 3.3. Laplacian Operator

The laplacian operator used to discretize the pressure Poisson equation is the same as the one used in (7) for the viscosity. This operator has already been applied to ISPH by Xu et al [12] and when applied to the pressure terms it reads:

$$\nabla^2 p_i = 2 \sum_j \frac{p_j r_{ij} \cdot \nabla W_{ij} V_j}{|r_{ij}|^2} \quad (9)$$

## 4. REVIEW OF THE TRADITIONAL ISPH METHODS

In order to get incompressible pressure and velocity fields the Navier-Stokes equations are splitted into two parts following the projection method [28]. The first contains the viscous and body forces, and the second contains the pressure force. Between the two parts, the pressure Poisson equation is solved, from which the incompressible pressure field results.

In the first part, considered to be the prediction part, intermediate particles velocities are computed by:

$$u_i^* = u_i^n + (\nu \nabla^2 u_i^n + F_i^n) \Delta t \quad (10)$$

and then intermediate particles positions are evaluated as:

$$r_i^* = r_i^n + u_i^* \Delta t \quad (11)$$

The final velocity field is obtained by applying the pressure gradient term from the momentum equation:

$$u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho_0} \nabla p_i^{n+1} \quad (12)$$

By applying the divergence operator to (12) and knowing that the incompressibility condition (1) must be respected, we get the pressure Poisson equation (PPE) for the divergence free (ISPH\_DF) variant:

$$\nabla \cdot \left( \frac{1}{\rho_0} \nabla p_i^{n+1} \right) = - \frac{1}{\Delta t} \nabla \cdot u_i^* \quad (13)$$

The pressure field obtained from (13) is incompressible, and is used to update particle velocities and positions in (12) and (14), respectively.

$$r_i^{n+1} = r_i^n + \Delta t \left( \frac{u_i^{n+1} + u_i^n}{2} \right) \quad (14)$$

The PPE for the ISPH\_DI variant is obtained by applying the continuity assumption (15) into (14).

$$\nabla \cdot \mathbf{u}_i^* = - \left( \frac{1}{\rho_0} \frac{d\rho_i}{dt} \right) \quad (15)$$

The PPE for the density invariant variant reads:

$$\nabla \cdot \left( \frac{1}{\rho_0} \nabla p_i^{n+1} \right) = \frac{\rho_0 - \rho_i^*}{\rho_0 \Delta t^2} \quad (16)$$

where  $\rho_i^*$  is the intermediate density given by intermediate particle positions  $r_i^*$ . Particle velocities and positions are updated via the same integration as in the ISPH\_DF variant: (12) and (14).

This method, as already stated in the introduction, presents a major downside: its computational cost. Actually, in order to solve the PPE one has to use an iterative solver like BiCGStab or GMRES, but the use of these methods implies higher CPU costs and also scalability issues. This particularity would not be seen as a downside if the time step could be largely increased as it happens in Eulerian mesh-based Finite Volume Methods. However, due to the semi-implicit nature of the ISPH method, the time step cannot be increased of more than  $O(\approx 5)$  times the one used in WCSPH. One could justify the use of the method by the quality of its results, but enhanced WCSPH variations like Riemann-SPH and  $\delta$ -SPH are able to give as smooth and accurate pressure results without a large computational cost increase.

## 5. EXPLICIT ISPH

Here we will present the novel ISPH method by which we search to avoid the computational cost of solving the pressure Poisson equation as well as the scalability issues resulting from a linear system traditional solving procedure. As seen on the previous section, the PPE can be represented by the following form:

$$\sum_j (p_j - p_i) A_{ij} = B_i \quad (19)$$

where  $A_{ij}$  is a matrix variable depending on the formulation used to discretize the laplacian operator,  $B_i$  is RHS vector of the PPE. In a simplified manner we may represent (19) as:

$$Ax = B \quad (20)$$

Equation (20) is generally solved using a BiCGStab or a GMRES algorithm, but, as one may also notice, it can be solved using a Jacobi method. Within this method we assume that we can split the  $A_{ij}$  matrix into two parts: one containing the diagonal terms ( $D$ ) and another containing the terms outside the diagonal ( $R$ ). Then we can iterate until reaching the solution by using the following relation:

$$x^{k+1} = D^{-1}(B - Rx^k) \quad (21)$$

where  $k$  is the iterative step. If we apply (21) into (19) and if we replace  $k$  (iteration) by  $n$  (time step) we get an explicit expression for the PPE:

$$p^{n+1} = \frac{\left( B_i + \sum_j A_{ij} p_j^n \right)}{\sum_j A_{ij}} \quad (22)$$

where  $\sum_j A_{ij}$  is nothing more than  $A_{ii}$  (in other words, the diagonal term  $D$ ) and  $\sum_j A_{ij} p_j^n$  stands for the terms outside the diagonal ( $R$ ) multiplied by the pressure. Needless to say that the operators  $A_{ij}$  and  $B_i$  are evaluated at positions  $r_i^*$ .

