Direct Forcing for Lagrangian Rigid-Fluid Coupling

Markus Becker, Hendrik Tessendorf, and Matthias Teschner

Abstract—We propose a novel boundary handling algorithm for particle-based fluids. Based on a predictor-corrector scheme for both velocity and position, one- and two-way coupling with rigid bodies can be realized. The proposed algorithm offers significant improvements over existing penalty-based approaches. Different slip conditions can be realized and nonpenetration is enforced. Direct forcing is employed to meet the desired boundary conditions and to ensure valid states after each simulation step. We have performed various experiments in 2D and 3D. They illustrate one- and two-way coupling of rigid bodies and fluids, the effects of hydrostatic and dynamic forces on a rigid body, as well as different slip conditions. Numerical experiments and performance measurements are provided.

Index Terms—Physically-based simulation, fluid dynamics, smoothed particle hydrodynamics, rigid bodies, boundary handling.

1 Introduction

The simulation of fluids has attracted increasing attention in computer graphics in recent years. Various sophisticated methods have been proposed and a thorough introduction of fluid simulation techniques has been presented by Bridson and Müller-Fischer in their ACM SIGGRAPH '07 course notes [1].

As a fluid is generally simulated in a domain with fixed and moving obstacles, it is necessary to consider the interaction of the fluid with these obstacles. Often, different kinds of boundary conditions need to be incorporated. While this problem has been dealt with extensively in the context of grid-based methods, there are still only a few approaches to boundary conditions for particle-based methods such as Smoothed Particle Hydrodynamics (SPH). This is to some extent due to the fact that one is interested in preserving the local nature given in many Lagrangian fluid simulations. Despite these challenges, the interest in boundary conditions for particle-based fluids is motivated by the usefulness of Lagrangian fluids for irregular domains. Typically applied penalty methods as, e.g., provided by Monaghan [2] offer only limited control of the boundary handling. To enforce nonpenetration, large penalty forces have to be applied, which introduce stiffness to the equations.

We propose a novel method for the two-way coupling of compressible Lagrangian fluids and rigid objects. Control forces are incorporated in the discretized momentum equations in order to obtain specific relative velocities at a boundary in each time step. This is known as direct forcing. It is realized in a predictor-corrector fashion. Using the proposed formulation, a large range of slip and Neumann boundary conditions can be imposed for arbitrarily shaped, fixed, or moving boundaries. Regularly and irregularly shaped boundaries can be handled in a unified manner. Dynamic and hydrostatic fluid forces acting on the boundaries are considered. The local nature of employed SPH method is preserved by the boundary handling. Many drawbacks of penalty-based methods such as oscillations at the boundary and limited control of the boundary conditions can be avoided.

The proposed technique extends previous Lagrangian boundary approaches such as the one of Hieber [3]. In contrast to [3], it avoids force interpolations and it guarantees nonpenetration for fixed and moving rigid boundaries. To enforce the boundary velocities, we adopt the interpenetration resolution proposed in [4] and [5]. Fluid leaking through boundaries is avoided by controlling particle velocities and positions in separate substeps. A minimal parameter set with a known parameter range allows for an intuitive setup of the simulation.

Experiments are performed using the corrected SPH algorithm of Bonet and Kulasegaram [6] and the weakly compressible pressure formulation described, e.g., in [7] and [8]. Fig. 1 illustrates a first example. In this stone-skipping simulation, the reflection of the stone from the fluid surface is realized using the proposed boundary handling.

The remainder of this paper is organized as follows: After discussing related approaches in Section 2, we briefly describe the employed fluid model and the particle-based representation of the rigid bodies in Sections 3 and 4, respectively. The boundary handling approach for the two-way coupling of Lagrangian fluids and rigid objects is presented in Section 5. Parameters, implementation issues, and limitations are discussed. In Section 6, various experiments are described to illustrate the capabilities of the proposed approach. The experiments cover the major features of the boundary handling approach. They include

• *The authors are with the Computer Graphics Group, Computer Science Department, Albert-Ludwigs-University Freiburg, Georges-Koehler-Allee 052, D-79110 Freiburg im Breisgau, Germany.*
E-mail {mbecker, tessendo, teschner}@informatik.uni-freiburg.de.

