

Predictive-Corrective Incompressible SPH

B. Solenthaler *
University of Zurich

R. Pajarola †
University of Zurich



Figure 1: Three examples produced with our incompressible simulation: (Left) 2M particles splashing against the simulation boundaries. (Center) Close-up view of a wave tank. (Right) A fluid represented by 700k particles colliding with cylinder obstacles.

Abstract

We present a novel, incompressible fluid simulation method based on the Lagrangian *Smoothed Particle Hydrodynamics* (SPH) model. In our method, incompressibility is enforced by using a prediction-correction scheme to determine the particle pressures. For this, the information about density fluctuations is actively propagated through the fluid and pressure values are updated until the targeted density is satisfied. With this approach, we avoid the computational expenses of solving a pressure Poisson equation, while still being able to use large time steps in the simulation. The achieved results show that our *predictive-corrective incompressible* SPH (PCISPH) method clearly outperforms the commonly used *weakly compressible* SPH (WCSPH) model by more than an order of magnitude while the computations are in good agreement with the WCSPH results.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation.

Keywords: fluid simulation, SPH, incompressibility

1 Introduction and Previous Work

Enforcing incompressibility in fully particle-based fluid simulations represents the most expensive part of the whole simulation

process and thus renders particle methods less attractive for high quality and photorealistic water animations. In the context of Smoothed Particle Hydrodynamics (SPH), two different strategies have been pursued to model incompressibility. First, the weakly compressible SPH (WCSPH) method has been used where pressure is modeled using a stiff equation of state (EOS), and second, incompressibility has been achieved by solving a pressure Poisson equation. Although both methods satisfy incompressibility, the computational expenses of simulating high resolution fluid animations are too large for practical use.

In the standard SPH and WCSPH model the particle pressures are determined by an EOS. The characteristics of this equation and the stiffness parameter determine the speed of the acoustic waves in a medium. The EOS-based SPH with low stiffness according to [Desbrun and Cani 1996] was used in a series of papers to simulate water [Müller et al. 2003; Adams et al. 2007], multiple fluids [Müller et al. 2005; Solenthaler and Pajarola 2008], fluid-solid coupling [Müller et al. 2004b; Lenaerts et al. 2008], melting solids [Müller et al. 2004a; Keiser et al. 2005; Solenthaler et al. 2007], and fluid control [Thürey et al. 2006]. In contrast to the standard SPH formulation, WCSPH uses a stiff EOS [Monaghan 2005; Becker and Teschner 2007; Becker et al. 2009] resulting in acoustic waves traveling closer to their real speed through the medium. Typically, the stiffness value is chosen so large that the density fluctuations do not exceed 1%. The required stiffness value to achieve this, however, is difficult or even impossible to determine before running the simulation. Consequently, an animator cannot get around extensive testing and parameter tuning. Another drawback is that WCSPH imposes a severe time step restriction as the stiffness of the fluid usually dominates the Courant-Friedrichs-Levy (CFL) condition. Thus the computational cost increases with decreasing compressibility – since higher stiffness requires smaller time steps, making it infeasible to simulate high resolution fluids within reasonable time.

Rather than simulating acoustic waves, incompressibility in Lagrangian methods can be enforced by solving a pressure projection similar to Eulerian methods (e.g. [Enright et al. 2002]). These incompressible SPH (ISPH) methods first integrate the velocity field in time without enforcing incompressibility. Then, either the intermediate velocity field [Cummins and Rudman 1999], the resulting variation in particle density [Shao 2006], or both [J. Liu and Oka

*e-mail: solenthaler@ifi.uzh.ch

†e-mail: pajarola@acm.org

2005; Hu and Adams 2007; Losasso et al. 2008] are projected onto a divergence-free space to satisfy incompressibility through a pressure Poisson equation. With these ISPH methods density fluctuations of 1% to 3% have been reported. A problem with these methods, however, is the complexity to formulate and solve the equation system on unstructured particle configurations. Although ISPH allows larger time steps than WCSPH, the computational cost per physics step is much higher. A Poisson solver was also used in [Premoze et al. 2003] for the particle method Moving-Particle Semi-Implicit (MPS), increasing the cost per physics time step enormously. In contrast to the fully Lagrangian models, [Zhu and Bridson 2005] propose to use an auxiliary background grid to simplify the equation system to a sparse set of linear equations which can be efficiently solved. A similar hybrid solver for vorticity confinement is presented in [Selle et al. 2005]. In [Losasso et al. 2008], a two-way coupled level set method with an SPH solver is introduced to simulate dense and diffuse water volumes. They demonstrate how to enforce incompressibility and target the particle number density with a single Poisson solve.