Of course, using a simple Jacobi-like solution is known to be less accurate than using an iterative solution, but one must remember that we will use small time steps, resulting in a large number of Jacobi evaluations in the end. Thus, we intend to verify in the present paper that, with this new method, we are able to keep the quality of results given by the semi-implicit ISPH algorithm and decrease the CPU costs required by the latter.

## 5.1. Stabilizing method

It is stated in the literature that ISPH is unstable in the presence of a free-surface, where particles tend to move more freely. In order to correct this instability, several stabilizing methods have been proposed in the literature (see, [12], [14] and [15]), all of them showing to be very performing, specially the FVPM-like technique [12]. Here we decided to use the XSPH method [25]: a correction that acts smoothing the particle velocities and is defined as follows:

$$[\Delta u_i]_{XSPH} = \varepsilon \sum_j \frac{m_j}{\rho_i + \rho_j} (u_j - u_i) W_{ij} \quad (23)$$

where  $\varepsilon$  is usually taken as 0.5, but may vary according to the simulation.

The main advantage of this method is its simplicity compared to the already existing stabilizing techniques. In our experience, it showed to be as efficient as other techniques both when applied in the explicit and semi-implicit ISPH.

## 5.2. Time step

In WCSPH the time step is given by a CFL condition depending on the sound speed ( $C_0$ ) and for viscous flows there is also a viscosity-based CFL condition. In ISPH the time step is computed in a similar way, where the speed of sound is replaced by the maximum velocity of the flow ( $u_{max}$ ).

$$\Delta t_{\max} = CFL \frac{h}{u_{\max}} \quad (24)$$

$$\Delta t_{\max} = CFL \frac{h}{\nu_0} \quad (25)$$

where CFL is a factor chosen by the user varying according the simulation. By using a larger time step the convective stability criteria would not be respected and the incompressible con-

dition would be broken. The best compromise time step duration – accuracy has not been studied in detail yet, and we use rather low CFL numbers for now ( $O(1)$ ). However, this permits still faster incompressible SPH calculations than with a semi-implicit solver (cf. below).

## 6. BOUNDARY CONDITIONS AND FREE-SURFACE DETECTION

Here we describe the procedure applied to get the different types of boundary conditions that are necessary for the test cases: free-slip and no-slip walls, inflow, outflow, and the free-surface treatment. Within the explicit ISPH presented here, the free-surface is automatically taken into account by the SPH operators even for the pressure computation, as in WCSPH [19]. We are thus not theoretically obliged to detect the particles belonging to the free-surface in order to apply the dynamic boundary condition there. However, in the simulation of fast dynamics situations more stable results are obtained when imposing that dynamic boundary condition on the free-surface.

### 6.1. Ghost particles

The ghost-particle method is used in this article to apply the wall (slip or no-slip), inflow and outflow boundary conditions. Within this technique ghost particles are created inside the wall by mirroring fluid particles as proposed by [29] or [30]. This method has been widely used in ISPH to enforce boundary conditions ([5] and [12]). In order to enforce a free-slip boundary condition, velocities and pressures are symmetrised on the ghost particles. Recently De Lefte et al [31] proposed a modification on the no-slip boundary conditions in WCSPH where the no-slip conditions is only applied to the viscous part of the Navier Stokes equations and for the hyperbolic part the free-slip condition is applied. This way they achieved to prevent the particle instability close to the walls. In this article we decided to use the same assumption, i.e., we will use the no-slip condition only for the viscous part of (2) whereas for the PPE we will use the slip condition.

### 6.1. Free-surface detection algorithm

In the semi-implicit ISPH method the free-surface boundary condition must be applied to solve the PPE and to do so the particles belonging to the free-surface must be detected. Traditionally the free-surface is detected by evaluating the particle intermediate densities and if a particle satisfies  $\rho_i^* < 0.99\rho_0$ , it is considered to be at the free-surface. This method has been largely used in ISPH and in MPS (Moving Particle Semi-implicit method). A similar method has been proposed in [9] in which the detection depends on the divergence of particle positions, where a threshold value is used to define if the particle is on the free-surface.