Manuscript received 2 May 2008; revised 30 Aug. 2008; accepted 4 Sept. 2008; published online 1 Oct. 2008.
Recommended for acceptance by J. Stam.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2008-05-0062.
Digital Object Identifier no. 10.1109/TVCG.2008.107.

Fig. 1. Three frames of a stone-skipping simulation. The stone is reflected from the surface in case of an impact. This effect is obtained using a novel two-way coupling approach of rigid bodies and particle-based fluids.

a comparison with the penalty approach in [2], performance measurements, the effects of hydrostatic and dynamic forces, as well as one and two-way coupling.

2 Related Work

In this section, we discuss some related literature concerning boundary handling and rigid-fluid coupling for different fluid simulations. The related work covers Eulerian boundary handling approaches, mixed formulations, and Lagrangian approaches.

Many sophisticated solutions have been proposed for the solid-fluid coupling of Eulerian fluids and early coupling approaches date back to, e.g., Chen and Lobo [9]. The authors introduce two types of one-way coupling for a 2D Navier-Stokes solver where the third dimension is modeled using a height field. An idea for the two-way coupling is outlined but not implemented. Various authors have realized a one-way solid-fluid coupling and fixed boundaries in 3D by voxelizing the boundaries on the fluid grid [10], [11], [12], [13]. These approaches commonly adjust the fluid velocity of grid points covered by the solid to the velocity of the solid. Several improvements have been proposed, e.g., a corrected normal for free tangential slip [14]. Still, this method tends to produce stair-step artifacts for boundaries that are not aligned with the grid. This is especially noticeable in the case of coarse grids. Takahashi et al. [15] provide a simple two-way coupling for voxelized buoyant rigid bodies. They take the pressure on the surface into account. However, dynamic forces, i.e., forces due to relative velocities in the fluid, are neglected. Yngve et al. [16] provide an approach to the two-way coupling of deformable and fracturing solids with a compressible fluid. Still, solids need to be voxelized on the grid in order to realize a solid-to-fluid coupling.

To avoid voxelization artifacts, some authors have proposed to adaptively align the fluid grid at boundaries or to use remeshing. An octree refinement is introduced by Losasso et al. [17]. Irving et al. [18] introduce a regular one-layer refinement for MAC grids with moving objects. Klingner et al. [19] and Chentanez et al. [20] address irregular geometries by using tetrahedral meshes for the fluid simulation. The mesh is regenerated in each time step according to the current configuration of the boundaries. The work of Klingner et al. [19] is extended by Chentanez et al. [21] to handle both rigid and deformable solids. Feldman et al. [22] handle moving boundary conditions by deforming the underlying simulation mesh.

There exist alternative concepts to incorporate boundary conditions for Eulerian fluids. Batty et al. [23] improve the FLIP method in [24] for two-way rigid-fluid coupling. In this approach, the pressure projection is formulated as a kinetic energy minimization problem. Carlson et al. [25] use Distributed Lagrangian multipliers to project fluid nodes covered by rigid bodies onto rigid body motion. Génevaux et al. [26] use damped springs to attach solids to fluid marker particles. Guendelmann et al. [27] present an alternating two-way coupling for deformable and rigid thin shells. This algorithm uses ray-casting to avoid fluid leaking through thin solids represented by triangles. Liu et al. [28] present a GPU approach for the semi-Lagrangian scheme of Stam [12]. Arbitrary boundary conditions for the fluid simulation are generated directly in image space.

Some authors propose a mixed formulation, using an Eulerian formulation for the fluid and a Lagrangian formulation for the solid. The Immersed Boundary Method (IBM) introduced by Peskin [29], [30] samples a solid with a finite set of force points. As the boundary velocities are interpolated on the grid, boundaries are not required to coincide with the fluid grid. Therefore, the approach is appropriate for irregular and detailed geometries. In the context of IBM, direct forcing has been employed [31], [32], [33]. A force term is added to the discretized momentum equations to obtain the adequate velocity of the fluid along the boundary after a single time step. This direct forcing approach is free of parameters and can therefore be handled conveniently. The Immersed Interface Method of Le et al. [34] improved on IBM for rigid boundaries and moving deformable solids. It can handle sharp interfaces, since forces are not distributed on the boundary.

