

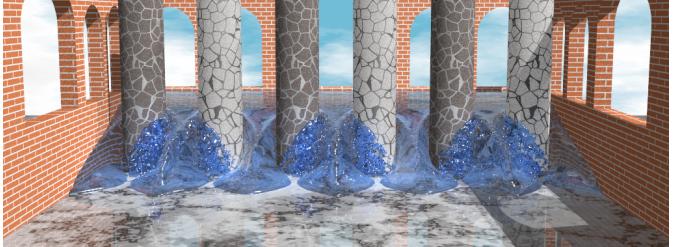
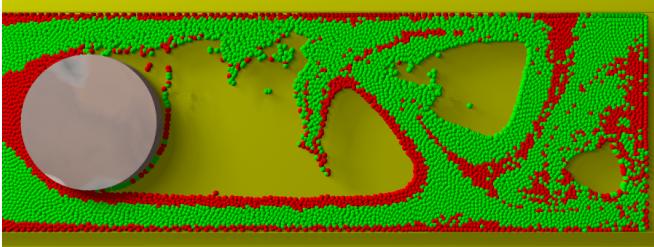
# Staggered Meshless Solid-Fluid Coupling

Xiaowei He \*†‡, Ning Liu‡, Guoping Wang‡, Fengjun Zhang†, Sheng Li‡, Songdong Shao§, and Hongan Wang†

†Institute of Software, Chinese Academy of Sciences

‡Peking University

§University of Bradford



**Figure 1:** Left: fluid flowing around a cylinder with no-slip boundary condition in two dimensions. Right: dam break of fluid flowing around rotating cylinders with no-slip boundary condition in three dimensions.

## Abstract

Simulating solid-fluid coupling with the classical meshless methods is an difficult issue due to the lack of the Kronecker delta property of the shape functions when enforcing the essential boundary conditions. In this work, we present a novel staggered meshless method to overcome this problem. We create a set of staggered particles from the original particles in each time step by mapping the mass and momentum onto these staggered particles, aiming to stagger the velocity field from the pressure field. Based on this arrangement, an new approximate projection method is proposed to enforce divergence-free on the fluid velocity with compatible boundary conditions. In the simulations, the method handles the fluid and solid in a unified meshless manner and generalizes the formulations for computing the viscous and pressure forces. To enhance the robustness of the algorithm, we further propose a new framework to handle the degeneration case in the solid-fluid coupling, which guarantees stability of the simulation. The proposed method offers the benefit that various slip boundary conditions can be easily implemented. Besides, explicit collision handling for the fluid and solid is avoided. The method is easy to implement and can be extended from the standard SPH algorithm in a straightforward manner. The paper also illustrates both one-way and two-way couplings of the fluids and rigid bodies using several test cases in two and three dimensions.

**CR Categories:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

**Keywords:** staggered SPH, physically-based animation, solid-fluid coupling

**Links:**

\*clouddon.he@gmail.com

### ACM Reference Format

He, X., Liu, N., Wang, G., Zhang, F., Li, S., Shao, S., Wang, H. 2012. Staggered Meshless Solid-Fluid Coupling. *ACM Trans. Graph.* 31, 6, Article 149 (November 2012), 12 pages. DOI = 10.1145/2366145.2366168  
<http://doi.acm.org/10.1145/2366145.2366168>

### Copyright Notice

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, fax +1 (212) 869-0481, or permissions@acm.org.

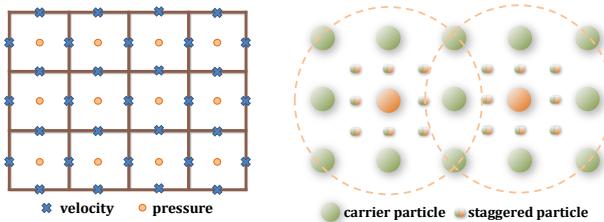
© 2012 ACM 0730-0301/12/11-ART149 \$15.00 DOI 10.1145/2366145.2366168  
<http://doi.acm.org/10.1145/2366145.2366168>

## 1 Introduction

The solid-fluid interactions are common in everyday life, such as when pouring beer into a glass, dropping a stone into the water, etc. In the documented meshless methods, much attention has been paid to the fluid animations, such as the works [Adams et al. 2007; Solenthaler and Pajarola 2009; Solenthaler and Gross 2011] based on the Lagrangian Smoothed Particle Hydrodynamics (SPH) method. However, under this purely Lagrangian framework, only few works have dealt with the coupling problem since there are still several problems in the SPH which have not been fully addressed [Liu and Liu 2010]. For example, due to the lack of the Kronecker delta property of the shape functions, the free-slip and no-slip boundary conditions cannot be imposed as easily as in a mesh-based method. The spurious zero-energy modes are also well known for the null-space issues, which is caused by the fact that the field variables and their derivatives are calculated at the same positions. As a result, high-frequency oscillations in the variable field may persist or even grow up that will lead to the simulation failure.

Referring to the grid-based methods, a novel “MAC-grid” as shown in Figure 2 (left) [Harlow et al. 1965] is commonly used to overcome the problem of the zero-energy modes. Inspired by this brilliant work, [Vignjevic et al. 2000; Randles and Libersky 2000] extended the SPH to a stress-point formulation which was composed of the stress particles and velocity particles. With this arrangement, the zero-energy modes can be trivially avoided as the collocated nature of the standard SPH is removed by adding the stress particles. The boundary treatment is also simple, since it is possible to use either type of the particles to enforce the boundary conditions. However, this method has several disadvantages that make it inappropriate to apply directly to the solid-fluid interactions. One major problem comes from the irregular distributions of the particles in the simulation. Both compression and expansion of the particles can severely degrade the solution, resulting in the simulation failure. Besides, the interleaved pattern of the stress particles and velocity particles cannot be well preserved if complex fluid interactions with the solid occur. In addition, the problems, such as extra computing cost of finding the neighbors for stress particles and the numerical dissipations, also limit the application of this method.

In this paper, a new arrangement of different particles is devised as shown in Figure 2 (right). To ensure the stability, we enforce the density constraint on the carrier particles, which represent the discretization of the problem domain, and create the staggered particles at each time step at the positions between each pair of the



**Figure 2:** Left: field variable value placement in MAC grid, velocities are defined at cell boundaries while pressures are defined at cell centers. Right: particle placement of our method where the staggered particles are created from carrier particles at each time step to avoid simulation failure. Here only the staggered particles related to the orange ones are shown.

neighboring carrier particles. Based on the different types of particles, an approximate projection method will be further developed to enforce the divergence-free on the fluids with the required boundary conditions, which is the key to model various solid/fluid interactions. To impose the free surface boundary condition, a robust method to detect the surface particles is also given. Then, the free surface boundary and various solid wall boundary conditions can be exerted on the carrier particles and staggered particles, respectively. In our unified meshless handling of the fluids and rigid bodies, we extended the formulations of the viscous and pressure forces given by [Müller et al. 2003] to handle the discontinuities at the solid/fluid boundary. As a result, no explicit collision detection algorithm is required to prevent the fluid particles from penetrating into the solid due to the use of the generalized pressure force, and various slip boundary conditions can be implemented with the assistance of the staggered particles and the generalized viscous force.

## 2 Related Work

There are a variety of computational methods to simulate the fluids and their interactions with the solids. A dominant approach referred as the “voxelized pressure solve” can be traced back to [Foster and Metaxas 1996] who imposed boundary conditions onto the grid by voxelizing the solid obstacles. [Stam 1999] followed this treatment and proposed an unconditionally stable model for producing the complex fluid-like flows. However, this method generalized significant stair-step artifacts if the boundaries were not aligned with the grids. Later, [Foster and Fedkiw 2001] presented a general method for modeling and animating the liquids, and attempted to mitigate the artifacts by only enforcing the boundary conditions in normal directions, leaving the tangential part of the velocities unchanged. [Enright et al. 2002b] also used this improvement. As all these works only dealt with the one-way coupling, [Takahashi et al. 2002] introduced a simple two-way coupling between the fluids and rigid bodies based on an explicit two-level collision detection algorithm. [Houston et al. 2003] proposed an extension to the previous works, intending to unify the treatment of static and dynamic objects with the level set method. Unfortunately, all these methods still suffer from the previously mentioned artifacts. More recently, [Batty et al. 2007] observed the equivalence between minimizing the total kinetic energy and solving the fluid incompressibility with compatible solid boundary conditions for the rigid-fluid coupling. They have introduced a new variational framework to achieve a fully implicit two-way coupling which was free of the grid artifacts even on the coarse grid.

Another group of researchers typically use an Euler formulation for the fluid and a Lagrangian formulation for the solid [Yngve et al.

2000; Génevaux et al. 2003; Carlson et al. 2004]. [Guendelman et al. 2005] presented a novel approach to treat an infinitesimally thin solid modeled by using lower dimensionally triangulated surfaces. They used a robust ray casting method to avoid the fluid leaking through the triangulated surface. [Robinson-Mosher et al. 2008] pointed out that the impulse distribution necessary to maintain the contact varies with the material densities for the dual cell in voxelized methods such as [Batty et al. 2007]. The nonphysical pressure profile may result in an incorrect torque. Instead of explicitly enforcing the velocity compatibility on the dual cell centers, they derived an implicit solid-fluid coupling method based on the law of conservation of momentum. [Robinson-Mosher et al. 2009] further improved this work to obtain more accurate tangential velocities for the solid-fluid coupling, which allows for the freely flowing tangential velocities.