In this paper, we propose a novel, fully Lagrangian, incompressible SPH method featuring the advantages of both WCSPH and ISPH in one model, namely low computational cost per physics update and large time steps. Our method makes use of a prediction-correction scheme which propagates the estimated density values through the fluid and updates the pressures in such a way that incompressibility is achieved. The propagation stops as soon as a previously user-defined density variation limit is reached for each individual particle. We will show in this paper that our new *predictive-corrective incompressible* SPH (PCISPH) method outperforms WCSPH by more than an order of magnitude while the computations are in good agreement with the WCSPH results. The efficiency of our method enables an animator to produce high-resolution fluid animations within reasonable time without compressibility artifacts.

2 PCISPH Model

2.1 Basic SPH / WCSPH Algorithm

In SPH, liquids are approximated by artificial, slightly compressible fluids. The basic SPH method is summarized in Algorithm 1. In each physics update, the local neighborhood N_i of each particle i is found and then used to evaluate the density, pressure, and the resulting forces acting on each particle [Monaghan 1992]. The density ρ_i of particle i at location \mathbf{x}_i can be found by summing up the weighted contributions of the neighboring particles j

$$\rho_i = m \sum_j W(\mathbf{x}_{ij}, h), \quad (1)$$

where m is the particle mass (we assume that all particles have equal masses), W is the weighting kernel with smoothing length h , and $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. The pressure p_i of a particle is then derived from the EOS according to [Batchelor 1967]

$$p_i = \frac{k\rho_0}{\gamma} \left(\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right),$$

where k is a stiffness parameter and ρ_0 the reference density. [Desbrun and Cani 1996] use a γ of 1 and a small value for k , while in WCSPH (e.g. [Monaghan 2005]) γ is set to 7 and k is chosen so that the speed of sound is large enough to keep the density fluctuations small ($\sim 1\%$). Note that the CFL condition [Courant et al. 1967] requires smaller time steps for stiffer fluids which increases the overall computation cost tremendously when simulating water. The pressure force field is directly derived from the Navier-Stokes

Algorithm 1 SPH / WCSPH

```

1 while animating do
2   for all  $i$  do
3     find neighborhoods  $N_i(t)$ 
4   for all  $i$  do
5     compute density  $\rho_i(t)$ 
6     compute pressure  $p_i(t)$ 
7   for all  $i$  do
8     compute forces  $\mathbf{F}^{p,v,g,ext}(t)$ 
9   for all  $i$  do
10    compute new velocity  $\mathbf{v}_i(t+1)$ 
11    compute new position  $\mathbf{x}_i(t+1)$ 

```

Algorithm 2 PCISPH

```

1 while animating do
2   for all  $i$  do
3     find neighborhoods  $N_i(t)$ 
4   for all  $i$  do
5     compute forces  $\mathbf{F}^{v,g,ext}(t)$ 
6     initialize pressure  $p(t) = 0.0$ 
7     initialize pressure force  $\mathbf{F}^p(t) = 0.0$ 
8   while  $(\rho_{err}^*(t+1) > \eta) \text{ || } (\text{iter} < \text{minIterations})$  do
9     for all  $i$  do
10      predict velocity  $\mathbf{v}_i^*(t+1)$ 
11      predict position  $\mathbf{x}_i^*(t+1)$ 
12    for all  $i$  do
13      predict density  $\rho_i^*(t+1)$ 
14      predict density variation  $\rho_{err}^*(t+1)$ 
15      update pressure  $p_i(t) += f(\rho_{err}^*(t+1))$ 
16    for all  $i$  do
17      compute pressure force  $\mathbf{F}^p(t)$ 
18    for all  $i$  do
19      compute new velocity  $\mathbf{v}_i(t+1)$ 
20      compute new position  $\mathbf{x}_i(t+1)$ 

```

equations and given by

$$\mathbf{F}_i^p = -m^2 \sum_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W(\mathbf{x}_{ij}, h).$$

In our implementation, we use the viscous force and weighting kernels presented in [Monaghan 1992].