Fedkiw [35] uses so-called ghost fluid nodes to couple compressible Eulerian fluids and deformable Lagrangian solids. Ghost fluid nodes can be covered by the solid but are used in the finite difference scheme for the fluid update. The Eulerian and the Lagrangian parts of the simulation are properly interpolated. Combining Eulerian grids and Lagrangian meshes has also been proposed in the finite-element Arbitrary Lagrangian Eulerian (ALE) method of Hirt et al. [36]. A mixed formulation that couples SPH particles and particle level sets has been introduced by

Losasso et al. [37] to reduce volume loss for free surfaces. Robinson-Mosher et al. [38] recently proposed an approach to couple Cartesian fluid grids and Lagrangian solids derived from the law of conservation of momentum.

Most of the aforementioned methods cannot directly be applied to pure particle-based fluids and only a few methods have been proposed for pure particle-based simulations until now. Most authors use penalty-based approaches to handle static or moving rigid boundaries. The main concept of penalty-based approaches is to use either frozen or ghost particles. Frozen particles interact with other particles in the usual way, but they do not move. Ghost particles, on the other hand, are fluid particles mirrored across solid boundaries in each time step. Monaghan [2] proposes a force-based penalty method for fixed and moving boundaries, and in [6], a penalty boundary potential is used to calculate penalty forces. Keiser et al. [39] present a Lagrangian formulation to handle solids, fluids, and phase transitions. Solenthaler et al. [40] and Keiser et al. [41] process fluid and rigid-body particles in similar ways. Rigid-body particles are restricted to rigid-body motion in the update step. Falappi and Gallati [42] handle interacting fluids and granular materials by using SPH for both phases. Müller et al. [43] employ a penalty approach based on Lennard-Jones forces for repulsion and adhesion between mesh-based deformable solids and particle-based fluids.

There have been few approaches that try to model different kinds of boundary conditions with penalty methods. Ghost particles with the same mass, density, pressure, and viscosity but different velocity than their fluid counterpart have been employed to handle different slip conditions for straight [44] and curved [45] surfaces. However, penalty methods suffer from severe difficulties. As penalty forces only react upon penetration, the distance of SPH particles to the boundary slightly varies over time and particles might be accelerated at the boundary. Only limited control is offered to realize specific boundary conditions. Most approaches so far do not offer an easy way to adjust tangential damping in the full range from no- to free-slip. Finally, to ensure nonpenetration, large forces have to act on the fluid, leading to stiff equations. To illustrate the benefits of the proposed direct forcing approach, we provide a comparison with the penalty approach of Monaghan [2] in Section 6.

Few approaches have taken into account the actual forces acting on the boundary for the boundary handling so far. Oger et al. [46] propose a method for the two-way coupling of a particle-based fluid and a moving solid in two dimensions. However, only the local pressure on the solid surface is evaluated. Dynamic forces, e.g., due to viscosity, are neglected. Hieber [3] adopts IBM for the boundary handling of particle-based fluids with deformable solids and fixed boundaries, thereby allowing a greater amount of control of the fluid and taking into account all forces acting on the boundary. However, nonpenetration is not addressed. Additionally, a fixed number of Lagrangian force points is used to sample the boundary and an underlying Eulerian grid is used to transfer values between the fluid simulation and the force points. We share the idea of Hieber [3] to modify the underlying momentum equation to ensure the chosen boundary conditions with a direct forcing approach. However, in contrast to [3], forces can be applied directly at the contact point, and thereby, force interpolations are avoided. Nonpenetration is guaranteed after each time step and fluid leaking is avoided. This is especially important for the two-way coupling with rigid bodies due to the limited number of degrees of freedom. We furthermore propose a simple scheme to cover different kinds of velocity boundary conditions. The approach is realized in a predictor-corrector fashion.

The concept of direct forcing has been successfully applied in other simulation areas such as constraint handling for rigid [47], [48], [49] and deformable [50], [51] solids. However, these approaches are beyond the scope of this paper. The same applies to boundary conditions for other fluid methods such as the Lattice Boltzmann method [52]. The idea of predictor-corrector schemes has also been successfully applied in contact and collision handling for rigid and deformable bodies, see, e.g., [4] and [53].

3 Fluid Model

For the fluid simulation, we use a corrected SPH formulation (CSPH) [6] and the weakly compressible pressure formulation employing the Tait equation [7], [8].

