

# Boundary handling and adaptive time-stepping for PCISPH

Markus Ihmsen   Nadir Akinci   Marc Gissler   Matthias Teschner

University of Freiburg

---

## Abstract

*We present a novel boundary handling scheme for incompressible fluids based on Smoothed Particle Hydrodynamics (SPH). In combination with the predictive-corrective incompressible SPH (PCISPH) method, the boundary handling scheme allows for larger time steps compared to existing solutions. Furthermore, an adaptive time-stepping approach is proposed. The approach automatically estimates appropriate time steps independent of the scenario. Due to its adaptivity, the overall computation time of dynamic scenarios is significantly reduced compared to simulations with constant time steps.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

---

## 1. Introduction

The efficient animation of fast and turbulent fluids can be successfully realized with a variety of Eulerian [MMS04, TKPR06], Lagrangian [APKG07] and semi-Lagrangian [FF01, LTKF08] methods. This paper focuses on the simulation of incompressible fluids with large time steps using the Lagrangian Smoothed Particle Hydrodynamics (SPH) method.

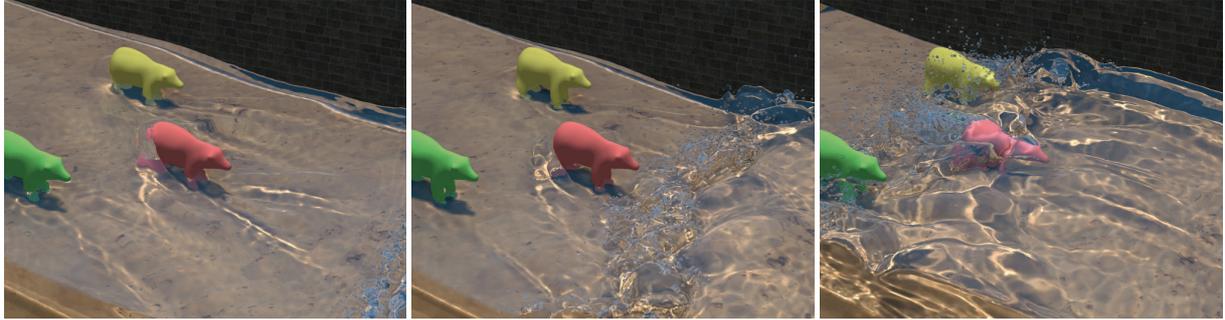
In SPH approaches, the enforcement of the incompressibility is a challenging problem and there exist different strategies to address this issue. For example, projection schemes - similar to Eulerian methods [EFFM02] - are successfully employed for particle based approaches, e. g. [CR99, LKO05, HA07, SBH09]. In these methods, velocities are projected onto a divergence-free space by solving a Poisson equation. Small density fluctuations are enforced. The projection, however, is rather expensive to compute for growing particle numbers.

In order to reduce expensive computations, state equations are commonly used in Computer Graphics. These equations relate the density with the pressure, while a stiffness parameter governs the compressibility. For compressible fluids, the gas equation [MCG03] and for weakly compressible SPH (WCSPH), the Tait equation [Mon94, BT07] are employed. In order to achieve small density fluctuations, however, the stiffness has to be chosen rather large. This results in large

pressure jumps for small density variations, which in turn requires small time steps.

Recently, a promising method for incompressible SPH has been proposed by Solenthaler and Pajarola [SP09]. In this approach, incompressibility is enforced by iteratively predicting and correcting the density fluctuation. This predictive-corrective incompressible SPH method (PCISPH) provides very smooth density and pressure distributions, resulting in significantly larger time steps compared to the WCSPH method. Although the computation time per simulation step is larger compared to WCSPH, the overall computation time is much smaller. In PCISPH, the density error is propagated within the fluid until the compression is resolved. For large time steps or huge impacts, however, the generally low number of required iterations can grow. In those cases, a smaller time step with less iterations might be more efficient.

The number of required iterations also depends on the boundary handling scheme. Basically, existing schemes apply penalty forces, e. g. [MST\*04], or force and position corrections, e. g. [BTT09] and [HKK07]. However, for large time steps, typically used in PCISPH, either stiff penalty forces are required or particle stacking occurs in high density regions. High density fluctuations can occur close to the fluid-boundary interface with a negative influence on the time step.



**Figure 1:** Wave scene. Gentle wave showering three bears, 1.8 million particles. Overall computation time 19.7 hours for 20 real seconds,  $\Delta t_{min} = 5.1 \cdot 10^{-4}$ ,  $\Delta t_{max} = 2.56 \cdot 10^{-3}$ .

**Our contribution:** In this paper, a predictive-corrective boundary handling method for one way fluid-rigid coupling is presented. In contrast to the direct forcing method of [BTT09], the pressure on the boundary is taken into account. Particle stacking is avoided and thus, high density differences at the boundary. The proposed scheme is applied to the PCISPH method and it is shown that high compressions are resolved more smoothly compared to existing boundary methods. Therefore, larger time steps can be used.

Furthermore, an adaptive time-stepping scheme for PCISPH is presented. This scheme automatically adapts the time step according to the density fluctuation, the maximum velocity and acceleration. The adaptive time-stepping reduces the overall computation time and can handle shocks. The initial time step is automatically determined.

In combination with PCISPH, the proposed boundary method can handle larger time steps compared to [BTT09] and [HKK07]. The adaptive time-stepping scheme further improves the performance up to an order of magnitude compared to constant time-stepping. A first example is illustrated in Fig. 1.