Alternatively, one can use fully Lagrangian meshless methods for both the fluid and solid [Müller et al. 2004a; Keiser et al. 2005; Solenthaler et al. 2007; Lenaerts et al. 2008]. However, imposing boundary conditions in the meshless methods is not as straightforward as that in the mesh-based methods and thus this has been traditionally regarded as one weakness of the meshless methods [Nguyen et al. 2008; Liu and Liu 2010]. In most SPH simulations, the solid boundary conditions are enforced by using either penalty forces [Monaghan 1994; Monaghan 2005; Solenthaler et al. 2007] or the virtual particles [Morris et al. 1997; Yildiz et al. 2009]. Only a few approaches have tried to model different kinds of boundary conditions and taken into account the actual forces exerted on the solid to achieve a two-way coupling. [Müller et al. 2004b] presented a method to model the interactions of fluids with a deformable solid based on the Lennard-Jones forces between the boundary particles and their neighboring fluid particles. [Oger et al. 2006] extracted local pressures near the solid boundary to achieve the two-way coupling of the particle-based fluids and a moving solid object in two dimensions. [Becker et al. 2009] considered a wide range of slip and Neumann boundary conditions by employing a direct forcing method. Thus both one-way and two-way solid-fluid coupling with the arbitrarily shaped boundaries can be effectively accomplished. However, since all these methods only take into account the local interactions of particles near the solid boundaries, the resulting velocity discontinuity may lead to the simulation failure. In contrast to this, the method proposed in this paper aims to solve the pressure field in the whole fluid domain to enforce both the fluid incompressibility and the solid boundary conditions.

## 3 Overview

The governing equations for a viscous fluid are written in a Lagrangian form as

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{v} = 0 \quad (1)$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f} \quad (2)$$

The two equations represent the conservation of mass and momentum, respectively. The vector field  $\mathbf{v}$  is the fluid velocity,  $\mathbf{f}$  is the external force. The scalar field  $\rho$  is the fluid density,  $p$  is the pressure, and  $\mu$  is the kinematic viscosity. We have used the shorthand notation  $\nabla^2 = \nabla \cdot \nabla$  where the symbol  $\nabla$  is the vector of spatial partial derivatives with the form  $(\partial/\partial x, \partial/\partial y)$  in two dimensions and  $(\partial/\partial x, \partial/\partial y, \partial/\partial z)$  in three dimensions.

For an ideal incompressible fluid, the density of each particle needs to be maintained constant during the whole simulation. More precisely, that is  $D\rho/Dt = 0$  referred to as the **density constraint** [Bodin et al. 2011]. The other constraint referred to as the **velocity constraint** follows from the continuity equation 1 as  $\nabla \cdot \mathbf{v} = 0$ .

In the grid-based methods, due to the pre-assumption of constant density, the researchers usually ignore the density constraint [Stam 1999; Batty et al. 2007]. However, both the density constraint and velocity constraint need to be considered in the meshless methods [Liu et al. 2005]. Therefore, we divide the pressure  $p$  into two parts: one is the density constraint pressure  $p^d$ , which enforces the density of every particle to be unchanged, and the other is the velocity constraint pressure  $p^v$ , which enforces the fluid velocity field to be divergence-free with compatible boundary conditions.

According to the above discussion, we improve the prediction-correction scheme proposed by [Shao and Lo 2003] to solve the incompressible fluid. At the prediction stage, along with the viscous and external forces, the pressure force contributed by  $p^d$  is also included in the Navier-Stokes equation to get an intermediate temporal velocity

$$\tilde{\mathbf{v}} = \mathbf{v} + \Delta t \left( -\frac{1}{\rho} \nabla p^d + \mu \nabla^2 \mathbf{v} + \mathbf{f} \right) \quad (3)$$

After integrating the velocity field forward for the position with  $\mathbf{x}+ = \Delta t \tilde{\mathbf{v}}$ , we can get a density constraint enforced fluid with  $\rho$  being constant for each particle.

For now, the velocity constraint is not satisfied, and a correction step is required to solve  $p^v$ . The new velocity after correction by  $p^v$  can be written as

$$\mathbf{v}^{new} = \tilde{\mathbf{v}} - \frac{\Delta t}{\rho} \nabla p^v \quad (4)$$

where  $\mathbf{v}^{new}$  satisfies the velocity constraint  $\nabla \cdot \mathbf{v}^{new} = 0$ . Substituting equation 4 into the velocity constraint, we can get the following pressure Poisson equation

$$\nabla \cdot \left( \frac{1}{\rho} \nabla p^v \right) = \frac{\nabla \cdot \tilde{\mathbf{v}}}{\Delta t} \quad (5)$$

with which we can solve  $p^v$ . We then substitute  $p^v$  back into equation 4 to get the new velocity. More details on how to solve both  $p^d$  and  $p^v$  in the solid-fluid coupling will be illustrated in the following sections.

## 4 Staggered SPH

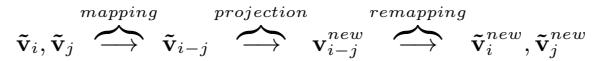
Our incompressible fluid solver is based on SPH (see [Liu and Liu 2010] for a good overview). In this purely meshless method, the problem domain is represented by a set of particles. Each particle  $P_i$  is defined by its position  $\mathbf{x}_i$ , its mass  $m_i$ , and the smoothing radius  $h$ . A physical quantity  $\langle q \rangle$  of particle  $P_i$  can be computed by summing up the contributions of all its neighboring particles  $P_j$

$$\langle q_i \rangle = \sum_j \frac{m_j}{\rho_j} q_j W(|\mathbf{x}_i - \mathbf{x}_j|, h) \quad (6)$$

where  $V_j$  is the volume of particle  $P_j$  and  $W$  is the smoothing function with compact support domain  $\Omega_i$ .

Before proceeding with our staggered meshless method, it will be convenient to introduce some additional notations. In general, we call the originally discretized particles (for both fluid and solid) as **carrier particle** (also particle for short) and the newly created particles as **staggered particle** to reflect the feature of where they are created. A subscript with form “ $i-j$ ” is used to mark the staggered particle  $P_{i-j}$  between the particle  $P_i$  and its neighboring particle  $P_j$ . The relative quantities of  $P_{i-j}$  are thus written as  $\mathbf{x}_{i-j}$  for the position,  $m_{i-j}$  for the mass and  $\mathbf{v}_{i-j}$  for the velocity. Superscripts “ $f$ ” and “ $s$ ” represent the fluid and the solid, respectively.

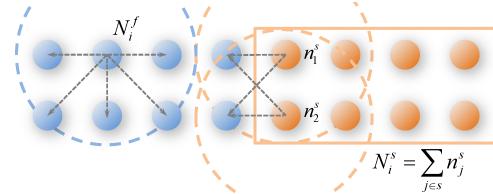
We do not distinguish between the fluid and solid particles if no superscripts are given. Based on above convention, an overview of enforcing the velocity constraint on the fluid is described as follows



After the prediction stage, a set of staggered particles with the velocity  $\tilde{\mathbf{v}}_{i-j}$  are first created by mapping the carrier particles' mass and momentum onto the staggered particles (Section 4.1). Then, in order to enforce the divergence-free on  $\tilde{\mathbf{v}}_{i-j}$ , an new approximate projection method is proposed to solve the velocity constraint pressure  $p_i^v$  defined on particle  $P_i$  (Section 4.2). Finally, the staggered particle velocity  $\mathbf{v}_{i-j}^{new}$  is remapped to get the final velocity  $\tilde{\mathbf{v}}_i^{new}$  of particle  $P_i$  (Section 4.3).

### 4.1 Staggered Particles

To create the staggered particle  $P_{i-j}$ , we take into account the two neighboring particles  $P_i$  and  $P_j$ . Three major physical quantities of  $P_{i-j}$  need to be identified for further use, including the mass  $m_{i-j}$ , the velocity  $\tilde{\mathbf{v}}_{i-j}$  and the position  $\mathbf{x}_{i-j}$ . For simplicity, we suppose that  $\mathbf{x}_{i-j}$  is located at the middle point on the line segment which connects particle  $P_i$  and  $P_j$ , and the other quantities are only contributed by these two particles.



**Figure 3:** Illustration for determining the neighbors of both the fluid (blue) and solid (orange) particles.

Since the newly created staggered particles can be viewed as another discretization of the problem domain, both the mass and momentum conservation should be guaranteed during the mapping. To transfer the mass onto the staggered particles, we use a further hypothesis that each particle  $P_i$  contributes its mass  $m_i$  equally to its  $N_i$  neighboring staggered particles. For the fluid particle  $P_i^f$ ,  $m_i^f$  corresponds to the initialized particle mass and  $N_i^f$  is the number of its neighbors. However, if  $P_i^s$  belongs to a rigid body, care should be taken in assigning values to  $m_i^s$  and  $N_i^s$ . As we know, for ideal rigid bodies, impact on one position of a rigid body will be instantly transmitted to the other part of the object. Thus, it is reasonable for us to assign the total mass  $m^s$  of a rigid body to its subordinate particle, that is  $m_i^s = m^s$ . In computing  $N_i^s$ , we need first find neighbors for each individual particle  $P_i^s$ . Then, we only sum up neighbors that belong to the fluid and assign the value to  $N_i^s$ . In other words,  $N_i^s$  represents the fluid neighbors of a rigid body if we view the rigid body as a particle as well. Based on the above discussion, mass of the staggered particle  $P_{i-j}$  can now be written as the following unified form

$$m_{i-j} = \frac{m_i}{N_i} + \frac{m_j}{N_j} \quad (7)$$

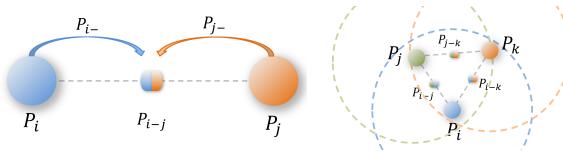
which also be easily verified that the total mass of staggered particles equals to the original system. If we denote the part contributed by  $P_i$  as  $P_{i-}$  and the other part as  $P_{j-}$ , by lumping  $P_{i-}$  and  $P_{j-}$  together as shown in Figure 4 (left), the velocity of  $P_{i-j}$  can now be written as

$$\tilde{\mathbf{v}}_{i-j} = \frac{m_{i-}\tilde{\mathbf{v}}_i + m_{j-}\tilde{\mathbf{v}}_j}{m_{i-j}} \quad (8)$$

according to the momentum conservation law. Here  $m_{i-}$  and  $m_{j-}$  indicate the mass of  $P_{i-}$  and  $P_{j-}$ , respectively. The mixed volume  $V_{i-j}$  and density  $\rho_{i-j}$  are derived from the incompressibility assumption and written as

$$V_{i-j} = \frac{m_{i-}}{\rho_i} + \frac{m_{j-}}{\rho_j} \quad \text{and} \quad \rho_{i-j} = \frac{m_{i-j}}{V_{i-j}} \quad (9)$$

Likewise, we will use  $V_{i-}$  and  $V_{j-}$  to denote the volume of  $P_{i-}$  and  $P_{j-}$  in the following context.