The basic idea of SPH is to represent a function $f(\mathbf{x})$ as a smoothed function $\langle f(\mathbf{x})\rangle$ using a finite set of sampling points $\mathbf{x}_b$ with mass m_b and density ρ_b, and a kernel function $W_b(\mathbf{x}) = W(\mathbf{x} - \mathbf{x}_b)$:

$$\langle f(\mathbf{x})\rangle = \sum_b \frac{m_b}{\rho_b} f(\mathbf{x}_b) W_b(\mathbf{x}). \tag{1}$$

However, this original SPH formulation developed independently by Gingold and Monaghan [54] and Lucy [55] suffers from inaccurate calculations at boundaries. Since the number of neighboring particles at boundaries is smaller than inside the fluid, an incorrect lower density is calculated and negative pressures can occur. As we extensively deal with boundaries in rigid-fluid coupling, we address this problem by using the constant correction technique for SPH proposed in [6]. By using an adapted kernel function

$$\tilde{W}_b(\mathbf{x}) = \frac{W_b(\mathbf{x})}{\sum_c V_c W_c(\mathbf{x})} \tag{2}$$

for the density calculation, this model avoids inaccurate pressures at boundaries. The overhead for the computation of the adapted kernel function $\tilde{W}_b(\mathbf{x})$ is negligible. Since the volume V_c of a particle is constant, it can be precomputed. Thus, an additional loop over the particles is avoided. For the computation of the fluid dynamics, we use the reformulated Euler equation with external forces denoted by $\mathbf{g}$

$$\frac{d\mathbf{v}}{dt} = -\left(\nabla\left(\frac{P}{\rho}\right) + \frac{P}{\rho^2}\nabla\rho\right) + \mathbf{g} \tag{3}$$

with pressure P, density ρ and velocity $\mathbf{v}$. It results in a symmetrized discrete momentum equation that conserves linear and angular momentum [2]. The discrete momentum equation for the acceleration $\frac{d\mathbf{v}_a}{dt}$ of a particle with added

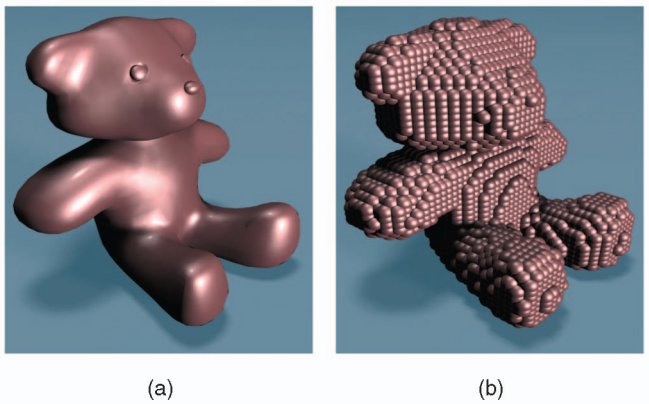


Fig. 2. Triangulated surface and particle representation of the teddy model.

artificial viscosity [2] and the original kernel function W_b thereby reads

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \Pi_{ab} \right) \nabla_a W_b(\mathbf{x}_a) + \mathbf{g} \qquad (4)$$

with the viscosity term

$$\Pi_{ab} = -\nu \left(\frac{\mathbf{v}_{ab}^T \mathbf{x}_{ab}}{|\mathbf{x}_{ab}|^2 + \delta h^2} \right). \qquad (5)$$

The pressure P is calculated using the Tait equation [7], [8] to ensure small density ratios between the current density ρ and the initial density ρ_0:

$$P = \frac{\rho_0 c_s^2}{7} \left(\left(\frac{\rho}{\rho_0} \right)^7 - 1 \right). \qquad (6)$$

The speed of sound c_s is usually chosen such that the Mach number of the simulation is below 0.1. For further details, see, e.g., [8].

4 Rigid Body Model

In the context of rigid-fluid coupling, various rigid-body representations have been proposed, e.g., triangle meshes [43], adaptively sampled distance fields [56], and particles [2]. Similar to [2], we employ a particle representation for arbitrary rigid-body surfaces. The particle representation is generated in a preprocessing step using a distance field [57]. An example is shown in Fig. 2.