## 2. Related work

In most SPH simulations, boundary conditions are enforced using **penalty forces** that scale with the distance of the fluid particle to the boundary. [Mon94] sampled the boundary with particles which exert central Lennard-Jones penalty forces. For this formulation, the boundary force scales polynomially with the distance of the fluid particle. This causes large pressure variations in the fluid. Thus, the time step has to be chosen very small for weakly compressible fluids. Müller et al. [MST\*04] adapted the penalty based forces to simulate the interaction of SPH fluids with particle sampled deformable meshes. They achieved very realistic two-way coupling by using a smooth formulation of the Lennard-Jones forces. Later, Monaghan [Mon05, MK09] employed a

more stable form of the repulsion forces using a scaled version of the cubic spline kernel. But still, the main difficulty in pure penalty based approaches is controlling the stiffness parameter. This parameter has to be balanced such that boundary penetration is avoided, while not causing pressures that are too high. Therefore, small time steps are required to produce smooth pressure distributions. Unfortunately, the stiffness parameter and consequently the time step must be chosen carefully for each scenario.

In order to achieve a smoother pressure distribution on the boundary, two methods also account for the density on the boundary, namely the ghost particle and the frozen-particle method. In the **ghost particle** method, fluid particles close to the boundary are mirrored across the boundary. A ghost particle gets the same pressure, viscosity, mass and density of its corresponding fluid particle while the normal component of the velocity is inverted. This method was successfully employed to simulate different slip boundary conditions for straight [HA06] and curved [MM97] boundary surfaces. Ghost particles are generated on the fly and thus, for complex boundaries, they are hard to generate. Furthermore, ghost masses might be introduced in sharp regions. In [FG07], fixed predefined mirror particles are proposed in order to handle sharp regions.

Some authors proposed to sample boundaries with **frozen fluid particles**. The frozen to fluid particle interaction is computed in the same way as the fluid to fluid particle interaction. However, the frozen particle positions are not integrated for static boundaries, while for rigid bodies they are restricted to rigid-body motion [SSP07, KAD\*06].

The density variation on the boundary is quite smooth for ghost and frozen particles boundary methods since the pressure forces are computed in the same way as for the fluid. However, for turbulent flows, the time step has to be chosen small enough, in order to guarantee non-penetration.

None of the above-mentioned boundary handling methods control the particle position after collision directly. There-

fore, high pressure forces might be introduced if the repulsion force is too strong. In order to overcome this problem and to have more control on the boundary condition, direct-forcing approaches were applied. In [BTT09], one- and two-way coupling of rigid bodies and fluids was realized by computing control forces and velocities using a predictor-corrector scheme. This method can simulate different slip-conditions. Furthermore, non-penetration is guaranteed for each time step which prevents large (penalty) forces. Therefore, larger time steps can be used in this method. However, stacking of particles occurs in high-density and sharp-feature regions like corners. The stacking leads to irregular density distributions close to the boundary. Harada et al. also applied a distance-based control force [HKK07] that constrains the fluid particle positions to the boundary surface if formulated in a predictive-corrective way. In order to avoid stacking artifacts, a wall weight function is employed which adds a weighted contribution of the boundary to the fluid density. This value is precomputed and depends only on the distance of the fluid particle to the boundary. We experienced that the stacking of particles is reduced in this approach. However, it also suffers from irregular density distributions at the boundary which might lead to unnatural accelerations.

In this paper, we propose a new boundary scheme that incorporates density estimates at the boundary into the pressure force acting on the fluid. Similar to [BTT09], we predict and correct the particle positions. However, by taking the current pressure on the boundary into account, smoother density distributions at fluid-boundary interfaces are achieved and stacking of particles is prevented. For the proposed scheme, larger time steps can be used compared to existing approaches while parameter tweaking is not required.

### 3. Method

#### 3.1. PCISPH algorithm

In the SPH method, a quantity  $A$  at position  $\mathbf{x}$  is approximated by a smooth function. This function interpolates  $A(\mathbf{x})$  using a finite set of sampling points  $\mathbf{x}_j$  located within a distance  $h > \|\mathbf{x} - \mathbf{x}_j\|$ . It is defined as

$$A(\mathbf{x}) = \sum_j V_j A_j W(\mathbf{x} - \mathbf{x}_j, h), \quad (1)$$

where  $V_j$  is the volume represented by  $\mathbf{x}_j$  and  $W(\mathbf{x} - \mathbf{x}_j, h)$  is a kernel function with support radius  $h$ .

In order to simulate fluids, the continuum is discretized into a finite set of particles  $i$  with position  $\mathbf{x}_i$ , mass  $m_i$ , density  $\rho_i$ , pressure  $p_i$  and velocity  $\mathbf{v}_i$ . The particle positions and velocities are integrated according to internal and external forces. Internal forces are viscosity  $\mathbf{F}^v$ , surface tension  $\mathbf{F}^{st}$  and pressure forces  $\mathbf{F}^p$ . Among these, the pressure force is a very dominant force as it governs the macroscopic

---

#### Algorithm 1: PCISPH method

---

```

while animating do
    foreach particle i do
        find neighbors  $N_i(t)$ ;
    foreach particle i do
        compute forces  $\mathbf{F}_i^{0,st,ext}(t)$ ;
        set pressure  $p_i(t) = 0$ ;
        set pressure force  $\mathbf{F}_i^p(t) = (0, 0, 0)^T$ ;
     $k = 0$ ;
    while ( $\max(\rho_{err_i}^*) > \eta$  or  $k < 3$ ) do
        foreach particle i do
            predict velocity  $\mathbf{v}_i^*(t + \Delta t)$ ;
            predict position  $\mathbf{x}_i^*(t + \Delta t)$ ;
        foreach particle i do
            update distances to neighbors  $N_i(t)$ ;
            predict density  $\rho_i^*(t + \Delta t)$ ;
            predict density variation  $\delta\rho_{err_i}^*(t + \Delta t)$ ;
            update pressure  $p_i(t) += \delta p_{err_i}^*(t + \Delta t)$ ;
        foreach particle i do
            compute pressure force  $\mathbf{F}_i^p(t)$ ;
         $k += 1$ ;
    foreach particle i do
        compute new velocity  $\mathbf{v}_i(t + \Delta t)$ ;
        compute new position  $\mathbf{x}_i(t + \Delta t)$ ;

```

---

behavior of the fluid. We integrate the proposed boundary handling scheme and the adaptive time-stepping into the PCISPH method, which can handle large time steps.