**Figure 4:** Left: the staggered particle created between the two neighboring particles  $P_i$  and  $P_j$ . Right: the redundant staggered particle  $P_{j-k}$  existed inside the support domain  $\Omega_i$  of a particle  $P_i$ .

If we take a look at the staggered particles' distribution inside the support domain  $\Omega_i$  of a particle  $P_i$ , a lot of redundant staggered particles, such as  $P_{j-k}$  shown in Figure 4 (right), can also be found in  $\Omega_i$ . However, in the discretization for  $\langle \nabla \cdot \tilde{\mathbf{v}}_i \rangle$ , only the staggered particles between  $P_i$  and its neighboring particles are considered. The divergence of velocity defined on particle  $P_i$  can thus be discretized as

$$\langle \nabla \cdot \tilde{\mathbf{v}}_i \rangle = - \sum_j \frac{m_{i-j}}{\rho_{i-j}} \tilde{\mathbf{v}}_{i-j} \cdot \nabla W(r_{ij}/2, h) \quad (10)$$

by considering  $|\mathbf{x}_i - \mathbf{x}_{i-j}| = r_{ij}/2$ . Here we denote  $r_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ .

## 4.2 Approximate Projection

The projection method is based on the Helmholtz decomposition [Petronetto et al. 2010], which states that any vector field can be resolved into the sum of an irrotational (curl-free) vector field and a solenoidal (divergence-free) vector field. A variety of projection algorithms exist both in the grid-based and meshless methods, which either use an exact projection or an approximate projection. [Cummins and Rudman 1999] first introduced an approximate projection method for SPH. This method can effectively prevent the problem of pressure decoupling which arises when an exact projection is used. Later, [Shao and Lo 2003] improved on the approximate Laplacian operator to get a symmetric form. However, the authors didn't give an intuitive explanation for why it worked with the approximate Laplacian operator. In addition, the difficulty of exerting solid wall boundary conditions remains unresolved.

Based on the staggered particles, we can derive a more intuitive approximate Laplacian operator with the additional advantage that solid wall boundary conditions can be easily applied. During the derivation, we follow the convention given by [Cummins and Rudman 1999] to define the Laplacian operator as  $L = D\sigma\mathbf{G}$ , where  $D$  is the divergence operator,  $\mathbf{G}$  is the gradient operator and  $\sigma = 1/\rho$ . In the discretization of  $L$ , the gradient operator  $\mathbf{G}$  is defined on the staggered particles. Since  $P_{i-j}$  is only related to particle  $P_i$  and  $P_j$ , we simply use a finite difference method to define  $\mathbf{G}$  as

$$\mathbf{G}p_{i-j}^v = \frac{p_j^v - p_i^v}{r_{ij}} \mathbf{n}_{ij}^v \quad (11)$$

where  $\mathbf{G}p_{i-j}^v$  represents the pressure gradient at  $\mathbf{x}_{i-j}$  and  $\mathbf{n}_{ij}^v$  indicates the direction of the velocity constraint pressure force which

will be detailed in section 5.2. If both particle  $P_i$  and  $P_j$  belong to the fluid,  $\mathbf{n}_{ij}^v$  can be written as  $(\mathbf{x}_j - \mathbf{x}_i)/|\mathbf{x}_j - \mathbf{x}_i|$ . Otherwise, it depends on the imposed solid wall boundary conditions. With the same technique for discretizing  $\langle \nabla \cdot \tilde{\mathbf{v}}_i \rangle$ , we can define the approximate Laplacian  $Lp_i^v$  on particle  $P_i$  as follows

$$\begin{aligned} Lp_i^v &= \langle \nabla \cdot \left( \frac{1}{\rho} \mathbf{G}p_i^v \right)_i \rangle \\ &= - \sum_j \frac{m_{i-j}}{\rho_{i-j}^2} \frac{p_j^v - p_i^v}{r_{ij}} \mathbf{n}_{ij}^v \cdot \nabla W(r_{ij}/2, h) \end{aligned} \quad (12)$$

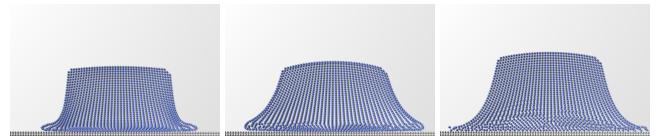
Substituting both equation 12 and equation 10 into the pressure Poisson equation 5, we can get

$$\begin{aligned} &\sum_j \frac{m_{i-j}}{\rho_{i-j}^2} \frac{p_j^v - p_i^v}{r_{ij}} \mathbf{n}_{ij}^v \cdot \nabla W(r_{ij}/2, h) \\ &= \frac{1}{\Delta t} \sum_j \frac{m_{i-j}}{\rho_{i-j}} \tilde{\mathbf{v}}_{i-j} \cdot \nabla W(r_{ij}/2, h) \end{aligned} \quad (13)$$

to solve  $p_i^v$  for each particle  $P_i$ . As a result, the velocity of  $P_{i-j}$  can be updated as follows

$$\mathbf{v}_{i-j}^{new} = \tilde{\mathbf{v}}_{i-j} - \frac{\Delta t}{\rho_{i-j}} \mathbf{G}p_{i-j}^v \quad (14)$$

In our purely meshless method, the solvability of correct  $p_i^v$  largely depends on whether the particles are well-distributed. Since the staggered particles are recreated at each time step from the carrier particles, we only need to guarantee the well distribution of the carrier particles, which is equivalent to enforcing the density constraint. Without considering the density constraint, fluid particles can clump as shown in Figure 5 (left), which eventually leads to the simulation failure. However, solving another Poisson equation for  $p^d$  is time consuming, thus [Losasso et al. 2008] combined the two Poisson equations together to solve a mixed pressure  $p^d + p^v$ . Unfortunately, the particle clumping problem still exists, as shown



**Figure 5:** Left: only enforcing the velocity constraint, resulting in heavy particle clumping in certain direction. Middle: enforcing both the density constraint and the velocity constraint by solving a mixed pressure Poisson equation, the problem of particle clumping still exists. Right: enforcing both the density constraint and the velocity constraint with our method, robust simulation is obtained.

in Figure 5 (middle). To achieve both the efficiency and stability, we apply a similar strategy with [Raveendran et al. 2011] to iteratively adjust the fluid density before solving the pressure Poisson equation. At the beginning of each iteration, the equation of state (EOS) from LPSPH [He et al. 2012] is used to estimate the density constraint pressure  $p^d$ . Then, a generalized formulation is used to compute the density constraint force  $\mathbf{F}_i^{density}$  (detailed in Section 5.1) to avoid the penetration of fluid particles into the solid. The iterative adjusting of the carrier particles' position maintains a relatively uniform distribution of the carrier particles. Figure 5 (right) shows the good result obtained with our method by enforcing both the density constraint and velocity constraint.

## 4.3 Remapping

As we know, the primary strength of grid-based methods comes at the simplicity of discretizing problem domain and enforcing the

incompressibility. Unfortunately, these methods suffer from the difficulty of fluid tracking. To solve this problem, marker particles [Harlow et al. 1965; Enright et al. 2002a] are commonly used as an assistant for advection. There are typically two approaches to handle this advection. Particle-in-cell (PIC) [Harlow 1962] was an early method designed to solve compressive flow, but with a major problem of numerical dissipation which was caused by repeatedly averaging and interpolating the grid variables. Later, [Brackbill and Ruppel 1986] proposed the fluid-implicit-particle (FLIP) method, and achieved indissipative model by only considering the changes of the grid velocity. In our staggered meshless method, the carrier particles play the same role as marker particles for the fluid tracking, thus requiring us to update the carrier particle velocity after solving the pressure field  $p^v$ . However, the position updating will be delayed until all other explicit forces are added up according to equation 3. To avoid the numerical dissipation, we derive a new velocity updating method for the staggered meshless method based on the principle of momentum conservation.

Following the discussion of [Robinson-Mosher et al. 2008], we introduce a virtual pressure  $p_{i-j}^v$  at  $\mathbf{x}_{i-j}$  (Figure 6 (left)) and unlump the staggered particle  $P_{i-j}$  into  $P_{i-}$  and  $P_{j-}$  to compute their respective gradient. Given the pressure  $p_i^v$  and  $p_j^v$  of particle  $P_i$  and  $P_j$ , the momentum increments of  $P_{i-}$  and  $P_{j-}$  can be written as

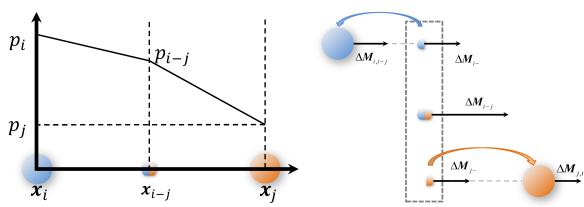
$$\begin{aligned}\Delta \mathbf{M}_{i-} &= -\Delta t \frac{p_i^v - p_{i-j}^v}{r_{i-}} V_{i-} \mathbf{n}_{ij}^v \\ \Delta \mathbf{M}_{j-} &= -\Delta t \frac{p_j^v - p_{i-j}^v}{r_{j-}} V_{j-} \mathbf{n}_{ji}^v\end{aligned}\quad (15)$$

where  $r_{i-}$  denotes  $|\mathbf{x}_i - \mathbf{x}_{i-j}|$  and  $r_{j-}$  denotes  $|\mathbf{x}_j - \mathbf{x}_{i-j}|$ . Then, the new velocities of  $P_{i-}$  and  $P_{j-}$  can be updated to

$$\mathbf{v}_{i-}^{new} = \mathbf{v}_i + \frac{\Delta \mathbf{M}_{i-}}{m_{i-}} \quad \text{and} \quad \mathbf{v}_{j-}^{new} = \mathbf{v}_j + \frac{\Delta \mathbf{M}_{j-}}{m_{j-}} \quad (16)$$

Now, if we lump  $P_{i-}$  and  $P_{j-}$  together,  $\mathbf{v}_{i-}^{new}$  and  $\mathbf{v}_{j-}^{new}$  should satisfy  $\mathbf{v}_{i-}^{new} = \mathbf{v}_{j-}^{new}$ . Besides, according to the principal of momentum conservation law, the total momentum change of  $P_{i-}$  and  $P_{j-}$  should be equal to the momentum change of  $P_{i-j}$ , which results another relationship  $\Delta \mathbf{M}_{i-} + \Delta \mathbf{M}_{j-} = \Delta \mathbf{M}_{i-j}$ . Combining these two relations together, we can eliminate the unknown variables and compute the momentum change for  $P_{i-}$  and  $P_{j-}$  as

$$\begin{aligned}\Delta \mathbf{M}_{i-} &= \frac{m_{i-}}{m_{i-j}} \Delta \mathbf{M}_{i-j} + \frac{m_{i-j} - m_{j-}}{m_{i-j}} (\mathbf{v}_j - \mathbf{v}_i) \\ \Delta \mathbf{M}_{j-} &= \frac{m_{j-}}{m_{i-j}} \Delta \mathbf{M}_{i-j} + \frac{m_{i-j} - m_{i-}}{m_{i-j}} (\mathbf{v}_i - \mathbf{v}_j)\end{aligned}\quad (17)$$



**Figure 6:** Left: A graph of the pressure gradient profile on the staggered particle  $P_{i-j}$ . Right: momentum transfer due to the staggered particle  $P_{i-j}$ .