Although the proposed rigid-body representation allows for a unified handling of rigid bodies and fluids in certain aspects of the simulation, e.g., particle-particle collision tests, our boundary handling approach could be combined with alternative representations such as triangle meshes or distance fields. These representations, however, are beyond the scope of this paper.

5 Boundary Handling

In this section, we introduce a novel technique to enforce boundary conditions for particle-based rigid-fluid contacts. Boundary conditions model the relative velocities and positions of the fluid at the boundary. Our approach allows to control both the normal and tangential relative velocities and the relative positions effectively to realize various boundary conditions. The relative velocities and positions are controlled in separate substeps as, e.g., proposed in [4] and [53] for collision and contact handling of rigid and deformable solids. The boundary model for the relative velocities is discussed in Section 5.1. Enforcing the desired velocities and positions is realized using a direct forcing approach discussed in Sections 5.2 and 5.3, respectively. It is implemented in a predictor-corrector fashion. Nonpenetration is thereby addressed. In Section 5.4, we discuss the entire pipeline of a single simulation step. At the end, simplifications for one-way solid-to-fluid coupling and static boundaries are discussed. To detect collisions, we follow [53] in advancing the positions without boundary forces and performing the collision detection on that advanced positions. Intermediate advanced values are denoted with a single or double asterisk $(^*, ^{**})$.

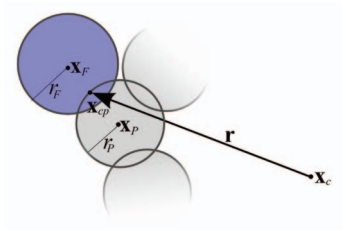


Fig. 3. Rigid particle $\mathbf{x}_P$ and fluid particle $\mathbf{x}_F$ in contact. The position $\mathbf{x}_{cp}$ denotes the contact point of the fluid and the rigid body. The values r_F and r_P denote the radii of the respective particles. The vector $\mathbf{r}$ denotes the distance of the contact point $\mathbf{x}_{cp}$ to the center of mass $\mathbf{x}_c$ of the rigid body.

5.1 Controlling the Relative Velocity

In this section, we discuss the employed model for the velocity control at the boundary. As noted in Section 4, rigid bodies are sampled with particles. If we detect a collision between a fluid particle with position $\mathbf{x}_F$, velocity $\mathbf{v}_F$, mass m, and a rigid-body particle $\mathbf{x}_P$, we calculate a contact point $\mathbf{x}_{cp} = \mathbf{x}_P + r_P \mathbf{n}$ at the boundary of the colliding rigid-body particle (see Fig. 3). $\mathbf{n}$ denotes the unit surface normal of the rigid body at $\mathbf{x}_{cp}$. m_c is the total mass of the colliding rigid body.

The rigid-body velocity at the contact point is given by

$$\mathbf{v}_{cp}^*(t+h) = \mathbf{v}_c^*(t+h) + \omega_c^*(t+h) \times \mathbf{r}^*(t+h) \qquad (7)$$

with $\mathbf{r}^*(t+h) = \mathbf{x}_{cp}^*(t+h) - \mathbf{x}_c^*(t+h)$ being the relative position of $\mathbf{x}_{cp}^*(t+h)$ with respect to the center of mass $\mathbf{x}_c^*(t+h)$ of the rigid body. $\mathbf{v}_c^*(t+h)$ and $\omega^*(t+h)$ denote the linear and angular velocities of the rigid body, respectively.

Now, we want to impose a boundary condition on the relative velocity $\mathbf{v}_r = \mathbf{v}_F - \mathbf{v}_{cp}$ of the following form:

$$\mathbf{v}_r(t+h) := \mathbf{v}_F(t+h) - \mathbf{v}_{cp}(t+h) = \varepsilon \left[\mathbf{v}_r^*(t+h) \right]_t - \delta [\mathbf{v}_r(t)]_n. \qquad (8)$$