The PCISPH method [SP09] predicts density fluctuations which are then corrected using pressure forces. Thereby, the pressure  $p_i(t)$  is iteratively computed for each particle such that the predicted density fluctuation  $\rho_{err_i}^*(t + \Delta t)$  is lower than a predefined maximum value  $\eta$ . In each iteration, the predicted particle position  $\mathbf{x}_i^*(t + \Delta t)$  and velocity  $\mathbf{v}_i^*(t + \Delta t)$  are estimated based on  $\mathbf{x}(t)$ ,  $\mathbf{v}(t)$  and the predicted acceleration.

Based on the predicted positions, the predicted densities  $\rho_i^*(t + \Delta t)$  are computed as

$$\rho_i^*(t + \Delta t) = \sum_j m_j W(\mathbf{x}_{ij}^*, h), \quad (2)$$

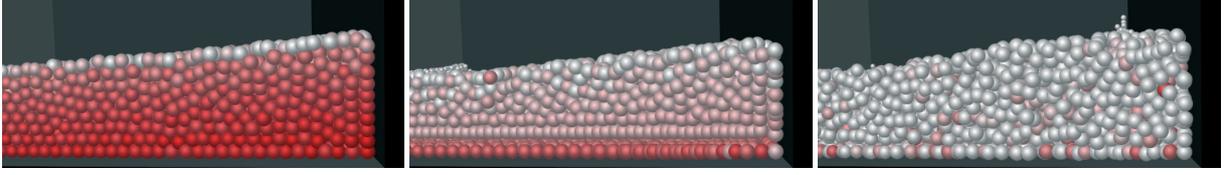
where  $\mathbf{x}_{ij}^* = \mathbf{x}_i^*(t + \Delta t) - \mathbf{x}_j^*(t + \Delta t)$ .

Subsequently, the deviation from the reference density  $\rho_{err_i}^*(t + \Delta t) = \rho_i^*(t + \Delta t) - \rho_0$  is taken to update the pressure that corrects the predicted density error

$$p_i(t) += \delta\rho_{err_i}^*(t + \Delta t), \quad (3)$$

where  $\delta$  is a precomputed value which is defined as

$$\delta = \frac{-1}{\beta(-\sum_j \nabla W_{ij}^0 \cdot \sum_j \nabla W_{ij}^0 - \sum_j (\nabla W_{ij}^0 \cdot \nabla W_{ij}^0))} \quad (4)$$



**Figure 2:** Influence of the boundary method on the pressure distribution. Particles are colored according to the current pressure where red is maximum and white is minimum. The position correction of direct-forcing approaches might lead to stacking of particles (middle [BTT09]) and high density ratios (middle [BTT09] and right [HKK07]). By additionally taking the pressure on the boundary into account, a smooth distribution is achieved (left, proposed method). All boundary methods are integrated into PCISPH.

and

$$\beta = 2 \left( \frac{m_i \cdot \Delta t}{\rho_0} \right)^2, \quad (5)$$

where  $m_i$  is assumed to be same for all particles. In (4),  $W_{ij}^0 = W(\mathbf{x}_{ij}^0)$  with  $\mathbf{x}_i^0$  denoting the initial position of a prototype particle with a filled neighborhood  $j$ .

Finally, the pressure force

$$\mathbf{F}_i^p(t) = -m_i \sum_j m_j \left( \frac{p_i(t)}{(\rho_i^*)^2} + \frac{p_j(t)}{(\rho_j^*)^2} \right) \nabla W(\mathbf{x}_{ij}, h) \quad (6)$$

is used to recompute the predicted positions and velocities for all particles. This procedure is repeated until all predicted particle variations are smaller than a user-defined threshold value  $\eta$ . Solenthaler and Pajarola suggest a minimum of three iterations in any loop in order to limit temporal fluctuations in the pressure field. The PCISPH method is illustrated in Alg. 1.

Note that in this paper  $\eta$  is used as an absolute value and not as the percentage  $\eta\%$ . Thus, for a density fluctuation of 2%,  $\eta = 2 \frac{\rho_0}{100}$ .

### 3.2. Boundary handling

External forces such as boundary forces  $\mathbf{F}^b$  influence the density distribution of the fluid. High pressure forces have to be employed to counteract the compression when, e. g. a fast moving fluid hits a static obstacle. The fluid-rigid interaction, therefore, challenges the simulation method for incompressible fluids. In this section, we present a new boundary method that combines the idea of direct-forcing [BTT09] with the pressure-based frozen-particles method. The proposed boundary method enforces non-penetration of rigid objects even for large time steps. By incorporating density estimates at the boundary into the pressure force, unnatural accelerations resulting from high pressure ratios are avoided.

Similar to [BTT09], we sample the boundaries (walls and rigid-bodies) with non-moving fluid particles which we call boundary particles. Each boundary particle  $b$  stores its position  $\mathbf{x}_b$ , normalized normal  $\mathbf{n}_b$ , mass  $m_b$ , density  $\rho_b$  and

pressure  $p_b$ . In the following, we assume that the spacing of the boundary particle is equal to the equilibrium distance  $r_0 = 0.5 \cdot h$  of the fluid particles. Thus, a fluid particle  $i$  is considered to penetrate the boundary at position  $\mathbf{x}_b$ , if  $\|\mathbf{x}_i - \mathbf{x}_b\| < r_0$ .

In [BTT09], the penetration is predicted after the fluid and rigid body forces are computed. For penetrations of non-moving rigid bodies, the position and velocity of  $i$  is updated as

$$\mathbf{v}_i(t + \Delta t) = \varepsilon [\mathbf{v}_i^*(t + \Delta t)]_t - \delta [\mathbf{v}_i(t)]_n \quad (7)$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i^*(t + \Delta t) + \|(\mathbf{x}_i^*(t + \Delta t) - \mathbf{x}_b)\| \cdot \mathbf{n}_b \quad (8)$$

where  $[\mathbf{v}_i(t)]_n = (\mathbf{v}_i(t) \cdot \mathbf{n}_b) \cdot \mathbf{n}_b$  denotes the normal velocity and  $[\mathbf{v}_i^*(t + \Delta t)]_t = \mathbf{v}_i^*(t + \Delta t) - [\mathbf{v}_i^*(t + \Delta t)]_n$  the tangential velocity.  $\varepsilon, \delta \in [0, 1]$  are controlling the friction and the elasticity of the collision.