From another point of view, if we consider  $P_{i-}$  and  $P_{j-}$  as a fraction of particle  $P_i$  and  $P_j$ , the momentum increments of  $P_i$  and  $P_j$

due to  $P_{i-}$  and  $P_{j-}$  are written as

$$\begin{aligned}\Delta \mathbf{M}_{i,i-j} &= -\Delta t \frac{p_{i-j}^v - p_i^v}{r_{i-}} V_{i-} \mathbf{n}_{ij}^v \\ \Delta \mathbf{M}_{j,i-j} &= -\Delta t \frac{p_{i-j}^v - p_j^v}{r_{j-}} V_{j-} \mathbf{n}_{ji}^v\end{aligned}\quad (18)$$

By considering  $\mathbf{n}_{ji}^v = -\mathbf{n}_{ij}^v$ , we can easily verify that  $\Delta \mathbf{M}_{i,i-j} = \Delta \mathbf{M}_{i-}$  and  $\Delta \mathbf{M}_{j,i-j} = \Delta \mathbf{M}_{j-}$ .

Eventually, we can derive the following equation to update the fluid velocity

$$\Delta \mathbf{v}_i^f = -\frac{\Delta t}{N_i} \sum_j \frac{1}{\rho_{i-j}} \mathbf{G} p_{i-j}^v \quad (19)$$

where we have ignored the last term of equation 17 for the reason that the viscous force will be modeled separately in section 5.1.

In fact, we can view our staggered SPH method as a variant of the FLIP method with a moving computing grid where we have elegantly combined the traditional fixed grid and marker particles together to form a unified meshless method. Although FLIP usually performs more efficiently for fluid simulations with compact domains and static boundaries, our method are more intuitive and powerful for the fluid simulation with dynamic boundaries, such as the two-way solid-fluid coupling.

## 5 Solid-Fluid Coupling

Since our method focuses on the unified handling for both the fluid and solid, we use the same way as fluid to discretize a rigid body into a set of solid particles. At each particle  $P_i^s$ , we define its signed distance to surface as  $\phi_i^s$  [Adams et al. 2007]. By applying the discretization technique in SPH, the local unit normal pointing outward can then be computed as follows

$$\mathbf{n}_i^s = -\alpha_i \sum_j \frac{m_j^s}{\rho_j^s} \phi_i^s \nabla W(|\mathbf{x}_i^s - \mathbf{x}_j^s|, h) \quad (20)$$

where  $\alpha_i$  is a coefficient for the normalization. In order to constrain the motion of a rigid body to translation and rotation, we need to accumulate the total forces acting on the solid particles

$$\mathbf{F}^s = \sum_i \mathbf{F}_i^s \quad \text{and} \quad \tau^s = \sum_i (\mathbf{x}_i^s - \mathbf{X}_i^s) \times \mathbf{F}_i^s \quad (21)$$

where  $\mathbf{X}_i^s$  is the center of mass,  $\mathbf{F}_i^s$  denotes the total force exerted on the solid particle  $P_i^s$ ,  $\mathbf{F}^s$  and  $\tau^s$  represent the total force and torque of the rigid body.

In the traditional meshless methods, there is no efficient way to impose essential boundary conditions due to the lack of the Kronecker delta property of shape functions, resulting in the difficulty in implementing the solid-fluid coupling. However, in our method, a set of staggered particles are also created between the fluid particles and solid particles, which can be used to impose various boundary conditions easily. According to equation 3 and 5, three force types are involved in computing the total force exerted on each particle  $P_i$ , which include the viscous force  $\mathbf{F}_i^{viscous}$ , the density constraint force  $\mathbf{F}_i^{density}$  and the velocity constraint force  $\mathbf{F}_i^{velocity}$ . If we look deep into the interaction between two neighboring solid particles, we can easily verify that the resultant force and torque are zero for the rigid body. Thus we only need to consider the fluid-fluid and fluid-solid interactions. In the following context, if not specified, we will not distinguish between the fluid and solid particles.

## 5.1 Generalized Forces

To handle the density discontinuities at interface between the fluid and solid, we compute the density of fluid particle  $P_i^f$  according to the formulation for multiple fluids proposed by [Solenthaler and Pajarola 2008]

$$\rho_i^f = m_i^f \sum_j W(|\mathbf{x}_i^f - \mathbf{x}_j|, h) \quad (22)$$

The density of solid particle is initialized at the beginning of simulation and remains unchanged. The density constraint pressure  $p_i^d$  for  $P_i$  can then be computed according to LSPSH [He et al. 2012] with the special case  $\hbar_{max} = 0$ . In order to treat  $\mathbf{F}_i^{density}$  and  $\mathbf{F}_i^{viscous}$  in a unified manner, we now generalize the formulations given by [Müller et al. 2003].

**Generalized pressure force** In continuum mechanics, it usually regards the fluid stress tensor as the sum of a spherical part  $-p\mathbf{I}$  and a deviatoric part  $\mu\nabla\mathbf{v}$ . If we consider a surface element of the solid with normal  $\mathbf{n}$ , the spherical part will always act the pressure force on the surface in the normal direction. To reflect this feature, the generalized pressure force can be written as follows

$$\mathbf{F}_i^{density} = - \sum_j m_j \frac{p_i^d + p_j^d}{2\rho_j} \frac{\partial W(|\mathbf{x}_i - \mathbf{x}_j|, h)}{\partial \mathbf{n}_{ij}^d} \quad (23)$$

where  $\partial W(|\mathbf{x}_i - \mathbf{x}_j|, h)/\partial \mathbf{n}_{ij}^d$  represents the directional derivative of the smoothing function  $W$ ,  $\mathbf{n}_{ij}^d$  indicates the force direction of particle  $P_j$  exerting on  $P_i$ . Depending on the different types of  $P_i$  and  $P_j$ ,  $\mathbf{n}_{ij}^d$  can be written as

$$\mathbf{n}_{ij}^d = \begin{cases} (\mathbf{x}_j - \mathbf{x}_i)/|\mathbf{x}_j - \mathbf{x}_i| & i^f, j^f \\ \mathbf{n}_i^s & i^s, j^f \\ -\mathbf{n}_j^s & i^f, j^s \end{cases}$$

where  $i^f$  means particle  $P_i$  belongs to the fluid and  $i^s$  means particle  $P_i$  belongs to the solid. From the above formulation, we can also find that the generalized pressure force retains the symmetry of forces between  $P_i$  and  $P_j$ .

**Generalized viscous force** Now we take a look at the deviatoric part  $\mu\nabla\mathbf{v}$ . It acts the viscous force tangentially on the surface element, which also indicates that only the tangential part of the fluid velocity will be damped down by the solid. The generalized viscous force can then be written as

$$\mathbf{F}_i^{viscous} = \sum_j \frac{m_j \sqrt{\mu_i \mu_j}}{\rho_j} T(\mathbf{v}_j - \mathbf{v}_i) \nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|, h) \quad (24)$$

The velocity mapping operator  $T$  is defined as below

$$T = \begin{cases} \mathbf{I} & i^f, j^f \\ \mathbf{I} - (\mathbf{n}_i^s)(\mathbf{n}_i^s)^T & i^s, j^f \\ \mathbf{I} - (\mathbf{n}_j^s)(\mathbf{n}_j^s)^T & i^f, j^s \end{cases}$$

where  $\mathbf{I}$  is the identity mapping. When either particle  $P_i$  or  $P_j$  belongs to the solid,  $T$  maps the relative velocity  $\mathbf{v}_j - \mathbf{v}_i$  to its tangential component. As a result, various slip boundary conditions can be achieved by controlling the value of  $\mu^s$  for the solid.

## 5.2 Boundary Conditions

The solvability of the pressure Poisson equation 5 also depends on the appropriate imposing of the free surface and solid boundary conditions on the fluid. To impose the free surface boundary condition, we detect the surface particles near air to set their velocity constraint pressure  $p_i^v$  to zero. To impose various slip solid boundary conditions, difficulties arise in previous *incompressible* SPH (ISPH) [Lee et al. 2008] method. However, our method can easily handle various slip conditions by enforcing different velocity constraints on the staggered particles, including both the no-slip and free-slip conditions.