2.2 PCISPH Algorithm

To avoid the time step restriction of WCSPH we propose to use a prediction-correction scheme based on the SPH algorithm (PCISPH). In our method, the velocities and positions are temporarily forwarded in time and the new particle densities are estimated. Then, for each particle, the predicted variation from the reference density is computed and used to update the pressure values, which in turn enter the recomputation of the pressure forces. Similar to a Jacobi iteration for linear systems, this process is iterated until it converges, i.e. until all particle density fluctuations are smaller than a user-defined threshold η (for example 1%). Note that this is a nonlinear problem since we include collision handling and updated kernel values in our iteration process. As a final step, the velocities and positions of the next physics update step are computed. The PCISPH method is illustrated in Algorithm 2.

2.3 Pressure Derivation

One of the main difficulties is to derive the pressure change from the predicted density variation (line 15 of Algorithm 2). This pressure

update is executed in each iteration, reducing the density fluctuation of the particle. The aim is to find a pressure p which changes the particle positions in such a way that the predicted density corresponds to the reference density. Over the course of this section, a set of approximations will be made to derive a simple update rule for the pressure (Equations 8 to 10). Although the approximations increase the number of convergence iterations which are needed until the desired density fluctuation limit is reached, they keep the final pressure update rule simple and thus efficient to compute.

For a given kernel smoothing length h , the density at a point in time $t + 1$ is computed using the SPH density summation equation analogously to Equation 1

$$\begin{aligned}\rho_i(t+1) &= m \sum_j W(\mathbf{x}_i(t+1) - \mathbf{x}_j(t+1)) \\ &= m \sum_j W(\mathbf{x}_i(t) + \Delta\mathbf{x}_i(t) - \mathbf{x}_j(t) - \Delta\mathbf{x}_j(t)) \\ &= m \sum_j W(\mathbf{d}_{ij}(t) + \Delta\mathbf{d}_{ij}(t))\end{aligned}$$

where $\mathbf{d}_{ij}(t) = \mathbf{x}_i(t) - \mathbf{x}_j(t)$, and $\Delta\mathbf{d}_{ij}(t) = \Delta\mathbf{x}_i(t) - \Delta\mathbf{x}_j(t)$. Assuming that $\Delta\mathbf{d}_{ij}$ is relatively small, the first order Taylor approximation can be applied to the term $W(\mathbf{d}_{ij}(t) + \Delta\mathbf{d}_{ij}(t))$ resulting in

$$\begin{aligned}\rho_i(t+1) &= m \sum_j W(\mathbf{d}_{ij}(t)) + \nabla W(\mathbf{d}_{ij}(t)) \cdot \Delta\mathbf{d}_{ij}(t) \\ &= m \sum_j W(\mathbf{x}_i(t) - \mathbf{x}_j(t)) + \\ &\quad m \sum_j \nabla W(\mathbf{x}_i(t) - \mathbf{x}_j(t)) \cdot (\Delta\mathbf{x}_i(t) - \Delta\mathbf{x}_j(t)) \\ &= \rho_i(t) + \Delta\rho_i(t).\end{aligned}$$

In this equation, the term $\Delta\rho_i(t)$ is unknown and, as we show later, a function of p which we are looking for. After reformulation and using $W_{ij} = W(\mathbf{x}_i(t) - \mathbf{x}_j(t))$ we get

$$\begin{aligned}\Delta\rho_i(t) &= m \sum_j \nabla W_{ij} \cdot (\Delta\mathbf{x}_i(t) - \Delta\mathbf{x}_j(t)) \\ &= m \left(\sum_j \nabla W_{ij} \Delta\mathbf{x}_i(t) - \sum_j \nabla W_{ij} \Delta\mathbf{x}_j(t) \right) \\ &= m \left(\Delta\mathbf{x}_i(t) \sum_j \nabla W_{ij} - \sum_j \nabla W_{ij} \Delta\mathbf{x}_j(t) \right) \quad (2)\end{aligned}$$

$\Delta\mathbf{x}$ can be derived from the time integration scheme (Leap-Frog). Neglecting all forces but the pressure force we get

$$\Delta\mathbf{x}_i = \Delta t^2 \frac{\mathbf{F}_i^p}{m}. \quad (3)$$

If we make the simplistic assumption that neighbors have equal pressures \tilde{p}_i and that the density corresponds to the rest density ρ_0 (according to the incompressibility condition), this results in

$$\mathbf{F}_i^p = -m^2 \sum_j \left(\frac{\tilde{p}_i}{\rho_0^2} + \frac{\tilde{p}_i}{\rho_0^2} \right) \nabla W_{ij} = -m^2 \frac{2\tilde{p}_i}{\rho_0^2} \sum_j \nabla W_{ij}. \quad (4)$$