The current normal velocity is thereby given as $[\mathbf{v}_r(t)]_n = (\mathbf{v}_r(t) \cdot \mathbf{n})\mathbf{n}$ and the uncontrolled tangential velocity of the next time step is given as $[\mathbf{v}_r^*(t+h)]_t = \mathbf{v}_r^*(t+h) - [\mathbf{v}_r^*(t+h)]_n$. The first term of (8) controls the slip. It can be

used to damp the relative tangential velocity of the fluid and the rigid body. Here, we use the predicted velocity of the subsequent time step to properly consider accelerations due to body forces such as gravity. For example, for a free-slip condition, the component of the fluid velocity tangential to a vertical boundary would not be damped. The second term of (8) controls the elasticity of the collision. δ is called the coefficient of restitution. $\delta = 1.0$ thereby corresponds to a perfectly elastic collision, while $\delta = 0.0$ results in a perfectly inelastic collision. If not stated otherwise, we use $\delta = 0.0$ in our scenarios, i.e., the relative normal velocity between the fluid and the rigid body at the boundary layer is zero. Both damping parameters ε and δ are always in the interval [0, 1].

To avoid sticking, we substitute the boundary condition (8) by

$$\mathbf{v}_r(t+h) = \varepsilon\big[\mathbf{v}_r^*(t+h)\big]_t + \big[\mathbf{v}_r^*(t+h)\big]_n \tag{9}$$

for $\mathbf{v}_r^*(t+h)\cdot\mathbf{n} > 0$. This leaves the normal component of the relative velocity unchanged if the fluid particle and the rigid body are moving away from each other. To simplify the subsequent explanations, we generally use (8) and omit (9) due to the similarities of both cases.

5.2 Velocity Update

In this section, the enforcement of the velocity constraints is described. We first handle the case of a single fluid particle in contact with a rigid body. Then, we generalize the idea to several fluid particles in contact with a single rigid body. For a single fluid particle with index i, the contact point of the fluid particle with the rigid body has the absolute position $\mathbf{x}_{cp,i}^*(t+h)$ and the relative position $\mathbf{r}_i^*(t+h)$ with respect to the center of mass of the rigid body. To enforce our boundary condition (8) on the relative velocity $\mathbf{v}_{r,i}$, we exchange a control force $\mathbf{F}_i$ between the fluid particle and the rigid body. Assuming a simple Euler step, we end up with the constrained velocities for the fluid particle

$$\mathbf{v}_i(t+h) = \mathbf{v}_i^*(t+h) + \frac{h}{m_i}\mathbf{F}_i \tag{10}$$

and the rigid body at the contact point

$$\begin{aligned}\mathbf{v}_{cp,i}(t+h) = \mathbf{v}_{cp,i}^*(t+h) - \frac{h}{m_c}\mathbf{F}_i \\ + h\tilde{\mathbf{r}}_i^*(t+h)\mathbf{I}^{-1}(t)\tilde{\mathbf{r}}_i^*(t+h)\mathbf{F}_i,\end{aligned} \tag{11}$$

with $\tilde{\mathbf{r}}$ being the cross product matrix of the vector $\mathbf{r}$. In order to predict the velocities $\mathbf{v}_i^*(t+h)$ and $\mathbf{v}_{cp}^*(t+h)$, we take into account all forces such as pressure forces, viscous forces, and gravity.

Using the right-hand sides of (10) and (11) for the relative velocity $\mathbf{v}_{r,i}(t+h)$ in the constraint equation (8) and solving for the unknown control force $\mathbf{F}_i$ yields

$$\mathbf{F}_i = \frac{1}{h}\left[\left(\frac{1}{m_i}+\frac{1}{m_c}\right)\mathbf{E}_3 + \tilde{\mathbf{r}}_i^{*T}(t+h)\mathbf{I}^{-1}(t)\tilde{\mathbf{r}}_i^*(t+h)\right]^{-1}\hat{\mathbf{v}}_i, \tag{12}$$

with $\hat{\mathbf{v}}_i := \varepsilon[\mathbf{v}_{r,i}^*(t+h)]_t - \delta[\mathbf{v}_{r,i}(t)]_n - \mathbf{v}_{r,i}^*(t+h)$ and the 3×3 identity matrix $\mathbf{E}_3$. If we have k fluid particles in contact with a single rigid body, solve each contact separately using (12), and simply add up the forces and torques on the rigid body, we term this **local approach**.