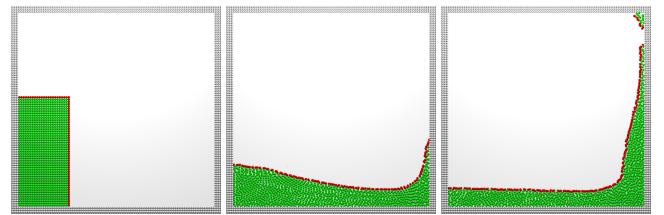
**Free surface boundary condition** Since no particles exist beyond the free surface, one can take into account the asymmetry of particle distribution to identify free surface particles. We first define the measurement of asymmetry for a fluid particle  $P_i^f$  as follows

$$Asy_i^f = \left| \mathbf{x}_i^f - \sum_j \mathbf{x}_j \tilde{W}(|\mathbf{x}_i^f - \mathbf{x}_j|, h) \right| \quad (25)$$

where  $\tilde{W}$  represents the normalized smoothing function satisfying  $\sum_j \tilde{W}(\mathbf{x}_j - \mathbf{x}_i, h) = 1$ . The larger value  $Asy_i^f$  is, the more likely  $P_i^f$  belongs to the free surface. However, if the fluid particle density  $\rho_i^f$  is too far away from the reference density  $\bar{\rho}_i^f$ ,  $Asy_i^f$  cannot give a good reflection on the particle distribution feature. According to these two understandings, we give the criterion for identifying the free surface particles in our implementation as those meet

$$Asy_i^f > 0.3h \text{ or } \rho_i^f < 0.7\bar{\rho}_i^f$$

Besides, a further step is required to identify the isolated regions where no surface particles are detected. We explicitly mark all these particles as surface particles to ensure the solvability of the linear system. The identified free surface particles at different times are shown in Figure 7 with the 2D dam break case. We can find that almost all surface particles on the free surface have been detected, but not all. However, this is acceptable as the undetected surface particles still have pressure very close to zero. By including the solid particles, we also effectively avoid the false detection for free surface particles at the interface between the fluid and solid.



**Figure 7:** Detected boundary particles (red) for the 2D dam break case.

**Solid boundary conditions** Now, we would like to impose constraints on the relative velocity of the fluid and solid. The no-slip condition for viscous fluids states that the fluid at the solid surface has zero velocity relative to the solid boundary. However, the free-slip boundary condition only constrains the normal component of the relative velocity to zero. To impose these constraints, we can view the staggered particles between the fluid particles and solid particles as a discretization of the contact surface between the fluid and solid. Since each staggered particle is only related to its two

neighboring carrier particles, we can easily impose different boundary conditions on the staggered particle. By taking into account a staggered particle  $P_{i-j}$  between a fluid particle  $P_i^f$  and a solid particle  $P_j^s$ , the solid boundary velocity at position  $\mathbf{x}_{i-j}$  is set to the velocity of  $P_j^s$ , which is  $\mathbf{u}_{i-j}^s = \mathbf{U}^s + \omega \times (\mathbf{x}_j - \mathbf{X}^s)$  where  $\mathbf{U}^s$  is the translational velocity and  $\omega$  is the angular velocity of the rigid body. Since the no-slip condition requires that  $\mathbf{v}_{i-j}^{new} = \mathbf{u}_{i-j}^s$ , we can easily get

$$\mathbf{G}p_{i-j} = \frac{\rho_{i-j}}{\Delta t} (\tilde{\mathbf{v}}_{i-j} - \mathbf{u}_{i-j}^s) \quad (26)$$

by substituting the requirement into equation 14. In the same way, we can also substitute the free-slip condition  $\mathbf{v}_{i-j}^{new} \cdot \mathbf{n}_j^s = \mathbf{u}_{i-j}^s \cdot \mathbf{n}_j^s$  into equation 14 to get

$$\mathbf{G}p_{i-j} = \frac{\rho_{i-j}}{\Delta t} [(\tilde{\mathbf{v}}_{i-j} - \mathbf{u}_{i-j}^s) \cdot \mathbf{n}_j^s] \mathbf{n}_j^s \quad (27)$$

which will leave the tangential component of the relative velocity unchanged. Accompanied with the free surface boundary condition, either equation 26 or equation 27 can be substituted into the discretized pressure Poisson equation 13 to implement the required slip boundary condition. For completeness, we give the formulation of the velocity constraint force  $\mathbf{F}_i^{velocity}$  as follows

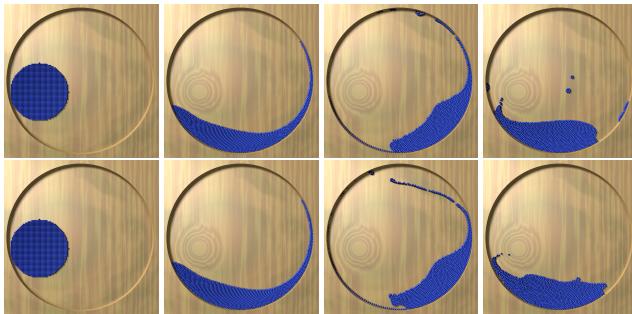
$$\mathbf{F}_i^{velocity} = -m_{i-j} \sum_j \frac{1}{\rho_{i-j}} \mathbf{G}p_{i-j}^v \quad (28)$$

Here  $\mathbf{G}p_{i-j}^v$  is related to both the type of the two neighboring carrier particles and the imposed solid wall boundary condition, which can be represented in the following unified formulation

$$\mathbf{G}p_{i-j}^v = \begin{cases} \left[ (p_j^v - p_i^v) / r_{ij} \right] \mathbf{n}_{ij}^v & i^f, j^f \\ \frac{\rho_{i-j}}{\Delta t} [(\tilde{\mathbf{v}}_{i-j} - \mathbf{u}_{i-j}^s) \cdot \mathbf{n}_{ij}^v] \mathbf{n}_{ij}^v & i^f, j^s \\ -\frac{\rho_{i-j}}{\Delta t} [(\tilde{\mathbf{v}}_{i-j} - \mathbf{u}_{i-j}^s) \cdot \mathbf{n}_{ij}^v] \mathbf{n}_{ij}^v & i^s, j^f \end{cases}$$

where  $\mathbf{n}_{ij}^v$  for the free-slip boundary condition is just the same as  $\mathbf{n}_{ij}^d$ , and  $\mathbf{n}_{ij}^v$  for the no-slip boundary condition is defined below

$$\mathbf{n}_{ij}^v = \begin{cases} (\mathbf{x}_j - \mathbf{x}_i) / |\mathbf{x}_j - \mathbf{x}_i| & i^f, j^f \\ (\tilde{\mathbf{v}}_{i-j} - \mathbf{u}_{i-j}^s) / |\tilde{\mathbf{v}}_{i-j} - \mathbf{u}_{i-j}^s| & i^f, j^s \text{ or } i^s, j^f \end{cases}$$



**Figure 8:** Top: the fluid sticks to the solid wall unnaturally with the usage of the standard free-slip solid wall boundary condition. Bottom: the fluid can be separated from the solid wall plausibly with our treatment.

When implementing the free-slip boundary condition, a common numerical artifact resulting from  $\mathbf{v}_{i-j}^{new} \cdot \mathbf{n}_j^s = \mathbf{u}_{i-j}^s \cdot \mathbf{n}_j^s$  is that the fluid will stick to the solid wall and even crawl along the wall. In grid-based method, [Batty et al. 2007] solved this problem

by introducing a complementarity condition to convert the kinetic energy minimization problem into an inequality-constrained QP problem. Unfortunately, the low performance of QP solver reduces the efficiency of the simulation. In our staggered meshless method, to avoid the particle sticking, we use a trick by only imposing the solid boundary condition on the staggered particles that approach the wall. More precisely, it states as  $(\mathbf{u}_{i-j}^s - \mathbf{v}_{i-j}^{new}) \cdot \mathbf{n}_j^s \geq 0$ . Compared to the treatment given by [Batty et al. 2007], our method does not reduce the simulation efficiency since no boundary condition of inequality is included. Figure 8 demonstrates a comparison between the standard free-slip boundary condition and the no-sticking free-slip boundary condition.

## 6 Implementation

### Algorithm 1 Staggered Meshless Coupling

---

```

1 while animating do
2   add body force and generalized viscous force  $\mathbf{F}^{viscous}$ 
3   predict fluid and solid velocities and positions
4   compute density constraint pressure  $p^d$  (Algorithm 2)
5   update solid velocities
6   compute velocity constraint pressure  $p^v$  (Algorithm 3)
7   update fluid and solid velocities with  $\mathbf{F}^{velocity}$ 
```

---

### Algorithm 2 compute density constraint pressure $p^d$

---

```

1 while ( $iter < minIterations$ ) do
2   compute generalized fluid density  $\rho^f$ 
3   compute  $p^d$  according to LSPSH with  $\hbar_{max} = 0$ 
4   compute generalized pressure force  $\mathbf{F}^{density}$ 
5   update fluid velocities and positions with  $\mathbf{F}^{density}$ 
```

---

### Algorithm 3 Compute Velocity Constraint Pressure $p^v$

---

```

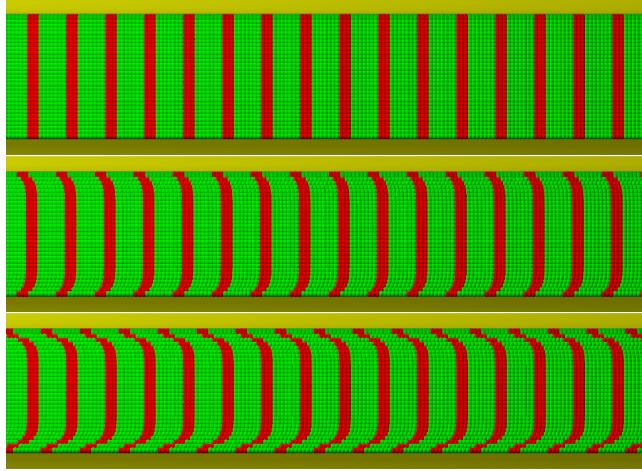
1 create staggered particles
2 track surface particles and set their pressure to zero
3 apply solid wall boundary conditions
4 solve pressure Poisson equation with PCG for  $p^v$ 
```

---

At the initialization stage, we apply the same resolution to discretize both the fluid and solid into a set of carrier particles. Then we use a common approach termed *weak coupling* to implement the two-way coupling, which is shown in Algorithm 1 for an overview. After adding the gravitational and generalized viscous forces, we first predict both the fluid and solid velocities and positions. Then, we enforce the density constraint on the fluid as shown in Algorithm 2. After that, the solid velocities are updated by the pressure constraint force  $\mathbf{F}^{density}$ . At the moment, the fluid velocity field does not satisfy the incompressibility condition, we need to compute the velocity constraint pressure  $p^v$  (Algorithm 3) with our staggered meshless method to ensure that the fluid velocity field is divergence free with required boundary conditions. Finally, we update the velocities of the fluid and the solid in the following step. To handle scripted rigid bodies (also including static scenes), instead of setting the rigid bodies' mass to  $\infty$ , we use a trick to assign  $\rho^s = \bar{\rho}^f$  and compute  $N_i^s$  in the same way as  $N_i^f$  to achieve the continuity of physical quantities across solid/fluid boundaries, thus making it unified in treating scripted and non-scripted rigid bodies.