Inserting Equation 4 into Equation 3 we get

$$\Delta\mathbf{x}_i = -\Delta t^2 m \frac{2\tilde{p}_i}{\rho_0^2} \sum_j \nabla W_{ij}. \quad (5)$$

Due to the pressure p_i of particle i the position of a neighboring particle changes by $\Delta\mathbf{x}_{j|i}$. As the pressure forces are symmetric, particle j gets the following contribution from i

$$\mathbf{F}_{j|i}^p = m^2 \left(\frac{\tilde{p}_i}{\rho_0^2} + \frac{\tilde{p}_i}{\rho_0^2} \right) \nabla W_{ij} = m^2 \frac{2\tilde{p}_i}{\rho_0^2} \nabla W_{ij},$$

and the position of j changes by

$$\Delta\mathbf{x}_{j|i} = \Delta t^2 m \frac{2\tilde{p}_i}{\rho_0^2} \nabla W_{ij}. \quad (6)$$

Note that we only consider the effect of the central particle i here, i.e. $\Delta\mathbf{x}_j = \Delta\mathbf{x}_{j|i}$. Equation 5 and Equation 6 can now be inserted into Equation 2 resulting in

$$\begin{aligned}\Delta\rho_i(t) &= m \left(-\Delta t^2 m \frac{2\tilde{p}_i}{\rho_0^2} \sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} - \right. \\ &\quad \left. \sum_j (\nabla W_{ij} \cdot \Delta t^2 m \frac{2\tilde{p}_i}{\rho_0^2} \nabla W_{ij}) \right) \\ &= \Delta t^2 m^2 \frac{2\tilde{p}_i}{\rho_0^2} \left(-\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} - \right. \\ &\quad \left. \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}) \right)\end{aligned}$$

After solving for \tilde{p}_i we get

$$\tilde{p}_i = \frac{\Delta\rho_i(t)}{\beta(-\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} - \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}))} \quad (7)$$

where β is

$$\beta = \Delta t^2 m^2 \frac{2}{\rho_0^2}.$$

The meaning of Equation 7 is that a pressure \tilde{p}_i is needed to achieve a change in density of $\Delta\rho_i(t)$. As we know the predicted density error $\rho_{err,i}^* = \rho_i^* - \rho_0$ of a particle, we can thus reverse that error by applying a pressure of

$$\tilde{p}_i = \frac{-\rho_{err,i}^*}{\beta(-\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} - \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}))}.$$

This formula shows problems in situations where i is suffering from particle deficiency in the neighborhood resulting in falsified values. To circumvent that problem, we precompute a single scaling factor δ according to the following formula which is evaluated for a prototype particle with a filled neighborhood. The resulting value is then used for all particles. Finally, we end up with the following equations which are used in the PCISPH method

$$\delta = \frac{-1}{\beta(-\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} - \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}))} \quad (8)$$

and

$$\tilde{p}_i = \delta \rho_{err,i}^*. \quad (9)$$

Since we repeat the prediction-correction step as long as the incompressibility condition is not yet satisfied, the correction pressures of the individual iterations are accumulated as indicated on line 15 of Algorithm 2

$$p_i += \tilde{p}_i. \quad (10)$$

2.4 Implementation

2.4.1 Neighborhood Approximation

Before predicting the density $\rho_i^*(t+1)$ of a particle (line 13 of Algorithm 2), the neighborhood should be recomputed using the predicted positions $\mathbf{x}^*(t+1)$. However, for efficiency reasons we reuse the current neighbors $N_i(t)$ at time t and only recompute the distances and the kernel values. This approximation leads to small errors in the density and pressure estimates. In the case of density overestimation the final real densities show lower fluctuations than

the requested threshold η . In the opposite case – density underestimation – the correction loop might be aborted prematurely. Such situations are not yet handled in the current implementation but can be avoided by using sufficiently small time steps, or by recomputing the neighborhoods in these particular situations.