According to the boundary sampling, it is possible that a fluid particle  $i$  penetrates more than one boundary particle at the same time. Thus, for concave objects with sharp features, iteratively correcting the positions might lead to time-inconsistent corrections of the particle positions. We propose to estimate the penetration depth and direction by employing a weighting function to compute a time-consistent correction force  $\mathbf{F}_i^b$  in the sense of [HTK\*04].

We average the boundary normals  $\mathbf{n}_i^c$  of all boundary particles  $b$  which are penetrated by particle  $i$  as

$$\mathbf{n}_i^c = \sum_b w_{ib}^c \mathbf{n}_b \quad (9)$$

$$w_{ib}^c = \max \left( 0, \frac{r_0 - \|\mathbf{x}_{ib}^*\|}{r_0} \right) \quad (10)$$

where  $\|\mathbf{x}_{ib}^*\| = \|(\mathbf{x}_i^*(t + \Delta t) - \mathbf{x}_b)\|$ . The position of the particle  $i$  is then corrected according to  $\mathbf{n}_i^c$  with

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i^*(t + \Delta t) + \frac{1}{\sum_b w_{ib}^c} \sum_b w_{ib}^c (r_0 - \|\mathbf{x}_{ib}^*\|) \frac{\mathbf{n}_i^c}{\|\mathbf{n}_i^c\|} \quad (11)$$

while the resulting velocity is computed as

$$\mathbf{v}_i(t + \Delta t) = \varepsilon [\mathbf{v}_i^*(t + \Delta t)]_t. \quad (12)$$

As stated in [BTT09], the position update leads to higher density ratios in the fluid at the boundary interface (see Fig. 2). Thus, smaller time steps might be required in order to enforce incompressibility. Furthermore, stacking of particles might occur in boundary regions since the normal velocities are fully damped for  $\delta = 0$  in (7). In order to prevent stacking, the coefficient of restitution  $\delta$  might be chosen larger than zero. However, this can lead to rigid-body-like collisions for the fluid particles. In the proposed model, these drawbacks of the direct-forcing approach are eliminated by taking the density contribution of the boundary particles and, hence, the pressure at the boundary into account.

In the presented method, the densities of fluid and boundary particles  $i$  are updated in each prediction step as

$$\rho_i^*(t + \Delta t) = \sum_j m_j W(\mathbf{x}_{ij}^*, h) + \sum_b m_b W(\mathbf{x}_{ib}^*, h). \quad (13)$$

where we use the same support radius  $h$  for fluid and boundary particles. Thus, the different particle sets interact in the usual way. Since the pressure of the boundary particles increases with the surrounding fluid density, it counteracts high density fluctuations. Therefore, compressions can be rapidly resolved while stacking of particles is avoided.

The proposed approach can be applied to any SPH algorithm. We employed it for the PCISPH method since it allows for large time steps (see Alg. 2). In contrast to [SP09], we suggest to predict a non-penetrating fluid particle position  $\mathbf{x}_i^*$  before computing the density fluctuation  $\rho_{err,i}^*$ . Therefore, the predicted density fluctuation is more precise. Consequently, the compression is quickly resolved. As we show in Sec. 4, only the combination of the direct-forcing method with the density based pressure of the boundary particles yields smooth density distributions.

In this section, we have described a boundary handling method that exerts distance- and density-based forces on the fluid. In combination with the PCISPH method, high density ratios at the boundaries are rapidly resolved. Therefore, larger time steps can be used in comparison to existing methods as shown in Sec. 4. In the next section, we propose how the overall computation time can be further reduced by using adaptive time-stepping.

### 3.3. Adaptive time-stepping

For numerical stability and convergence, several time step constraints must be satisfied. The Courant-Friedrich-Levy (CFL) condition for SPH

$$\Delta t \leq \lambda_v \left( \frac{h}{v^{max}} \right) \quad (14)$$

states that the speed of numerical propagation must be higher than the speed of physical propagation, where  $v^{max} = \max_{\forall t, \forall i} (\|\mathbf{v}_i(t)\|)$  is the maximum magnitude of the velocity throughout the simulation.  $\lambda_v$  is a constant factor, e.g.  $\lambda_v = 0.4$  in [Mon92]. In other words, a particle  $i$  must not

---

**Algorithm 2:** PCISPH with the proposed boundary handling and adaptive time-stepping.

---

```

while animating do
    foreach particle i,b do
        find neighbors  $N_{i,b}(t)$ ;
    foreach particle i do
        compute forces  $\mathbf{F}_i^{0,st,g}(t)$ ;
        set pressure  $p_i(t) = 0$ ;
        set pressure force  $\mathbf{F}_i^p(t) = (0, 0, 0)^T$ ;
    foreach particle b do
        set pressure  $p_b(t) = 0$ ;
     $k = 0$ ;
    while  $k < 3$  do
        foreach particle i do
            predict velocity  $\mathbf{v}_i^*(t + \Delta t)$ ;
            predict position  $\mathbf{x}_i^*(t + \Delta t)$ ;
            predict world collision  $\mathbf{x}_i^*(t + \Delta t)$ ;
        foreach particle i,b do
            update distances to neighbors  $N_{i,b}(t)$ ;
            predict density  $\rho_{i,b}^*(t + \Delta t)$ ;
            predict density variation  $\rho_{err,i,b}^*(t + \Delta t)$ ;
            update pressure  $p_{i,b}(t) + = \delta \rho_{err,i,b}^*(t + \Delta t)$ ;
        foreach particle i do
            compute pressure force  $\mathbf{F}_i^p(t)$ ;
         $k + = 1$ ;
    foreach particle i do
        compute new velocity  $\mathbf{v}_i(t + \Delta t)$ ;
        compute new position  $\mathbf{x}_i(t + \Delta t)$ ;
        compute world collision (11) and (12);
    adapt time step  $\Delta t$ ;
    recompute  $\beta$  (5) and  $\delta$  (4);

```

---

move more than its smoothing length  $h$  in one time step. Furthermore, high accelerations might influence the simulation results negatively. Therefore, the time step must also satisfy

$$\Delta t \leq \lambda_f \left( \sqrt{\frac{h}{F^{max}}} \right) \quad (15)$$

where  $F^{max} = \max_{\forall t, \forall i} (\|\mathbf{F}_i(t)\|)$  denotes the magnitude of the maximum force per unit mass for all particles throughout the simulation. In [Mon92],  $\lambda_f = 0.25$  is suggested.