Here we propose the use of the method proposed in [20] which is divided into two steps: first the eigenvalues of the renormalization matrix are computed for each particle and then, based on the higher eigenvalue ( $\lambda$ ), we split the fluid domain into three subsets of par-

ticles:  $E$ ,  $B$  and  $I.E$  represents the particles that belong to jets or drops of water;  $B$  those who are close to the free-surface, and  $I$  those who are in the interior of the fluid.

This first step being performed only the subset of particles  $B$  is checked again in the second part of the algorithm which takes into account the geometrical characteristics of the particle distribution to determine whether a particle is or is not on the free-surface. For more details, refer to [20].

## 7. INTERNAL FLOWS VALIDATIONS

The explicit ISPH method was tested and validated on 2 different benchmarks tests for internal flows: lid-driven cavity flow and 2D flow past a circular cylinder. In this article we consider the Divergence Free variant of the explicit Incompressible SPH method. Comparison is made between the new method and existing SPH algorithms.

### 7.1. Lid-driven cavity flow

This benchmark is a test widely used to validate numerical solvers. It has been used to validate the Finite Volume method, see e.g.[32].Here we confront explicit ISPH results to WCSPH and FVM results. The geometry is a square of side  $L$  (1m) with 3 fixed walls and one moving top-wall with a constant velocity of 1 m/s as shown in Figure 1.

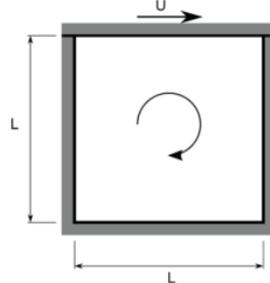


Figure 1. Lid-driven cavity flow sketch.

The kernel used is the cubic spline with an average of 30 neighbor particles for all the simulations. The simulation was performed until a stationary state was achieved, i.e. after 50 seconds in physical time.

For the Reynolds number of  $Re=1000$ , we compare the explicit ISPH\_DF method to the WCSPH [31] method (with a RK4 integration scheme) and to Finite Volume method (FVM) results. The particle distribution considered ( $\Delta x=0.005m$ ) results in 40000 particles. The velocity profiles are compared to Ghia et al. [32] results and the pressure profiles to a FVM package (STARCCM+) as displayed in Figure 2.

Figure 2 shows a very good agreement between the explicit ISPH\_DF, the Finite Volume method, and WCSPH on the velocity profile. From the left-hand side of Figure 2 we can see that the new explicit ISPH method results in a much less noisy pressure field than the one from standard WCSPH. The pressure profile is also quite similar to the one predicted by the Eulerian FVM solver, but for a background pressure. Actually, although in ISPH we no longer have the limitation given by the equation of state we still use the same operators as those

used in WCSPH and, as a consequence, the method suffers as well from tensile instability (TI) when confronted to negative pressures. This explains why there is no negative pressure in Figure 2: whenever there is a region under negative pressure, particles tend to move in a way to create a void in WCSPH, but in explicit ISPH no void was encountered during the simulations, errors resulted rather in an average positive pressure, which can be considered a background pressure. However, when a correction is applied to the operators, renormalization[33] for example, one may encounter high levels of tensile instability. In order avoid this instability one may use a TI control technique. Here we decided not to apply any TI control as its influence on ISPH method has not been verified yet. One quick solution for an internal flow is the use of background pressure in (12), which level has to be chosen to prevent any negative pressure.