2.4.2 Information Propagation

To limit temporal fluctuations in the resulting pressure field we found it advantageous to employ a minimum number of iterations in the pressure update loop. This gives the particles enough time to propagate information about predicted particle locations. We found a minimum of 3 iterations generally sufficient to achieve a low level of pressure fluctuations.

3 Results

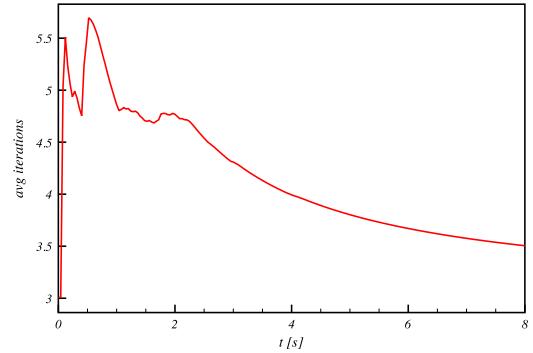
3.1 Performance Comparison

We set up a test scene (Figure 2) to compare the simulation times and visual results of both the commonly used WCSPH and our new PCISPH method. The performance measurements and simulation data are summarized in Table 1. All timings are given for an Intel Core2 2.66 GHz CPU.

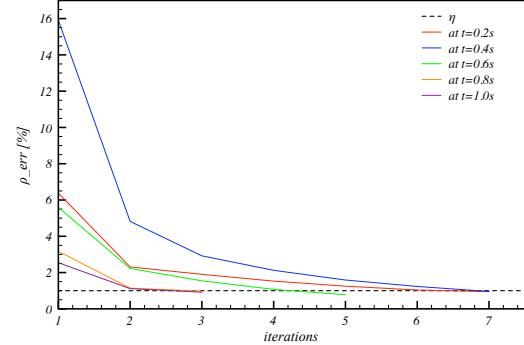
We executed different simulation runs with varying particle resolutions (10k and 100k) and varying error threshold η (1% and 0.1%) which defines the maximally allowed density fluctuation from the reference density. The 10k and 100k examples have corresponding scene setups but different fluid discretizations, meaning that a particle in the 10k example represents a larger fluid volume than one in the 100k example. Since in SPH a particle always needs to have around 30-40 neighbors, the support radius has to be increased with increasing particle volume, which in turn influences the time step size. The time step is set according to a CFL condition where the force terms, the stiffness parameter k , and the viscous term are involved [Monaghan 1992]. While in WCSPH the time step is dominated by k , it has no influence in PCISPH and can be omitted. Thus, for low viscosity fluids, the time step in PCISPH is dominated by the force terms, allowing significantly larger time steps than those used in WCSPH. The stiffness parameter k of WCSPH was determined by testing in such a way that η was satisfied, which was $k = 7 \cdot 10^4$ for $\eta = 1\%$, and $k = 6 \cdot 10^6$ for $\eta = 0.1\%$. In contrast, PCISPH does not have to cope with finding an appropriate stiffness value since the desired η can be specified directly. In the case of $\eta=1\%$, the time step of PCISPH is determined to be in fact 35 times larger than the one of WCSPH. With a smaller η the difference is even larger, $\eta = 0.1\%$ leads to an increase of the time step for PCISPH by a factor of 151. While in WCSPH the computation time per simulation step stays more or less constant, it varies in PCISPH since the time per simulation step depends on the number of executed convergence iterations. Therefore, we compare the overall computation time of WCSPH and PCISPH over the entire simulated time period. Although the cost per physics time step is higher with PCISPH than WCSPH, the overall speed-up over WCSPH still reaches a factor of 15 and 16 for $\eta = 1\%$ and 55 for $\eta = 0.1\%$, respectively.

3.2 Convergence Analysis

In the previously described test scenes, the average number of convergence iterations executed per physics step is between 3.24 and 4.46. Note that the particle resolution has no effect on the average number of iterations. For the simulation run with 100k particles the average number of iterations is plotted over time in Figure 3(a). The end time of 8s corresponds to the simulated real time.



(a) Average number of convergence iterations over time. After 8s of simulated real time, an average of 3.49 is reached.



(b) Several convergence examples at different points in time t .

Figure 3: Convergence statistics of the 100k particles simulation shown in Table 1 and Figure 2.