As stated in [SP09], the force term (15) dominates the time step for the PCISPH method. However, the maximum force  $F^{max}$  of all fluid particles is not known a priori, but must be estimated according to the initial setting of the scene, e.g. [BT07]. Unfortunately, nearly every change in the scene setting (for example the density fluctuation or the number of particles) requires resetting of the time step. For complex scenarios with moving and non-moving objects, it can be tedious to find an appropriate time step. Furthermore, the maximum velocity and pressure can heavily

vary throughout the simulation. In constant time stepping schemes, the minimum time step is used over the whole simulation time, even though the time step might be too restrictive for the main part of the simulation. In this section, we propose an adaptive time-stepping scheme for PCISPH, which smoothly adapts to the current simulation state. The proposed method does not require any manual setting of the time step.

The adaptive time integration discussed in [DC96] works only for WCSPH algorithms, e. g. [MCG03], but not for PCISPH. For PCISPH, computing the time step for each simulation step using (14) and (15) directly, causes spurious density shocks in the fluid if the instantaneous time step variation is not smooth. In PCISPH, the time step directly influences the pressure correction term where a smaller time step results in larger pressures (see (4) and (5)). Thus, large time step variations change the pressure field which changes the convergence of the simulation. In this case, the PCISPH method is not able to solve the compression, i. e. the prediction-correction loop does not converge. However, our observations suggest that those problems are completely prevented if the change in time step is not larger than 0.2% from one simulation step to the next.

The proposed method varies the time step  $\Delta t$  for  $n$  particles according to the current (overall) volume compression  $\rho_{err}^{avg} = \frac{1}{n} \sum_i \rho_{err_i}(t)$ , the maximum density fluctuation  $\rho_{err}^{max} = \max_{v_i}(\rho_{err_i}(t))$ , the maximum velocity  $v_i^{max} = \max_{v_i}(\|\mathbf{v}_i(t)\|)$  and the maximum force  $F_i^{max} = \max_{v_i}(\|\mathbf{F}_i(t)\|)$ . These values are related with the user-defined maximum volume compression  $\eta_{avg}$  and the maximum allowed density fluctuation  $\eta_{max}$ .

We observed that the difference of the maximum density fluctuation and the overall fluid compression  $\rho_{err}^{max} - \rho_{err}^{avg}$  is small for the main part of the simulation. However, for high impacts, this value might become more than ten times bigger without affecting the overall compression. Thus, we suggest to choose  $\eta_{max} = 10 \cdot \eta_{avg}$ .

We underestimate the initial time step as  $\Delta t = 0.25 \left( \frac{h}{v_0^{max}} \right)$ . During the simulation, the time step is increased by 0.2%, if and only if each of the following criteria is satisfied:

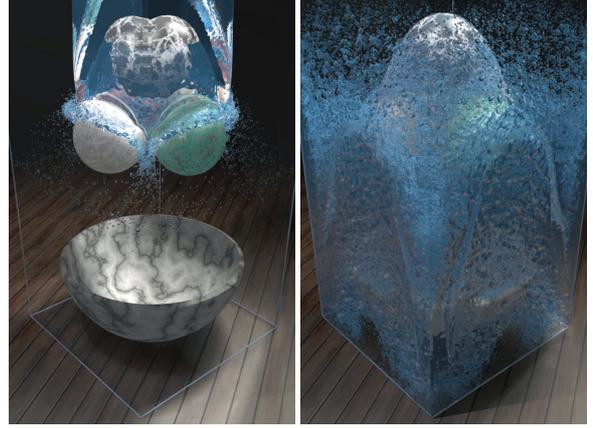
$$0.19 \left( \sqrt{\frac{h}{F_i^{max}}} \right) > \Delta t \quad (16)$$

$$\rho_{err}^{max} < 4.5 \cdot \eta_{avg} \quad (17)$$

$$\rho_{err}^{avg} < 0.9 \cdot \eta_{avg} \quad (18)$$

$$0.39 \left( \frac{h}{v_i^{max}} \right) > \Delta t \quad (19)$$

On the other hand, the time step is decreased by 0.2%, if one



**Figure 3:** Shock scene. High impact velocities might cause shocks for large time steps. The proposed adaptive time-stepping method handles such shocks in a predictive-corrective manner. The time step evolution is shown in Fig. 4.

of the following holds:

$$0.2 \left( \sqrt{\frac{h}{F_i^{max}}} \right) < \Delta t \quad (20)$$

$$\rho_{err}^{max} > 5.5 \cdot \eta_{avg} \quad (21)$$

$$\rho_{err}^{avg} \geq \eta_{avg} \quad (22)$$

$$0.4 \left( \frac{h}{v_i^{max}} \right) \leq \Delta t \quad (23)$$

The constants in (16) to (23) have been chosen empirically. Generally, these criteria underestimate the required time step according to (14), (15) and the density fluctuation. They can be relaxed and restricted by changing the constant parameters.

The proposed method adjusts the time step such that for the PCISPH method not more than three iterations are needed to resolve the volume compression. In contrast to [SP09], we therefore limit the number of iterations to three. Still, the overall volume compression is enforced according to (18), (22).