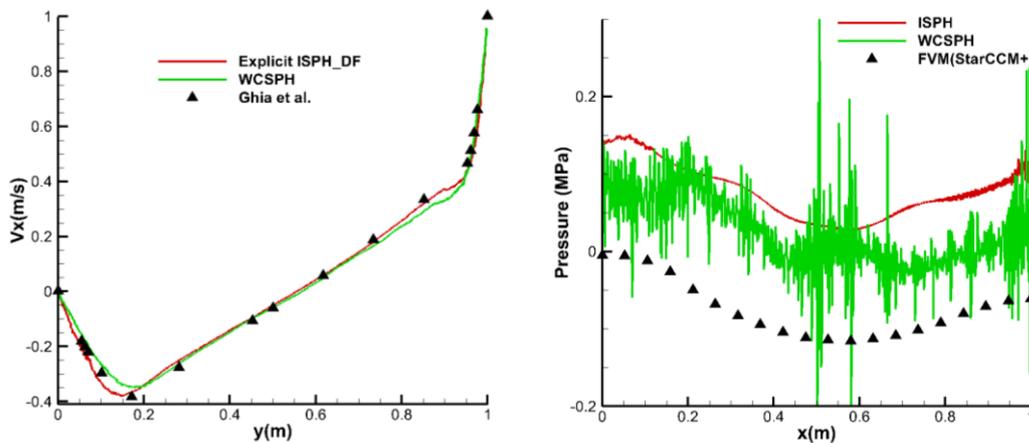


Figure 2. Velocity profile in the x-direction at a constant x section ( $x=0.5$ m) and pressure profile on a constant y section ( $y=0.5$ m). Comparison between FVM [32] (black dots), Explicit ISPH\_DF (red line), and WCSPH (green line) with a particle discretization of  $\Delta x = 0.005$ m.

## 7.2. 2D flow past a circular cylinder

The objective of this test is to validate the approach for non-flat geometries. The configuration of the simulations is shown in Figure 3, where  $D=1.0$ m. Results obtained by explicit ISPH are compared to experimental data [34] and to standard WCSPH.

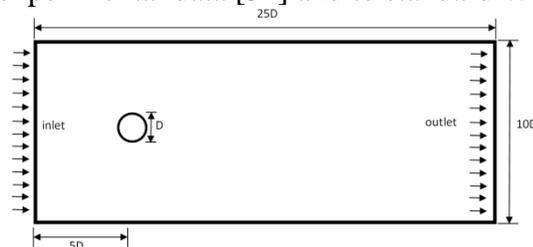


Figure 3. 2D flow past a cylinder simulation sketch.

In Figure 4 one can observe a steady-state comparison between WCSPH [31] and explicit ISPH\_DF for a flow of Reynolds number equal to 40. A very good agreement between the velocity fields predicted by these two methods is visible, although the explicit ISPH\_DF

results are slightly more disturbed on the edge of the wake due to a higher level of self-reorganization of the particles observed with this method.

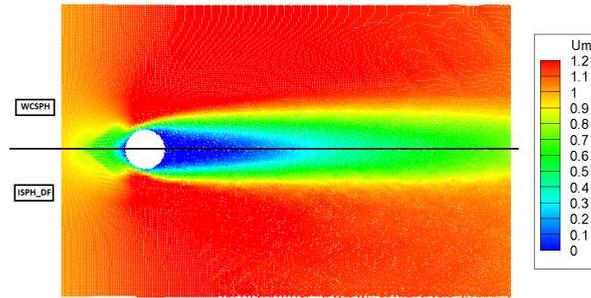


Figure 4. Comparison between WSPH and explicit ISPH\_DF at  $Re = 40$  for  $\Delta x = 0.005m$ .

Comparable pressure coefficient profiles are also found, see the right-hand side of Figure 5. We may also notice at the same figure the effect of particles reorganization on the boundary layer detachment ( $\theta=1.7$  for ISPH and  $\theta=1.9$  for WSPH): a pressure increase is verified at this position.

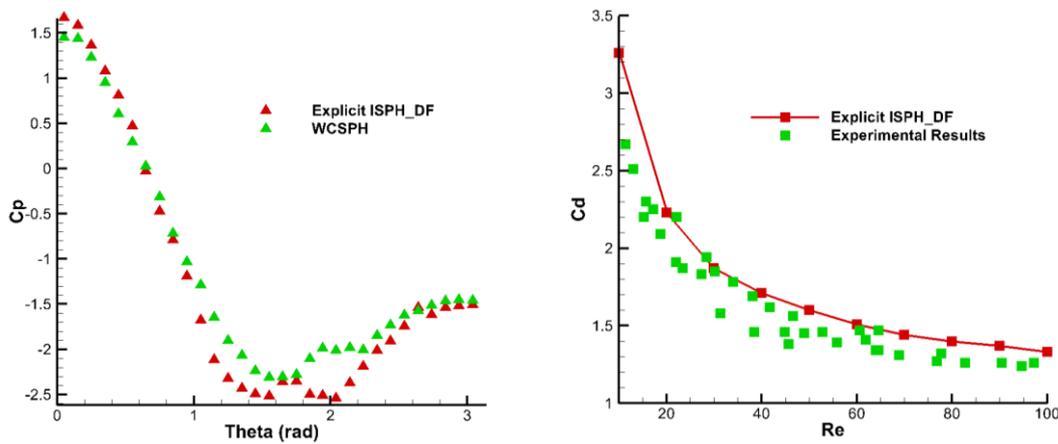


Figure 5. Pressure coefficient around the cylinder at  $Re=40$  and Drag coefficient for several Reynolds numbers compared to experimental results of [34].