In Algorithm 2, we implement the iterative-LSPSH method given by [He et al. 2012]. Here, we only take a fixed iteration number of 3, which is sufficient for us to avoid the particle clumping problem. Besides, to get rid of the unnatural contraction of surface particles caused by the particle deficiency, we require the density constraint pressure  $p^d$  should meet  $p^d \geq 0$ . To further avoid the explicit

collision detection step between the fluid and solid, the generalized pressure force  $\mathbf{F}^{density}$  is computed to prevent the fluid particle penetration into the solid. When computing  $p^v$  in Algorithm 3, the newly created staggered particles are stored in each particle's neighbor list. To obtain higher efficiency, both particle  $P_i$  and  $P_j$  save a copy of the staggered particle  $P_{i-j}$  in their neighbor lists, making it straightforward to extend the standard SPH into our staggered SPH. After applying the correct free surface and solid wall boundary conditions, we use a preconditioned conjugate gradient algorithm to solve the linear systems since the approximate Laplacian operator is symmetric and positive semi-definite. More details about the convergence will be discussed in section 7.1.



**Figure 9:** Fluid running through a tube in 2D with various solid wall boundary condition including free-slip (top), no-slip (bottom) and other slip condition (middle).

To implement the free-slip boundary condition, we impose the boundary condition given by equation 27 and set the value of  $\mu^s = 0$  to ensure that only the normal part of the relative velocity is restricted and the tangential part will not be damped down by the solid. Otherwise, to implement the no-slip boundary condition, we only need to replace equation 27 with 26 to restrict the relative velocity between the fluid and solid. However, if we would like to implement other slip boundary conditions, we need to include the generalized viscous force by setting the required value of  $\mu$  to damp down the tangential part of the relative velocity. Here we simply set  $\mu^s = \mu^f$  and keep the treatment for the normal part the same as implementing the free-slip boundary condition. Figure 9 gives an illustration of above three different cases by simulating the fluid running through a tube.

## 7 Results and Discussions

All the timings in the following cases are given for a single-threaded execution on an Intel Xeon W3550 workstation with four 3.07GHz processors and 6 GB of main memory.

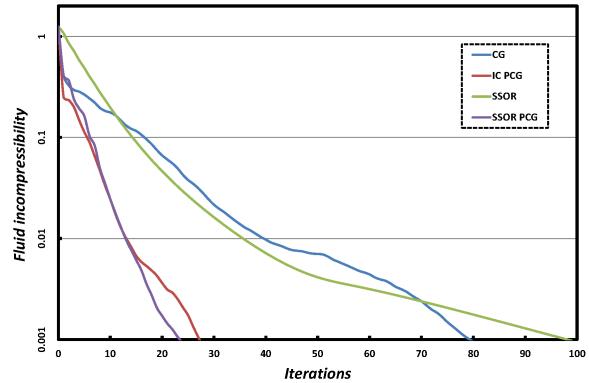
### 7.1 Convergence Analysis

In this part, the convergence in solving the pressure Poisson equation 5 is discussed. By invoking the continuity equation, we can define the degree to which the fluid incompressibility is achieved as follows

$$E = \frac{1}{N} \sum_{i=1}^N \frac{|\Delta \rho_i|}{\rho_i} = \frac{1}{N} \sum_{i=1}^N \frac{|\nabla \cdot \mathbf{v}_i|}{\Delta t} \quad (29)$$

where the summation is taken over the inner fluid particles only.

We choose a single physical update step at  $t = 0.1\text{s}$  in the example shown in Figure 1 (right) and aim to test several iterative strategies for solving  $p^v$ . The most commonly used algorithm in the grid-based methods for solving the symmetric positive semidefinite linear systems is the *conjugate gradient* (CG) algorithm. However, the problem with CG is that it requires a large number of iterations to converge to the desired accuracy (40 iterations for  $E = 1\%$  and 80 iterations for  $E = 0.1\%$ , as shown in Figure 10 and Table 1).



**Figure 10:** Convergence of solving  $p^v$  for the 3D dam break simulation at  $t = 0.1\text{s}$  with different numerical iterative methods.

	$E$	$t_{pre}$ [s]	$t_{iter}$ [s]	iterations	$t_{total}$ [s]
CG	1%	—	0.218	40	8.84
	0.1%	—	0.218	80	17.56
IC PCG	1%	5.80	0.324	14	10.46
	0.1%	5.80	0.324	28	14.99
SSOR	1%	—	0.203	36	7.43
	0.1%	—	0.203	99	20.22
SSOR PCG	1%	0.172	0.324	13	4.51
	0.1%	0.172	0.324	24	8.08

**Table 1:** Detailed time statistics for solving  $p^v$  with different numerical iterative methods.

An improvement to accelerate the convergence rate is to use the preconditioning techniques, among which the *incomplete Cholesky* (IC) preconditioner is one of the most popular preconditioners for its simplicity and efficiency [Benzi 2002]. As a result, the convergence rate of *incomplete Cholesky preconditioned conjugate gradient* (IC PCG) reaches nearly a factor of 3 speed-up over the CG (14 iterations for  $E = 1\%$  and 28 iterations for  $E = 0.1\%$ ). Nevertheless, attention is still needed since both the construction and application of the IC preconditioner require extra computational costs. For  $E = 1\%$ , it can be found from Table 1 that the enhanced convergence rate of IC PCG over CG fails to compensate for these extra computational costs, resulting in even longer total computational time. To explore more efficient preconditioners, we refer to another commonly used algorithm which is called the *symmetric successive over relaxation* (SSOR) method. Although the SSOR algorithm itself does not provide a sufficient amount of convergence rate, we find that the *SSOR preconditioned conjugate gradient* (SSOR PCG) method shows a quite similar convergence rate as the IC PCG. Additionally, the SSOR preconditioner, just like the Jacobi preconditioner, can be derived from the coefficient matrix with little CPU effort. Thus, for both  $E = 1\%$  and  $E = 0.1\%$ , the

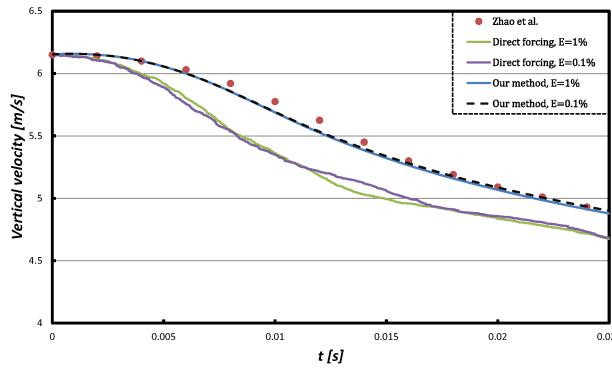
SSOR PCG method gives the best performance among all the mentioned strategies. In the implementation, the SSOR preconditioner is parameterized by  $\omega$  as follow [Barrett 1994]

$$M(\omega) = \frac{1}{2-\omega} \left( \frac{1}{\omega} D + L \right) \left( \frac{1}{\omega} D \right)^{-1} \left( \frac{1}{\omega} D + L \right)^T$$

where  $D$  and  $L$  represent the diagonal and lower parts of the coefficient matrix, respectively. Besides,  $\omega$  is set to 1.7 according to a variety of numerical tests, which gives the best performance.

## 7.2 Accuracy and Performance Comparison

In order to evaluate the accuracy and efficiency of the proposed staggered meshless coupling method, we set up a validation test case using a symmetric wedge impacting on the free water surface. The experiment has been carried out by [Zhao et al. 1996]. At  $t = 0s$ , a wedge with a deadrise angle of  $30^\circ$  and weighing  $141kg$  was dropped and entered the water surface with a  $6.15m/s$  initial vertical velocity. During the experiment, the vertical motion was the only degree of freedom allowed to the wedge. For more details, refer to their original work. The motion speed was experimentally recorded and plotted in Figure 11.



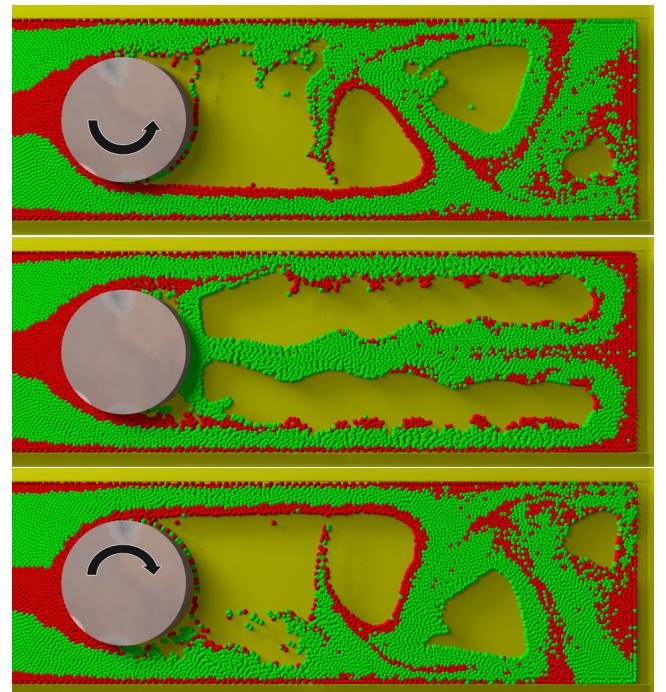
**Figure 11:** The falling velocity of a  $30^\circ$  deadrise angle wedge, timed from the moment of entry.