However, for simulations with large shocks, e. g. the corner breaking dam, the maximum density fluctuation might increase dramatically in one time step. In these scenarios, decrementing the time step by only 0.2% is not sufficient to resolve the density fluctuation. Therefore, we propose to handle such shocks separately.

**Shock handling.** The proposed adaptive time-stepping method varies the time step smoothly, if the density fluctuation, maximum force and velocity do not exceed predefined values. However, for high impact velocities, the time step might be too large. In order to handle those shocks, we go two simulation steps backwards and resume the sim-

ulation using an appropriate time step  $\Delta t_{new}$ . In case of a detected shock, the required time step is estimated with respect to the maximum force and velocity computed for the shock/impact:

$$\Delta t_{new} = \min \left( 0.2 \sqrt{\frac{h}{F_t^{max}}}, 0.25 \frac{h}{v_t^{max}} \right). \quad (24)$$

A shock is detected, if one of the following criteria is met:

$$\rho_{err}^{max}(t) - \rho_{err}^{max}(t - \Delta t) > \delta_{shock} \quad (25)$$

$$\rho_{err}^{max}(t) > \eta_{max} \quad (26)$$

$$0.45 \left( \frac{h}{v_t^{max}} \right) < \Delta t \quad (27)$$

with  $\eta_{avg} < \delta_{shock} < \eta_{max}$ . The method handles shocks in a predictive-corrective manner. Since a shock is detected, if  $\rho_{err}^{max}$  exceeds  $\eta_{max}$ , any user-defined maximum density fluctuation can be guaranteed.

Note that (24) varies the time step by more than 0.02. Thus, the pressure field is instantaneously changed. However, we prevent shocks in the subsequent simulation steps by explicitly underestimating the new, required time step. Thereby, the overall simulation result is not visibly changed (see accompanying video).

Our time stepping scheme automatically computes the required time step which enforces a predefined volume compression  $\eta_{avg}$  and guarantees a maximum density fluctuation  $\eta_{max}$ . The proposed scheme reduces the overall computation time without affecting the simulation results. Generally, the state of the fluid evolves very smoothly, therefore increasing and decreasing the time step by 0.2% is sufficient for most of the time. Shocks occur rarely and can be resolved instantly by the proposed scheme. Therefore, the shock handling does not affect the simulation results.

### 3.4. Implementation

In the presented method, walls and rigid bodies are represented by boundary particles. In order to represent the shape of the rigid bodies precisely, we require a sampling density equal to the rest distance of the fluid. For achieving a close to uniform sampling, we apply an isotropic, feature-sensitive remeshing algorithm to a two-manifold boundary object with borders [BK04, BPR\*06]. Afterwards, the boundary particles are placed at the vertices of the resultant mesh.

As the number of boundary particles influences the computational amount, we only take active boundary particles into account. Boundary particles are activated in the neighborhood search, for which we use the grid-based spatial hashing method [THM\*03]. As the fluid is only in contact with a small percentage of all boundary particles at once, the effect of boundary particles on the computation time is small.

In our SPH implementation, the viscosity and the surface tension forces are computed as described in [BT07]. In all SPH equations, we use the cubic spline kernel explained in [Mon92]. Furthermore, we use the constant correction technique as described in [BK02] in order to correct the density at the fluid surface. Positions and velocities are updated using the Euler-Cromer scheme.

For the videos, the fluid surface is reconstructed as proposed in [SSP07] and rendered using POV-Ray. Our simulation software is parallelized with OpenMP [Ope05].

## 4. Results

In this section, we show the capabilities of our approach. First, we show the effect of the boundary method on the density distribution and the required time step. Then, we discuss our adaptive time-stepping scheme with respect to performance, stability and influence on the simulation. Finally, our method is applied to complex scenarios. For these scenes, performance measurements and simulation data are summarized. All timings are given for an Intel Xeon 7460 with 24 2.66 GHz CPUs. In the given scenarios, we use either 8 or 24 CPUs.

### 4.1. Boundary method

In order to show the capabilities of our boundary scheme, we compared it with the direct-forcing methods [BTT09] and [HKK07]. The methods were tested for a simple corner breaking dam scene with 23k particles. Both reference boundary methods were integrated into the proposed PCISPH algorithm (see Alg. 2) without adapting the time step.

We compared the different schemes with respect to the influence on the density distribution and the time step. For all methods, we tried to find the maximum time step which satisfied an overall volume compression  $\eta_{avg}^{\%}$  smaller than 1%. Since for the PCISPH method, these values are also influenced by the number of iterations required to correct the density error, we used three iterations in all simulation steps. For all methods, we used a free-slip condition, i.e. the tangential velocity was not damped.

In Fig. 2, a side-by-side comparison of the tested methods is given. We colored the particles according to the pressure distribution. The coloring scheme assigns the color red to the highest pressure and white to the lowest pressure in the current time step. These values are interpolated according to  $\sqrt{\frac{p_i(t)}{p^{max}(t)}}$ . Note that the maximum pressure and hence the color schemes are not normalized, i.e. for different methods, the same color might refer to different pressures.

As can be observed, the pressure distribution differs a lot for the three boundary methods. In [BTT09], the normal velocity of colliding fluid particles is fully damped. In the test scenario, the fluid volume covers the floor completely

scene	# p	CPUs	$t_{real}$	$t_{sim}$ adaptive	$t_{sim}$ constant	speed up	$\max(\rho_{err}^{avg}(t))$	$\frac{\#p_{boundary}}{\#p_{fluid}}$
Glass	up to 75K	24	10s	4min ( $\Delta t_{min} = 0.0015, \Delta t_{max} = 0.0028$ )	6min $\Delta t = 0.0018$	1.5	0.65%	31.4%
CBD small	120K	8	20s	29min ( $\Delta t_{min} = 0.00114, \Delta t_{max} = 0.00388$ )	1h 43min $\Delta t = 0.00116$	3.55	0.95%	14%
Shock	1.3M	24	20s	11h 25min ( $\Delta t_{min} = 0.00011, \Delta t_{max} = 0.00311$ )	166h 38min $\Delta t = 0.00012$	14.50	0.70%	9.1%
CBD large	1.7M	24	20s	16h 30min ( $\Delta t_{min} = 0.00017, \Delta t_{max} = 0.00166$ )	85h 42min $\Delta t = 0.00018$	5.20	1.30%	5.4%
Wave	1.8m	24	20s	19h 41min ( $\Delta t_{min} = 0.00051, \Delta t_{max} = 0.00256$ )	29h 43min $\Delta t = 0.00058$	1.51	0.50%	6.5%