Several simulations have been performed at Reynolds numbers ranging in [10-100]. The left-hand side of Figure 5 shows the drag coefficient compared to experimental results by [34]. Explicit ISPH shows the same behavior as in experiments, but predicting slightly higher drags.

## 8. FREE-SURFACE FLOWS VALIDATIONS

### 8.1. 2D dam-break flow against a wall

In this first dam-break test case, the flow impacts a vertical wall. The sketch of the problem is shown in Figure 6 where  $p1$  and  $p2$  indicate two pressure sensors positioned at a height of 0.160m and 0.584m from the bottom, according to the experiment done by [35].

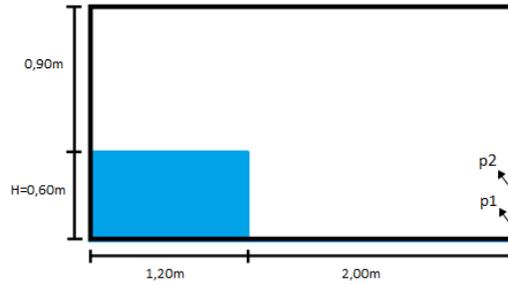


Figure 6. 2D dam-break simulation sketch

Two sets of results are shown: firstly for a discretization of  $\Delta x = 0.01m$  ( $H/\Delta x = 60$ ) then for  $\Delta x = 0.005m$  ( $H/\Delta x = 120$ ), resulting in 7200 and 28800 particles respectively. The results given by the new ISPH method are compared to experimental results [35] and to a Riemann solver based SPH [3] with a Runge-Kutta 4 time integration scheme.

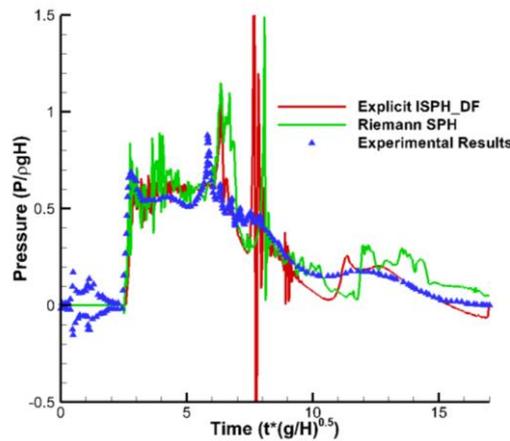


Figure 7. Pressure signal at  $p1$ : comparison between explicit ISPH\_DF, Riemann-SPH and experimental results for  $\Delta x = 0.01m$ .

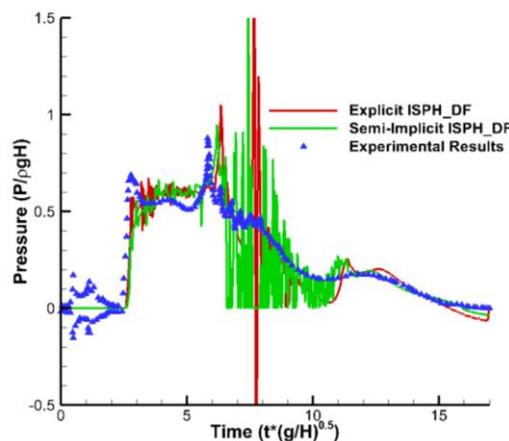


Figure 8. Pressure signal at  $p1$ : comparison between explicit ISPH\_DF, semi-implicit ISPH\_DF and experimental results for  $\Delta x = 0.01m$ .

In Figure 7, we show the  $p1$  pressure profile obtained with both the Riemann-based SPH and explicit ISPH solvers, which is compared to experimental results (using a Gaussian

kernel). The explicit ISPH solution presents a similar behavior to the already validated Riemann-SPH solver with a little less noisy signal. The proposed approach shows also a closer agreement with the experimental signal on average, even if the Riemann-SPH solver captures better the first peak at  $t \cong 1.9$ . Besides, the explicit ISPH solver better captures the pressure peak following the collapse of the void cavity which is entrapped by the plunging jet coming back from the wall (at  $t \cong 6.4$ ).