In the numerical setup, a particle spacing of  $\Delta x = 0.01m$  is selected to discretize both the water, which is  $2m$  wide and  $1m$  deep, and the wedge, which has a breadth of  $0.5m$ . Then, different simulations were made to run  $0.025s$  of real time with both the direct forcing coupling method [Becker et al. 2009] and the staggered coupling method. In the former case, the timestep for the simulation is restricted by the Courant-Friedrichs-Lowy (CFL) condition. Besides, since the method enforces the non-penetration of the fluid particles with the particle-particle collision detection algorithm, the timestep should be further restricted to ensure that two particles do not move more than one diameter distance towards each other in each step. In contrast, the latter method is free of the above mentioned limitations due to the use of an implicit incompressible fluid solver and the avoidance of the explicit particle-to-particle boundary handling, thus allowing significantly larger time steps. The simulation results for both  $E = 1\%$  and  $E = 0.1\%$  are depicted in Figure 11 to make a comparison with the experimental results, from which it is found that our method shows a better agreement with the experimental data. More details about the accuracy and efficiency are summarized in Table 2 where we have chosen the maximum allowable time step for both the methods. Specifically, it can also be noted from the table that the deviation from the experimental data decreases only by a small fraction (approximately 0.2%) as the compression is changed from 1% to 0.1%. Therefore, it should be sufficient to only enforce  $E = 1\%$  to get more efficiency.

	#p / #p <sup>f</sup> / #p <sup>s</sup>	E	$\Delta t [s]$	Deviation	$t_{total} [s]$	Speedup
<i>Direct Forcing</i>	23143/ 20301/ 2842	1%	$5.0 \times 10^{-5}$	4.5%	55	—
		0.1%	$1.6 \times 10^{-5}$	4.3%	173	—
		1%	$6.0 \times 10^{-4}$	0.63%	12	4.6
<i>Our method</i>		0.1%	$6.0 \times 10^{-4}$	0.46%	16	10.8

**Table 2:** The accuracy and performance of our staggered meshless method compared to the direct forcing method [Becker et al. 2009].

## 7.3 More Effects Demonstration

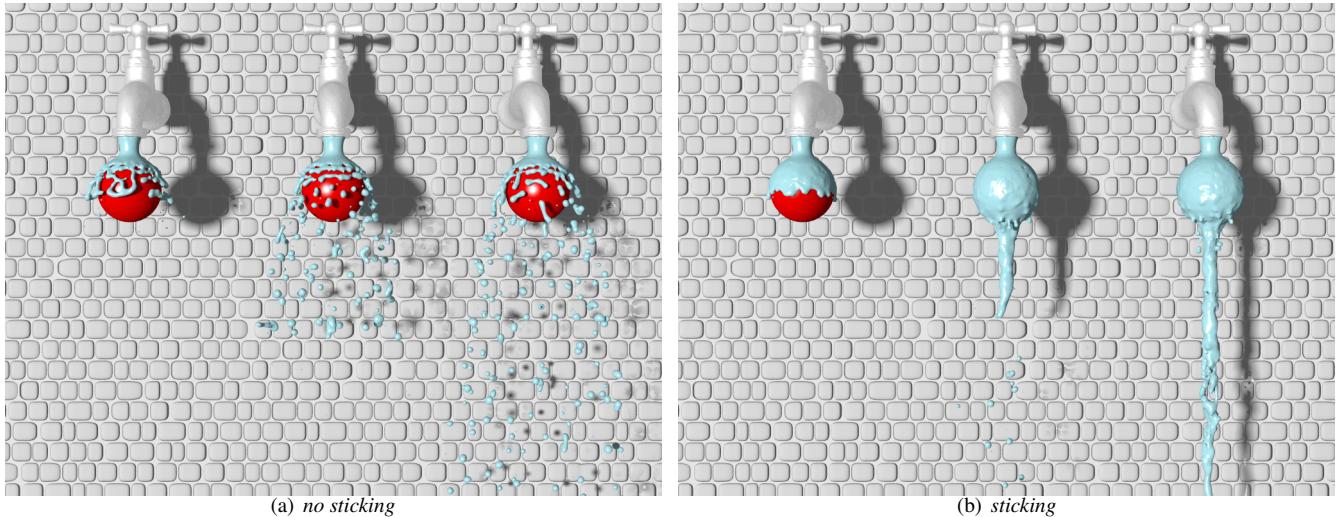


**Figure 12:** Fluid flowing around a cylinder with different angular velocities,  $10rad/s$  (top),  $0rad/s$  (middle),  $-10rad/s$  (bottom).

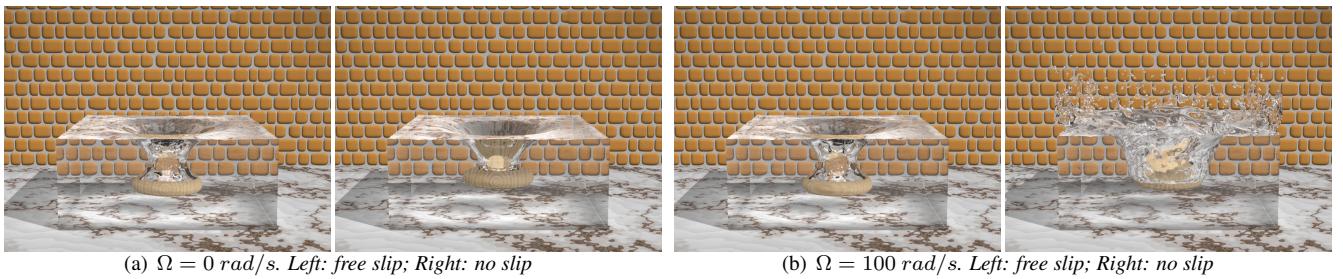
Figure 12 illustrates the one-way coupling with the combinations of both the free-slip and no-slip boundary conditions in a two-dimensional scene demonstrating fluid flowing around a cylinder. The outer boundary is static with free-slip boundary condition. For the inner cylinder, the no-slip boundary condition is imposed with three different angular velocities including  $10 rad/s$  (top),  $0 rad/s$  (middle) and  $-10 rad/s$  (bottom). Another case of dam break in three dimensions is shown in Figure 1 (right) where the darker pillars are rotating clockwise with an angular velocity of  $10 rad/s$  while the brighter ones are rotating anticlockwise with  $-10 rad/s$ .

Although we have discussed how to avoid the fluid sticking to wall in section 5.2, in some cases, it is required to model the realistic cohesion of the liquid to the solids. By imposing the fluid sticking to the solids, our method is able to model the phenomenon of fluid flowing around a ball as shown in Figure 13 (right) [Schechter and Bridson 2012], which is quite difficult for the standard SPH. Besides, a comparison is also shown in Figure 13 (left) where the fluid sticking to the solids has been avoided.

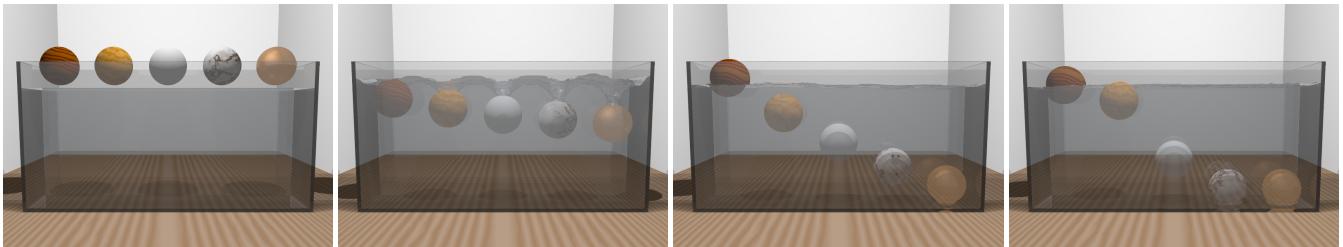
In Figure 14, the two-way coupling is tested by dropping an elliptical stone into a tank of water with varying initial angular velocities and different slip conditions. For the non-rotating case, similar



**Figure 13:** Tap water pouring on a sphere near the outlet. (a) By avoiding fluid sticking to solids, the water leaves the sphere freely. (b) By imposing fluid sticking to solids, the realistic stream flowing around the sphere is captured.



**Figure 14:** An elliptical stone dropping into a tank of water with different slip conditions and angular velocities  $\Omega$ .



**Figure 15:** Five balls with varying densities (from left to right:  $550 \text{ kg/m}^3$ ,  $775 \text{ kg/m}^3$ ,  $1000 \text{ kg/m}^3$ ,  $1225 \text{ kg/m}^3$ ,  $1450 \text{ kg/m}^3$ ) plunging into a tank of water ( $1000 \text{ kg/m}^3$ ) and correct buoyancy effects are captured.

Examples	#P/#P <sup>f</sup> /#P <sup>s</sup>	E	t <sub>neighbor</sub> [s]	t <sub>p<sup>d</sup></sub> [s]	t <sub>surface</sub> [s]	t <sub>creating</sub> [s]	t <sub>assemble</sub> [s]	t <sub>solving</sub> [s]	t <sub>step</sub> [s]
Fig.1(right)	1891k/1084k/807k	1%	17.2	5.5	1.2	1.9	0.97	4.6	31.6
Fig.12	14.2k/5.4k/19.6k	1%	0.070	0.048	0.008	0.014	0.009	0.035	0.19
Fig.13	86.1k/63.7k/22.4k	1%	0.738	0.202	0.066	0.098	0.052	0.125	1.35
Fig.14	562k/418k/144k	1%	5.3	1.9	0.45	0.67	0.37	3.6	12.7
Fig.15	158k/101k/57k	1%	1.4	0.51	0.20	0.17	0.094	0.44	3.0

**Table 3:** Timing statistics for the various components of our method. Here t<sub>neighbor</sub>, t<sub>p<sup>d</sup></sub>, t<sub>surface</sub>, t<sub>creating</sub>, t<sub>assemble</sub> and t<sub>solving</sub> represent the required time for finding neighbors, enforcing density constraint, marking fluid surface particles, creating staggered particles, assembling coefficient matrix and solving p<sup>v</sup>, respectively. The neighbor list size is set to 30 in the implementation, which also indicates that at most 30 staggered particles will be created for each carrier particle. However, per-particle storage is very small so large numbers of staggered particles are not a problem for computer memory.

simulation results were found except for a minor difference in the sinking velocities. For the free slip case with  $\Omega = 100 \text{ rad/s}$ , since only the normal part of the relative velocity is corrected, the stone's angular velocity is only slightly affected. However, in the no slip case, the stone's angular velocity is quickly damped down by the water.