The peaks indicate particle collisions with the ground and the side walls as in such situations larger density errors are predicted. Figure 3(b) shows several examples of the convergence within a single physics update step. It can be seen that the density error is approximately halved after the first iteration and continuously reduced in the following iterations until the error drops below η . In our experience this algorithm proved to be very robust and we did not encounter any divergence problems. However, it is likely that certain particle configurations exist that might show such problems.

3.3 Visual Result

The physical behavior and visual results of WCSPH and PCISPH are compared in Figure 2. It can be seen that the PCISPH computations are in full agreement with the WCSPH results with only very minor detail differences. The comparison of WCSPH and PCISPH using a simulation time constraint of 298min is shown in Figure 4. While with PCISPH a resolution of 100k particles can be simulated within the given time (see the corresponding entry in Table 1), with WCSPH the resolution has to be reduced to 17k particles. The lower resolution leads to less surface details and notably damped fluid movement. A higher resolution example computed with PCISPH is shown in Figure 1 (center and right) and Figure 5 where a wave generator agitates a water body consisting of 700k particles to interact with cylindrical obstacles in a tank. In Figure 1 (left) and Figure 6, 2M particles are used to simulate the collapsing column example with PCISPH. In both of these simulations, a η of 1% is enforced which eliminates compression artifacts and enables realistic wave breaking and splashing behavior. In all examples, the surface of the fluid is reconstructed and rendered with the raytracing approach presented in [Solenthaler et al. 2007].

Model	η [%]	#p	k	Δt [s]	Δt ratio [s]	avgIterations	t_{sim} [min]	speed-up
WCSPH	1.0	10k	$7 \cdot 10^4$	3.78e-5	-	-	142.05	-
PCISPH	1.0	10k	-	0.0013	35	3.24	9.37	15.2
WCSPH	1.0	100k	$7 \cdot 10^4$	1.78e-5	-	-	4941.5	-
PCISPH	1.0	100k	-	0.00062	35	3.49	297.7	16.6
WCSPH	0.1	10k	$6 \cdot 10^6$	4.08e-6	-	-	1327.66	-
PCISPH	0.1	10k	-	0.00062	151.96	4.46	23.97	55.39

Table 1: Comparison of WCSPH and PCISPH. The stiffness value k of WCSPH is chosen so that the density fluctuation percentage is below η , and the time step size is determined according to the CFL condition. With our PCISPH method and $\eta=1\%$, a speed-up of a factor of 15 and 16 over WCSPH is reached. By restricting the error to $\eta=0.1\%$, PCISPH reduces the computation time by a factor of 55.

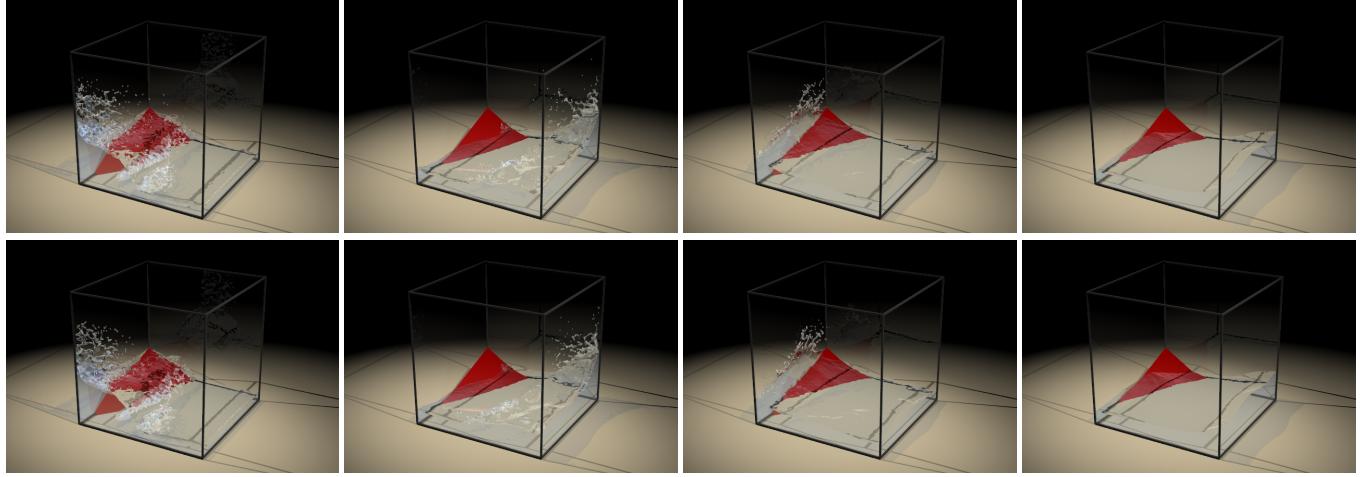


Figure 2: Side-by-side comparison of a fluid discretized by 100k particles and simulated with WCSPH (upper row) and PCISPH (lower row), respectively. The computations correspond to the statistics given in Table 1 for the 100k particles simulation.