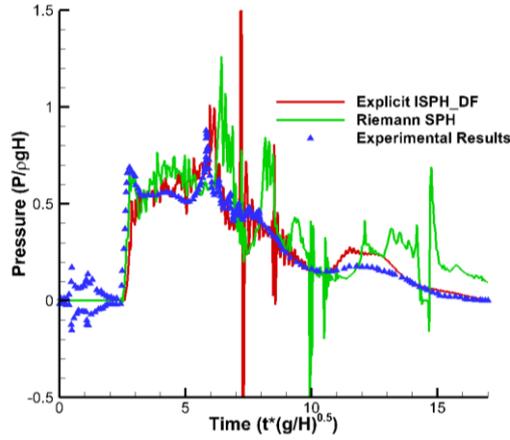


Figure 9. Pressure signal at  $p1$ : comparison between explicit ISPH\_DF, Riemann-SPH and experimental results for  $\Delta x = 0.005\text{m}$ .

On the other hand, in Figure 8, we can see that explicit ISPH and semi-implicit ISPH (both in their divergence-free variant) solvers present similar results although the latter predicts a much more noisy pressure profile even though the same time step is considered here. Actually, in order to better compare the accuracy of the different solvers, the same time step was used for all of them, chosen as the Riemann-SPH one.

When the number of particles is increased one can see in Figure 9 that the new ISPH scheme gets closer to the experience: the bubble collapse peak is well detected and the pressure decays in a very good agreement. Apart from the first peak, the explicit ISPH solution is also much closer to the experiments than the Riemann-SPH solution.

One can note as well that the cost of the semi-implicit variant is 3 times as big as for the explicit variant for the coarser mesh, and this figure increases when using a finer particle resolution. Even though the CFL number might be chosen a bit closer to 1 in the semi-implicit ISPH solver than in the explicit one, efficiency is still higher for the proposed explicit version which is also more accurate (less noisy).

## 8.2. 3D dam-break flow against a rectangular step

We now consider a dam-break with 3D effects where an accurate experience was performed by Kleefsman et al. [36]. This problem involves a rectangular step and its geometry is shown in Figure 10. Here only the results for one particle resolution are presented; we got similar conclusions for other distributions. The discretization with a particle spacing of  $\Delta x = 0.01\text{m}$  results in 676500 particles. Comparison is also made to two different SPH formula-

tions available on the SPH-Flow code developed by Ecole Centrale Nantes and Hydrocean: Riemann-SPH and  $\delta$ -SPH where the last one has already been validated for free-surface flows using this same test in [4]. These last two methods use a Runge-Kutta 4 time integration scheme and all the simulations are performed using a cubic spline kernel and an average of 80 neighbor particles.

In the experiments 8 pressure sensors were placed along the step as seen on the left hand side of Figure 10 at a constant  $y$  section ( $y=0.471\text{m}$ ). In right side of Figure 10 one can note that good agreement is found with the experiments and the other solvers. More specifically, the explicit ISPH solution is very close to the  $\delta$ -SPH one and less noisy than all the other SPH variants.

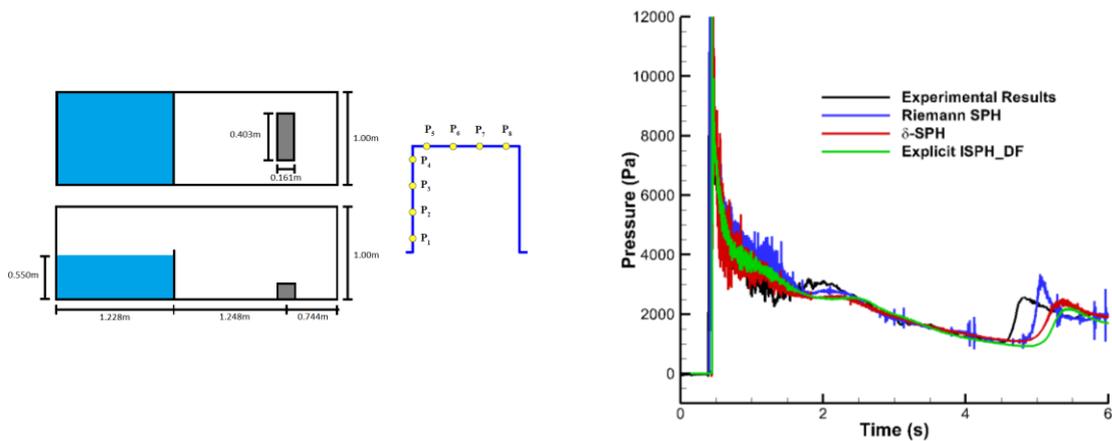


Figure 10. 3D dam-break simulation sketch (left-hand side) and Pressure signal comparison at the pressure sensor P1 (left-hand side).