Figure 15 illustrates that the buoyancy effect is properly captured by using the staggered meshless coupling method. In this example, five balls with varying densities (from left to right:  $550 \text{ kg/m}^3$ ,  $775 \text{ kg/m}^3$ ,  $1000 \text{ kg/m}^3$ ,  $1225 \text{ kg/m}^3$ ,  $1450 \text{ kg/m}^3$ ) were plunged into a tank of water whose reference density is set to  $1000 \text{ kg/m}^3$ .

Finally, the particle numbers and the performance measurements for each time step are summarized in Table 3. Because the computational time varies during the simulation, we only report the averaged time for one time step. In all the cases, we use a fixed integration time step of  $0.001\text{s}$  and reconstruct the fluid surface with the skinning method proposed by [Bhattacharya et al. 2011].

## 8 Conclusion

In this paper, a staggered meshless method is presented to overcome the difficulties in simulating the solid-fluid coupling with the meshless methods. The proposed method creates a set of staggered particles at each time step to separate the velocity field from the pressure field. As a result, the free surface and various solid wall boundary conditions can be applied to different types of the particles. To avoid the simulation failure, a new framework to enforce both the density constraint and the velocity constraint on the fluids is further proposed. In this way, various boundary conditions including the free-slip and no-slip can be easily implemented. The method is novel in that it provides a powerful tool to implement the stable one-way and two-way coupling between the fluids and rigid bodies.

However, the method does have some limitations. First, our current implementation is not yet suitable to deal with the infinitesimally thin solids, such as the cloth. To enable the fluid interaction with the thin solids, either very small timestep or robust collision detection algorithm [Becker et al. 2009] is required. Second, more accurate contact handling technique among the multi-bodies is needed. In the current computations, we only used the simple penalty forces between the rigid bodies to avoid penetration. Nonetheless, it would be an interesting future work to model the accurate behaviors of both the accurate multi-body contacts and the deformations with the staggered particles. Furthermore, attentions will also be paid to achieve the GPU implementations of the method.

## Acknowledgements

We would like to thank anonymous reviewers for their valuable advices on improving the paper. Support for authors in IS-CAS came from the National Basic Research Program of China (No.2009CB320804), the National Natural Science Foundation of China (No.61173059, 61135003). Support for authors in Peking University came from the National Basic Research Program of China (No.2010CB328002), the National Natural Science Foundation of China (No.60925007, 60833007, 61121002, 61170205).

## References

- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. 2007. Adaptively sampled particle fluids. *ACM Transactions on Graphics (TOG)* 26, 3, 48–54.
- BARRETT, R. 1994. *Templates for the solution of linear systems: building blocks for iterative methods*. No. 43. Society for Industrial Mathematics.
- BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 26, 3, 100.
- BECKER, M., TESSENDORF, H., AND TESCHNER, M. 2009. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics* 15, 3, 493–503.
- BENZI, M. 2002. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics* 182, 2, 418–477.
- BHATACHARYA, H., GAO, Y., AND BARGTEIL, A. 2011. A level-set method for skinning animated particle data. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 17–24.
- BODIN, K., LACOURSIERE, C., AND SERVIN, M. 2011. Constraint fluids. *IEEE Transactions on Visualization and Computer Graphics*, 99, 1–1.
- BRACKBILL, J., AND RUPPEL, H. 1986. Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics* 65, 2, 314–343.
- CARLSON, M., MUCHA, P., AND TURK, G. 2004. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics (TOG)* 23, 3, 377–384.
- CUMMINS, S., AND RUDMAN, M. 1999. An SPH projection method. *Journal of Computational Physics* 152, 2, 584–607.
- ENRIGHT, D., FEDIKIW, R., FERZIGER, J., AND MITCHELL, I. 2002. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 183, 1, 83–116.
- ENRIGHT, D., MARSCHNER, S., AND FEDIKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Transactions on Graphics (TOG)* 21, 3, 736–744.
- FOSTER, N., AND FEDIKIW, R. 2001. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 23–30.
- FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical models and image processing* 58, 5, 471–483.
- GÉNEVAUX, O., HABIBI, A., AND DISCHLER, J. 2003. Simulating fluid-solid interaction. In *Graphics Interface*, 31–38.
- GUENDELMAN, E., SELLE, A., LOSASSO, F., AND FEDIKIW, R. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics (TOG)* 24, 3, 973–981.
- HARLOW, F., WELCH, J., ET AL. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids* 8, 12, 2182.
- HARLOW, F. 1962. The particle-in-cell method for numerical solution of problems in fluid dynamics. Tech. rep., Los Alamos Scientific Lab., N. Mex.
- HE, X., LIU, N., LI, S., WANG, H., AND WANG, G. 2012. Local Poisson SPH for viscous incompressible fluids. *Computer Graphics Forum* 00, 0, 1–11.
- HOUSTON, B., BOND, C., AND WIEBE, M. 2003. A unified approach for modeling complex occlusions in fluid simulations. In *ACM SIGGRAPH 2003 Sketches & Applications*, ACM, 1–1.

- KEISER, R., ADAMS, B., GASSER, D., BAZZI, P., DUTRÉ, P., AND GROSS, M. 2005. A unified lagrangian approach to solid-fluid animation. In *Eurographics/IEEE VGTC Symposium Proceedings on Point-Based Graphics*, IEEE, 125–148.
- LEE, E., MOULINEC, C., XU, R., VIOLEAU, D., LAURENCE, D., AND STANSBY, P. 2008. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics* 227, 18, 8417–8436.
- LENAERTS, T., ADAMS, B., AND DUTRÉ, P. 2008. Porous flow in particle-based fluid simulations. *ACM Transactions on Graphics (TOG)* 27, 3, 49.
- LIU, M., AND LIU, G. 2010. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering* 17, 1, 25–76.
- LIU, J., KOSHIZUKA, S., AND OKA, Y. 2005. A hybrid particle-mesh method for viscous, incompressible, multiphase flows. *Journal of Computational Physics* 202, 1, 65–93.
- LOSASSO, F., TALTON, J., KWATRA, N., AND FEDKIW, R. 2008. Two-way coupled SPH and particle level set fluid simulation. *Visualization and Computer Graphics, IEEE Transactions on* 14, 4, 797–804.
- MONAGHAN, J. 1994. Simulating free surface flows with SPH. *Journal of computational physics* 110, 399–399.
- MONAGHAN, J. 2005. Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68, 1703.
- MORRIS, J., FOX, P., AND ZHU, Y. 1997. Modeling low Reynolds number incompressible flows using SPH. *Journal of computational physics* 136, 1, 214–226.
- MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 154–159.
- MÜLLER, M., KEISER, R., NEALEN, A., PAULY, M., GROSS, M., AND ALEXA, M. 2004. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 141–151.
- MÜLLER, M., SCHIRM, S., TESCHNER, M., HEIDELBERGER, B., AND GROSS, M. 2004. Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds* 15, 3-4, 159–171.
- NGUYEN, V., RABCZUK, T., BORDAS, S., AND DUFLOT, M. 2008. Meshless methods: a review and computer implementation aspects. *Mathematics and Computers in Simulation* 79, 3, 763–813.
- OGER, G., DORING, M., ALESSANDRINI, B., AND FERRANT, P. 2006. Two-dimensional SPH simulations of wedge water entries. *Journal of Computational Physics* 213, 2, 803–822.
- PETRONETTO, F., PAIVA, A., LAGE, M., TAVARES, G., LOPES, H., AND LEWINER, T. 2010. Meshless helmholtz-hodge decomposition. *IEEE Transactions on Visualization and Computer Graphics* 16, 2, 338–349.
- RANDLES, P., AND LIBERSKY, L. 2000. Normalized SPH with stress points. *Int. J. Numer. Meth. Engng* 48, 1445, 1462.
- RAVEENDRAN, K., WOJTAN, C., AND TURK, G. 2011. Hybrid smoothed particle hydrodynamics. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA ’11, 33–42.
- ROBINSON-MOSHER, A., SHINAR, T., GRETARSSON, J., SU, J., AND FEDKIW, R. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. 46.
- ROBINSON-MOSHER, A., ENGLISH, R., AND FEDKIW, R. 2009. Accurate tangential velocities for solid fluid coupling. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 227–236.
- SCHECHTER, H., AND BRIDSON, R. 2012. Ghost sph for animating water. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)* 31, 4.
- SHAO, S., AND LO, E. 2003. Incompressible sph method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources* 26, 7, 787–800.
- SOLENTHALER, B., AND GROSS, M. 2011. Two-scale particle simulation. *ACM Transactions on Graphics (TOG)* 30, 4, 81.
- SOLENTHALER, B., AND PAJAROLA, R. 2008. Density contrast SPH interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 211–218.
- SOLENTHALER, B., AND PAJAROLA, R. 2009. Predictive-corrective incompressible SPH. *ACM Transactions on Graphics (TOG)* 28, 3, 1–6.
- SOLENTHALER, B., SCHLÄFLI, J., AND PAJAROLA, R. 2007. A unified particle model for fluid-solid interactions. *Computer Animation and Virtual Worlds* 18, 1, 69–82.
- STAM, J. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 121–128.
- TAKAHASHI, T., UEKI, H., KUNIMATSU, A., AND FUJII, H. 2002. The simulation of fluid-rigid body interaction. In *ACM SIGGRAPH 2002 conference abstracts and applications*, ACM, 266–266.
- VIGNJEVIC, R., CAMPBELL, J., AND LIBERSKY, L. 2000. A treatment of zero-energy modes in the smoothed particle hydrodynamics method. *Computer methods in Applied mechanics and Engineering* 184, 1, 67–85.
- YILDIZ, M., ROOK, R., AND SULEMAN, A. 2009. SPH with the multiple boundary tangent method. *International Journal for Numerical Methods in Engineering* 77, 10, 1416–1438.
- YNGVE, G., O'BRIEN, J., AND HODGINS, J. 2000. Animating explosions. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 29–36.
- ZHAO, R., FALTINSEN, O., AND AARSNES, J. 1996. Water entry of arbitrary two-dimensional sections with and without flow separation. In *Proc. 21st Symposium on Naval Hydrodynamics*, 408–423.