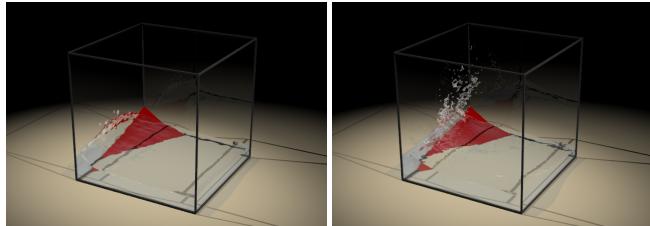


Figure 4: Comparison of WCSPH (left, 17k particles) and PCISPH (right, 100k particles) with equal computation times.

4 Conclusion

We proposed a novel incompressible SPH solver which combines the advantages of both WCSPH and ISPH in one model, namely low computational cost per physics update and large time steps. Our method includes a convergence loop which is executed in each physics update step consisting of a prediction and correction iteration. In each convergence iteration, the new particle positions and their densities are predicted and the variations from the reference density are computed. We have derived a formulation which relates the density fluctuation and the pressure, to reduce the density errors and to approach incompressibility. With this method, we gained a speed-up of more than an order of magnitude over the commonly used WCSPH method and we showed that the simulation results are in agreement with WCSPH.

One issue of the current implementation is the neighborhood ap-

proximation which can lead to underestimated density errors aborting the convergence loop prematurely as we have discussed in Section 2.4.1. This problem can be addressed by detecting such situations and adapting the time step size or recomputing the neighbors in this particular simulation step. Besides that, our current implementation does not yet account for the particle deficiency near boundaries. In these situations, density values are falsified and compression artifacts can occur. The inclusion of ghost particles in the density computation or the use of the corrected SPH formulation as described in [Becker et al. 2009] can solve this problem.

References

- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. *ACM Trans. Graph.* 26, 3, 48–54.
- BATCHELOR, G. 1967. *An Introduction to Fluid Dynamics*. Cambridge University Press.
- BECKER, M., AND TESCHNER, M. 2007. Weakly compressible SPH for free surface flows. In *Symposium on Computer Animation*, 209–217.
- 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.
- COURANT, R., FRIEDRICHHS, K., AND LEWY, H. 1967. On the partial difference equations of mathematical physics. *IBM J.* 11, 215–234.

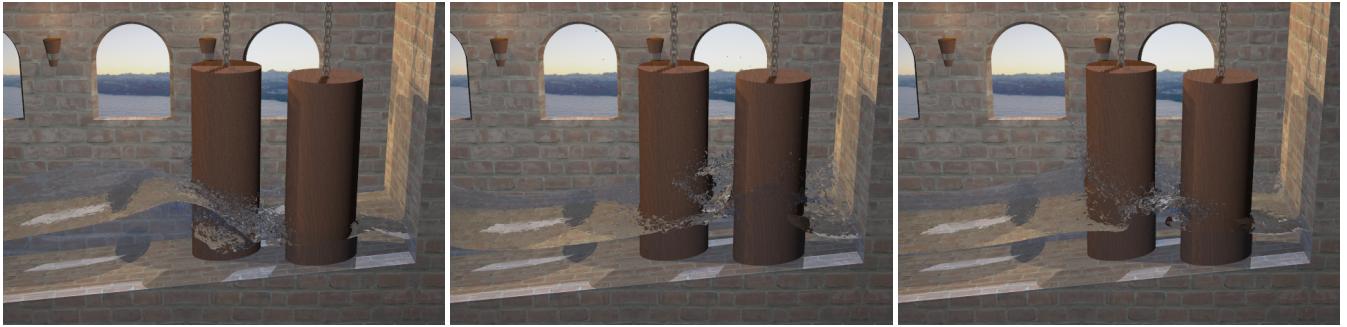


Figure 5: Wave breaking and splashing in a wave tank simulated with the proposed incompressible PCISPH method.