Table 1 compares the CPU times used by the confronted methods running with the same time step. Explicit ISPH outperforms the other methods in terms of CPU cost while keeping a similar quality in the results. This lower cost is explained by the smaller number of loops over all the particles necessary in the scheme presented in this article in comparison to the usual enhanced WCSPH schemes.

Table 1. Kleefsmans dam-break CPU time comparison

Method	T(s)	% Riemann	% $\delta$ -SPH
Riemann	273712,21	-	-
$\delta$ -SPH	147518,50	-46,10%	-
Explicit ISPH	97020,16	-64,55%	-34,26%

## 9. CONCLUSIONS

A novel incompressible SPH algorithm was presented where the pressure Poisson equation is solved in an explicit Jacobian-like manner. This drastically reduces the computational cost needed to solve this equation which was usually done by using a Bi-CGSTab or a GMRES solver. The proposed explicit ISPH method also relies on two new improvements: the use of a more precise free-surface detection algorithm proposed in [20] and the use of

XSPH to stabilize free-surface flows. A variety of test cases was used to validate the method: viscous flows, with the inclusion of new developments in the ghost-method to treat the no-slip boundary condition as in [31], and free-surface flows, both in 2D and 3D. The algorithm has shown, in some cases, a better accuracy than existing SPH methods (including the semi-implicit ISPH) and in other cases a similar behavior, with always a reduced CPU cost.

We can foresee the use of this method for other types of flows for which the existing SPH methods may lack of performance and to which the semi-implicit ISPH method has already been confronted: wave propagation, multi-fluid flows and fluid structure interaction. Future work will include a detailed study of the compromise time step duration-accuracy, especially to check how close one can get to the convection CFL limit.

## Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/ 2007-2013) under Grant Agreement No. 225967 "NextMuSE". This work has also been supported by the Pays de la Loire French region through a PhD scholarship.

## 10. REFERENCES

- [1] Lucy, L B. Numerical approach to the testing of the fission hypothesis. *Astron. J.* 1977, Vol. 82, pp. 1013-1024.
- [2] Gingold, R A and Monaghan, J J. Smoothed particle hydrodynamics -theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* 1977, Vol. 181, pp. 375-389.
- [3] Monaghan, J J. SPH and Riemann Solvers. *Journal of Computational Physics.* 1997, Vol. 136, pp. 298-307.
- [4] Marrone, S, Antuono, M, Colagrossi, A, Colicchio, G, Le Touzé, D and Graziani G.. Delta-SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering.* 2011, Vol. 200, pp. 1526-1542.
- [5] Cummins, S J and Rudman, M. An SPH projection method. *Journal of Computational Physics.* 1999, Vol. 152, pp. 584-607.
- [6] Ellero, M, Serrano, M and Español, P.. Incompressible smoothed particle hydrodynamics. *Journal of Computational Physics.* 2007, Vol. 226, pp. 1731-1752.
- [7] Shao, S and Lo, E Y. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources.* 2003, Vol. 26, pp. 787-800.
- [8] Shao, S. Incompressible SPH simulation of wave breaking and over topping with turbulence modelling. *International Journal for Numerical Methods in Fluids.* 2006, Vol. 50, pp. 597-621.
- [9] Lee, E-S., Moulinec, C, Xu, R, Violeau, D, Laurence, D and Stansby P K.. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics.* 2008, Vol. 227, pp. 8417-8436.