**Table 1:** Comparison of constant time step and adaptive time step performance. The last column lists the average ratio of the number of active boundary particles to the number of fluid particles.

and therefore, particles on the floor get stuck. This results in a high pressure ratio of the particles 'colliding' with the boundary and their 'non-colliding' neighbors (on top). Due to high pressures of the colliding particles, a spatial gap between the particles arises (see Fig. 2, middle). Furthermore, the high pressure ratio causes unnatural acceleration.

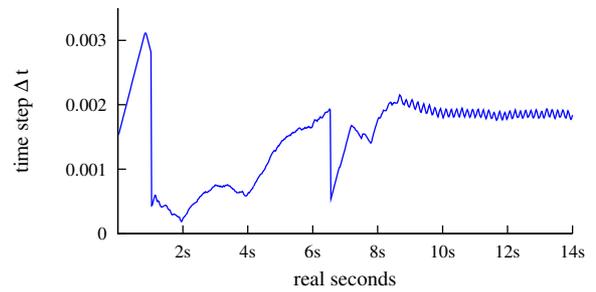
In [HKK07], the boundary contributes to the density of the fluid particles. However, the contribution is only related to the distance and not to the current pressure on the boundary. This results in a noisy pressure distribution with local high pressure ratios (see Fig. 2, right).

We observed, that large pressure ratios might result in strong pressure force and hence high acceleration. In contrast to [HKK07, BTT09], the proposed method computes the pressure on the boundary particles like for fluid particles. Therefore, particles do not get stuck on the boundary, while the pressure distribution is very smooth (see Fig. 2, left). Thus, our method can generally handle larger time steps compared to [HKK07, BTT09]. For the test scene, the following time steps could be used:  $\Delta t = 0.0008$  [BTT09],  $\Delta t = 0.0015$  [HKK07] and  $\Delta t = 0.0045$  for the proposed boundary method.

Note that in penalty-based boundary methods, large forces and, thus, small time steps are required to resolve the penetration into the boundary. Since the proposed method enforces non-penetration by directly correcting the positions, much larger time-steps can be used.

#### 4.2. Adaptive time-stepping

Our adaptive time-stepping method requires no manual setting of the time step, since it increases and decreases according to the current state of the simulation. Adaptively changing the time step might, however, change the simulation results. In order to show that our scheme does not affect the simulation results, we computed a corner breaking dam with 120k particles (*CBD small*) using constant and adaptive time-stepping. For the adaptive time-stepping,  $v^{max}$  and  $F^{max}$  is tracked. These values are used to compute



**Figure 4:** Time step evolution for the Shock scene. Two shocks have occurred around 1s and 6s.

the required constant time step according to (14) and (15). Note that for adaptive time-stepping, the time step is underestimated and thus,  $\Delta t_{min}$  is smaller than the constant time step. Both scenes were computed with the proposed boundary handling method.

As is shown in the accompanying video, the visual result of the adaptive time-stepping is in good agreement with the constant time-stepping. In this scene, the overall computation time  $t_{sim}$  to compute 20 real seconds  $t_{real}$  of simulation was 1 hour and 43 minutes with constant time-stepping and 29 minutes with our adaptive time-stepping method. Thus, for this scene the overall speed up is 3.55 (see Table 1).

In order to demonstrate the shock handling, we designed a scene with high impact velocities. A fluid volume with 1.3 million particles is dropped from a large height. The fluid hits some spheres with high impact velocities (see Fig. 3). In this case, a shock is detected and starting from two simulation step backwards, the simulation is resumed using an appropriate time step. Fig. 4 shows the evolution of the time step for this scenario. According to the pessimistic estimation of the required time step, subsequent shocks are prevented.

For scenarios with high impact velocities, like the *Shock*



**Figure 5:** Glass scene. Filling up a glass with up to 75K fluid particles. The whole glass consists of 39K boundary particles. Overall computation time 4 minutes,  $\Delta t_{min} = 0.0015$ ,  $\Delta t_{max} = 0.0022$ .

scene, our adaptive time-stepping method reduces the overall computation time significantly. For other scenes, like the *Wave* scene, the speed up might be rather small. However, in every case, it adapts to the current state of the simulation and prevents numerical instabilities.

#### 4.3. Performance and application

We applied the proposed algorithm (see Alg. 2) to a scene where up to 75K particles fill a concave glass (see Fig. 5). In this scene, the fluid interacts realistically with the static object while comparatively large time steps could be used.

Furthermore, we simulated a corner breaking dam with 1.7 million particles (see Fig. 6). According to the large number of particles, small details are well preserved. Due to the turbulent flow, a speed up of 5.2 is achieved with adaptive-time stepping.

The proposed boundary handling algorithm is designed for one-way rigid-fluid coupling with static objects. In order to show that the algorithm might also work for moving rigid bodies, we simulated a wave generator with a moving wall. In the *Wave* scene (see Fig. 1), the wave generator moves in one dimension according to a cosine function with peak velocity  $\mathbf{v}_w^{max} = (1, 0, 0)^T$ . Hence, the 1.8 million fluid particles forming a gentle wave that runs up a beach. Since the moving boundary did not cause any problems to the simulation, we believe that the proposed algorithm might be extended for two-way coupling of rigid bodies with fluid and deformable solids with fluid.

#### 5. Conclusion

We proposed a new boundary handling scheme which combines the advantages of the direct-forcing approach with the frozen-particle method. The combination eliminates the drawbacks of both methods. Applied to the PCISPH method, high density ratios at the boundary are rapidly resolved.