Now, we assume that a single rigid body is in contact with k fluid particles and we want to enforce all contact velocities simultaneously. This is termed **global approach**. Similar to the case of a single contact, we apply symmetric control forces $\mathbf{F}_i$ at each contact point. For the rigid body, these control forces sum up to a net force $\mathbf{F}$ and a net torque τ:

$$\mathbf{F} = -\sum_i \mathbf{F}_i, \tag{13}$$

$$\tau = -\sum \mathbf{r}_i^*(t+h)\times\mathbf{F}_i. \tag{14}$$

Once $\mathbf{F}$ and τ are known, we can calculate the future linear and angular velocities of the rigid body and thereby the rigid-body velocities $\mathbf{v}_{cp,i}(t+h)$ at the ith contact point. The velocities of the fluid particles $\mathbf{v}_i(t+h)$ can then be calculated using the constraint equations

$$\mathbf{v}_i(t+h) = \varepsilon\Big[\mathbf{v}_{r,i}^*(t+h)\Big]_t - \delta\big[\mathbf{v}_{r,i}(t)\big]_n + \mathbf{v}_{cp,i}(t+h). \tag{15}$$

To derive our system of equations for $\mathbf{F}$ and τ, we express the future velocity at the ith contact point as

$$\mathbf{v}_{cp,i}(t+h) = \mathbf{v}_{cp,i}^*(t+h) + \frac{h}{m_c}\mathbf{F} + h\tilde{\mathbf{r}}_i^{*T}(t+h)\mathbf{I}^{-1}(t)\tau. \tag{16}$$

Plugging this future velocity $\mathbf{v}_{cp,i}(t+h)$ for the contact point and the future velocity $\mathbf{v}_i(t+h) = \mathbf{v}_i^*(t+h) + \frac{h}{m_i}\mathbf{F}_i$ for the fluid particle into the constraint equation (15) and solving for the unknown constraint force $\mathbf{F}_i$ yields

$$\mathbf{F}_i = \frac{m_i}{h}\left[\hat{\mathbf{v}}_i + \frac{h}{m_c}\mathbf{F} + h\tilde{\mathbf{r}}_i^{*T}(t+h)\mathbf{I}^{-1}(t)\tau\right], \tag{17}$$

with the unknown net force $\mathbf{F}$ and the net torque τ on the right-hand side. The control forces $\mathbf{F}_i$ are now plugged into (13) and (14) for the net force $\mathbf{F}$ and torque τ on the rigid body. As a result, we get the symmetric, positive definite 6×6 linear system of equations for the unknowns $\mathbf{F}$ and τ

$$\mathbf{A}\begin{bmatrix}\frac{1}{m_c}\mathbf{F}\\ \mathbf{I}^{-1}(t)\tau\end{bmatrix} = \begin{bmatrix}-\sum\frac{m_i}{h}\hat{\mathbf{v}}_i\\ -\sum\frac{m_i}{h}\tilde{\mathbf{r}}_i^*(t+h)\hat{\mathbf{v}}_i\end{bmatrix}, \tag{18}$$

with the system matrix

$$A := \begin{bmatrix}(m_c+\sum m_i)\mathbf{E}_3 & \sum m_i\tilde{\mathbf{r}}_i^{*T}(t+h)\\ \sum m_i\tilde{\mathbf{r}}_i^*(t+h) & \mathbf{I}(t)+\sum m_i\tilde{\mathbf{r}}_i^*(t+h)\tilde{\mathbf{r}}_i^{*T}(t+h)\end{bmatrix}.$$

The employed concept is closely related to the interpenetration resolution scheme in [4] and [5]. In this approach, collisions between several rigid bodies are handled by setting one rigid body as central body and the others as outer bodies. The outer bodies are pushed out of the central body in a two-way coupled fashion. However, instead of interpenetration resolution we adopt this method to enforce our boundary velocity constraints in a two-way coupled fashion by applying symmetric forces.

Since the velocities of the fluid particles at the boundary are completely determined by the constraint equations, the boundary velocity calculations for k contact

TABLE 1
Some Exemplary Scenes: Number of Particles, Computation Time for One Simulation Step, Sound Speed, and Maximum Density Ratio

scene	fluid particles	cluster particles	fluid calc. [s]	simulation step [s]	speed of sound	max density ratio
Impact	850k	9k	3.47	7.07	600	1.19
Floating cuboids	2M	7.5k	5.79	11.4	250	1.13
Floating spheres	130k	760	0.56	1.14	250	1.026
Stone-skipping	240k	94	0.83	1.64	300	1.07
Flotsam	2.57M	18.6k	7.61	15.03	225	1.16

points at a single rigid body can be reduced to the linear 6×6 system of equations in (18). In Section 6, we discuss several experiments using both the local and global approaches. Additionally, we compare the performance of both approaches.