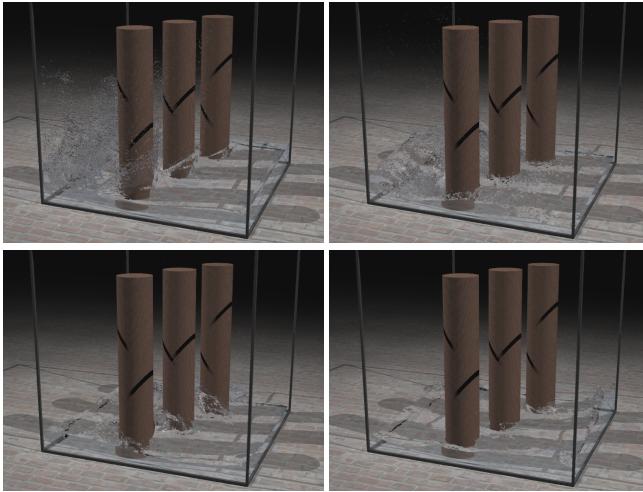


Figure 6: PCISPH simulation of 2M particles interacting with cylinder obstacles.

CUMMINS, S. J., AND RUDMAN, M. 1999. An SPH projection method. *J. Comput. Phys.* 152, 2, 584–607.

DESBRUN, M., AND CANI, M.-P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Eurographics Workshop on Computer Animation and Simulation*, 61–76.

ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Trans. Graph.* 21, 3, 736–744.

HU, X. Y., AND ADAMS, N. A. 2007. An incompressible multi-phase SPH method. *J. Comput. Phys.* 227, 1, 264–278.

J. LIU, S. K., AND OKA, Y. 2005. A hybrid particle-mesh method for viscous, incompressible, multiphase flows. *J. Comput. Phys.* 202, 1, 65–93.

KEISER, R., ADAMS, B., GASSER, D., BAZZI, P., DUTRE, P., AND GROSS, M. 2005. A unified Lagrangian approach to solid-fluid animation. In *Proceedings of Eurographics Symposium on Point-Based Graphics*, 125–133.

LENAERTS, T., ADAMS, B., AND DUTRÉ, P. 2008. Porous flow in particle-based fluid simulations. *ACM Trans. Graph.* 27, 3, 1–8.

LOSASSO, F., TALTON, J., KWATRA, J., AND FEDKIW, R. 2008. Two-way coupled SPH and particle level set fluid simulation. *IEEE TVCG* 14, 4, 797–804.

MONAGHAN, J. 1992. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Physics* 30, 543.

MONAGHAN, J. 2005. Smoothed particle hydrodynamics. *Rep. Prog. Phys.* 68, 1703–1759.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Symposium on Computer Animation*, 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 *Symposium on Computer Animation*, 141–151.

MÜLLER, M., SCHIRM, S., TESCHNER, M., HEIDELBERGER, B., AND GROSS, M. 2004. Interaction of fluids with deformable solids. *Journal of Computer Animation and Virtual Worlds* 15, 3-4, 159–171.

MÜLLER, M., SOLENTHALER, B., KEISER, R., AND GROSS, M. 2005. Particle-based fluid-fluid interaction. In *Symposium on Computer Animation*, 237–244.

PREMOZE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A., AND WHITAKER, R. T. 2003. Particle-based simulation of fluids. In *Proceedings of Eurographics*, 401–410.

SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.* 24, 3, 910–914.

SHAO, S. 2006. Incompressible SPH simulation of wave breaking and overtopping with turbulence modelling. *Int. J. Numer. Meth. Fluids* 50, 597–621.

SOLENTHALER, B., AND PAJAROLA, R. 2008. Density contrast SPH interfaces. In *Symposium on Computer Animation*, 211–218.

SOLENTHALER, B., SCHLÄFLI, J., AND PAJAROLA, R. 2007. A unified particle model for fluid-solid interactions. *Journal of Computer Animation and Virtual Worlds* 18, 1, 69–82.

THÜREY, N., KEISER, R., PAULY, M., AND RÜDE, U. 2006. Detail-preserving fluid control. In *Symposium on Computer Animation*, 7–15.

ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. *ACM Trans. Graph.* 24, 3, 965–972.