**Figure 6:** CBD large scene. A corner breaking dam with 1.7 million particles. Overall computation time 16.5 hours for 20 real seconds,  $\Delta t_{min} = 1.7 \cdot 10^{-4}$ ,  $\Delta t_{max} = 1.66 \cdot 10^{-3}$ .

Smooth density and pressure distributions are enforced and, therefore, larger time steps can be used.

Furthermore, we suggested an adaptive time-stepping method for the PCISPH method. This method increases and decreases the required time step according to the state of the simulation. While the adaptive time-stepping method reduces the overall computation time for turbulent fluids, the visual result of the simulation is in good agreement with the constant time-stepping. Moreover, our method does not require any initial time step decisions, since the time step is computed according to the current state of the fluid.

Our system reduces the overall computation time for the PCISPH method, since comparatively large time steps can be used.

## Acknowledgements

This project is supported by the German Research Foundation (DFG) under contract number TE 632/1-1. We thank the reviewers for their helpful comments. We also thank Gizem Akinci for rendering the scenes.

## References

- [APKG07] ADAMS B., PAULY M., KEISER R., GUIBAS L.: Adaptively sampled particle fluids. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM Press, p. 48.
- [BK02] BONET J., KULASEGARAM S.: A simplified approach to enhance the performance of smooth particle hydrodynamics methods. *Applied Mathematics and Computation* 126, 2-3 (2002), 133–155.
- [BK04] BOTSCH M., KOBBELT L.: A remeshing approach to multiresolution modeling. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (New York, NY, USA, 2004), ACM, pp. 185–192.
- [BPR\*06] BOTSCH M., PAULY M., ROSSL C., BISCHOFF S., KOBBELT L.: Geometric modeling based on triangle meshes. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), ACM, p. 1.
- [BT07] BECKER M., TESCHNER M.: Weakly compressible SPH for free surface flows. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, 2007), Eurographics Association, pp. 209–217.
- [BTT09] BECKER M., TESSENDORF H., TESCHNER M.: Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics* 15, 3 (2009), 493–503.
- [CR99] CUMMINS S., RUDMAN M.: An SPH Projection Method. *Journal of Computational Physics* 152, 2 (1999), 584–607.
- [DC96] DESBRUN M., CANI M.-P.: Smoothed Particles: A new paradigm for animating highly deformable bodies. In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)* (1996), Springer-Verlag, pp. 61–76. Published under the name Marie-Paule Gascuel.
- [EFFM02] ENRIGHT D., FEDKIW R., FERZIGER J., MITCHELL I.: A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 183, 1 (2002), 83–116.
- [FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 23–30.
- [FG07] FALAPPI S., GALLATI M.: SPH Simulation of water waves generated by granular landslides. In *Proc. of 32nd Congress of IAHR (International Association of Hydraulic Engineering & Research)* (2007).
- [HA06] HU X., ADAMS N.: A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics* 213, 2 (2006), 844–861.
- [HA07] HU X., ADAMS N.: An incompressible multi-phase SPH method. *Journal of Comp. Phys.* 227, 1 (2007), 264–278.
- [HKK07] HARADA T., KOSHIZUKA S., KAWAGUCHI Y.: Smoothed Particle Hydrodynamics on GPUs. In *Proc. of Computer Graphics International* (2007), pp. 63–70.
- [HTK\*04] HEIDELBERGER B., TESCHNER M., KEISER R., MUELLER M., GROSS M.: Consistent Penetration Depth Estimation for Deformable Collision Response. In *Proc. Vision, Modeling, Visualization* (2004), pp. 339–346.
- [KAD\*06] KEISER R., ADAMS B., DUTRÉ P., GUIBAS L., PAULY M.: *Multiresolution Particle-Based Fluids*. Tech. rep., ETH Zurich, 2006.
- [LKO05] LIU J., KOSHIZUKA S., OKA Y.: A hybrid particle-mesh method for viscous, incompressible, multiphase flows. *J. Comput. Phys.* 202, 1 (2005), 65–93.
- [LTKF08] LOSASSO F., TALTON J., KWATRA N., FEDKIW R.: Two-Way Coupled SPH and Particle Level Set Fluid Simulation. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 797–804.
- [MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 154–159.
- [MK09] MONAGHAN J., KAJTAR J.: SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications* 180, 10 (2009), 1811–1820.
- [MM97] MORRIS J., MONAGHAN J.: A Switch to Reduce SPH Viscosity. *Journal of Comp. Physics* 136, 1 (1997), 41–50.
- [MMS04] MIHALEF V., METAXAS D., SUSSMAN M.: Animation and control of breaking waves. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 315–324.
- [Mon92] MONAGHAN J.: Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.* 30 (1992), 543–574.
- [Mon94] MONAGHAN J.: Simulating free surface flows with SPH. *Journal of Computational Physics* 110, 2 (1994), 399–406.
- [Mon05] MONAGHAN J.: Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68, 8 (2005), 1703–1759.
- [MST\*04] MÜLLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds* 15, 34 (2004), 159–171.
- [Ope05] OPENMP ARCHITECTURE REVIEW BOARD: *OpenMP Application Program Interface, Version 2.5*, May 2005.
- [SBH09] SIN F., BARGTEIL A. W., HODGINS J. K.: A point-based method for animating incompressible flow. In *SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), ACM, pp. 247–255.
- [SP09] SOLENTHALER B., PAJAROLA R.: Predictive-corrective incompressible sph. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), ACM, pp. 1–6.
- [SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid-solid interactions. *Computer Animation and Virtual Worlds* 18, 1 (2007), 69–82.
- [THM\*03] TESCHNER M., HEIDELBERGER B., MÜLLER M., POMERANETS D., GROSS M.: Optimized Spatial Hashing for Collision Detection of Deformable Objects. In *Proc. of Vision, Modeling, Visualization (VMV)* (2003), pp. 47–54.
- [TKPR06] THÜREY N., KEISER R., PAULY M., RÜDE U.: Detail-preserving fluid control. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 7–13.