- [10]Hu, X Y and Adams, N A. An incompressible multi-phase SPH method. *Journal of Computational Physics*. 2007, Vol. 227, pp. 264-278.
- [11]Hu, X Y and Adams, N A. A constant-density approach for incompressible multi-phase SPH. *Journal of Computational Physics*. 2009, Vol. 228, pp. 2082-2091.
- [12]Xu, R, Stansby, P K and Laurence, D. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics*. 2009, Vol. 228, pp. 6703-6725.
- [13]Nestor, R, Basa, M and Quinlan N.. Moving boundary problems in the finite volume particle method. *Proceedings of the 3rd International SPHERIC workshop*. 2008, pp. 109-114.
- [14]Xu, R and Stansby, P K. The influence of the truncated kernel to free-surface predictions with ISPH and a new solution. *Proceedings of the 5th International SPHERIC workshop*. 2010, pp. 130-137.
- [15]Lind, S J, Xu, R, Stansby P K, Rogers B D.. A Stabilising Diffusion-Based Shifting Algorithm for Incompressible Smoothed Particle Hydrodynamics. *Proceedings of the 6th International SPHERIC workshop*. 2011, pp. 14-20.
- [16]Shadloo, M S, Zainali, A, Sadek S H and Yildiz, M.. Improved Incompressible Smoothed Particle Hydrodynamics method for simulating flow around bluff bodies. *Computer Methods in Applied Mechanics and Engineering*. 2011, Vol. 200, pp. 1008-1020.
- [17]Khayyer, A, Gotoh, H and Shao, S D. Corrected Incompressible SPH method for accurate water-surface tracking in breaking waves. *Coastal Engineering*. 2008, Vol. 55, pp. 236-250.
- [18]Khayyer, A and Gotoh, H. A higher order Laplacian model for enhancement and stabilization of pressure calculation by the MPS method. *Applied Ocean Research*. 2010, Vol. 32, pp. 124-131.
- [19]Colagrossi A., Antuono M., Le Touzé D. Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model. *Phys. Rev. E*.. 2009.
- [20]Marrone, S, Colagrossi, A, Le Touzé, D and Graziani G.. Fast free-surface detection and level-set function definition in SPH solvers. *Journal of Computational Physics*. 2010, Vol. 229, pp. 3652-3663.
- [21]Gresho, P M. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *International Journal for Numerical Methods in Fluids*. 1990, Vol. 11, pp. 587-620.
- [22]Gresho, P M and Chan, S T. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation. *International Journal for Numerical Methods in Fluids*. 1990, Vol. 11, pp. 621-659.
- [23]Gresho, P M, Chan, S T, Lee, R L and Upson, C D.. A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 1: Theory. *International Journal for Numerical Methods in Fluids*. 1984, Vol. 4, pp. 557-598.
- [24]Christon, M. A., Carroll, D. E.. Unstructured-grid, parallel, projection solver for computing low-speed flows. *Technical Report*. 1998.

- [25] Monaghan, J J. On the problem of penetration in particle methods. *Journal of Computational Physics*. 1989, Vol. 82, pp. 1-15.
- [26] Rafiee, A and Thiagarajan, K P. An SPH projection method for simulating fluid-hypoelastic structure interaction. *Computer Methods in Applied Mechanics and Engineering*. 2009, Vol. 198, pp. 2785-2795.
- [27] Morris, J P, Fox, P J and Zhu, Y. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*. 1997, Vol. 136, pp. 214-226.
- [28] Chorin, A J. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*. 1968, Vol. 22, pp. 745-762.
- [29] Libersky, L D, Petschek, A G, Carney, T C, Hipp, J R and Allahdadi, F A.. High Strain Lagrangian Hydrodynamics: A Three-Dimensional SPH Code for Dynamic Material Response. *Journal of Computational Physics*. 1993, Vol. 109, pp. 67-75.
- [30] Colagrossi, A and Landrini, M. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*. 2003, Vol. 191, pp. 448-475.
- [31] De Leffe, M., Le Touzé, D. and Alessandrini, B. A modified no-slip condition in weakly-compressible SPH. *Proceedings of the 6th International SPHERIC workshop*. 2011, pp. 291-297.
- [32] Ghia, U, Ghia, N and Shin, C T. High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. *Journal of Computational Physics*. 1982, Vol. 48, pp. 387-411.
- [33] Randle, P W and Libersky, L D. Smoothed particle Hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*. 1996, Vol. 139, pp. 375-408.
- [34] Tritton, D J. Experiments on the flow past a circular cylinder at low Reynolds. *Journal of Fluids Mechanics*. 1959, Vol. 6, p. 547.
- [35] Buchner, B. *Green Water on Ship-type Offshore Structures*. s.l. : PhD-thesis Delft University of Technology, 2002.
- [36] Kleefsman, K M T, Fekken, G, Veldman, A E P, Iwanowski, B and Buchner, B.. A Volume-of-Fluid based simulation method for wave impact problems. *Journal of Computational Physics*. 2005, Vol. 206, pp. 363-393.