This section has discussed the velocity update of the rigid body and the fluid particles. The position update for interpenetration handling is addressed in the Section 5.3.

5.3 Position Update

In addition to the correct relative velocity at the boundary, we want to enforce nonpenetration of the fluid particles with respect to the boundary. We therefore control the position of the fluid particles at the boundary in a separate substep. We enforce the centers of the boundary fluid particles with radius r_i to retain a distance r_i to the contact point $\mathbf{x}_{cp,i}$. Since we have considered this contact point for the computation of the control force, such a position correction does not influence the slip condition. The corrected position update is implemented using an additional control impulse $\mathbf{j}_i = j_i\mathbf{n}$ acting in normal direction, which is only applied in the position update of the integration step. It is calculated in the same manner as the control force for the velocity update. To meet the desired distance, we need to enforce

$$\left[\mathbf{x}_i^{**}(t+h) + h\mathbf{j}_i - \mathbf{x}_{cp,i}(t+h)\right]\mathbf{n} = r_F. \tag{19}$$

As the update is only applied to the fluid particles, we can use the final contact point position $\mathbf{x}_{cp,i}(t+h)$. Both $\mathbf{x}_{cp,i}(t+h)$ and the predicted future fluid positions $\mathbf{x}_i^{**}(t+h)$ are calculated using the modified velocities from the subsequent step. The control impulses $\mathbf{j}_i$ are then computed as

$$\mathbf{j}_i = \frac{1}{h}\left[(\mathbf{x}_{cp,i}(t+h) - \mathbf{x}_i^{**}(t+h)) \cdot \mathbf{n} + r_F\right]\mathbf{n}. \tag{20}$$

Unfortunately, the position update leads to a higher compression of the fluid at the boundary layer. However, higher density ratios are rapidly balanced in subsequent time steps with the employed weakly compressible pressure approach. The maximum density ratios encountered in our simulations are given in Table 1.

5.4 Two-Way Coupling

In Algorithm 1, we show all stages of a single simulation step. As we need to know the unconstrained velocities $\mathbf{v}_i^*(t+h)$ and $\mathbf{v}_{cp,i}^*(t+h)$ to calculate our control force $\mathbf{F}$ and torque τ, we perform a predictive integration step for the fluid and the rigid bodies. In the correction step, we only consider fluid particles and rigid bodies that are in contact. The same holds for the position update. We thereby take into account the modified velocities. Overall, up to three collision detection steps are performed.

Algorithm 1 Pseudocode for two-way coupled moving boundaries

Require: n fluid particles, m rigid bodies
1: Detect fluid-fluid collisions
2: Calculate fluid and rigid-body forces
3: Integrate fluid and rigid body (prediction)
$\mathbf{x}(t) \rightarrow \mathbf{x}^*(t+h)$, $\mathbf{v}(t) \rightarrow \mathbf{v}^*(t+h)$
4: Detect rigid-fluid collisions
5: Calculate net force $\mathbf{F}$ and net torque τ
6: Integrate fluid and rigid body (correction)
$\mathbf{x}^*(t) \rightarrow \mathbf{x}^{**}(t+h)$, $\mathbf{v}^*(t+h) \rightarrow \mathbf{v}(t+h)$
7: **if** (any contacts in 4) **then**
8: Detect rigid-fluid collisions
9: Correct fluid positions $\mathbf{x}^{**}(t+h) \rightarrow \mathbf{x}(t+h)$
10: **end if**

Steps 7-9 in Algorithm 1 are only performed once, even if some penetrations are not resolved due to conflicting constraints. See Section 7 for some notes on this issue.

5.5 One-Way Solid-to-Fluid Coupling and Static Boundaries

For the two-way coupling, we need up to three collision detection steps, which are comparatively time-consuming. In some cases, however, the influence of the fluid on the solid is small and could be neglected (e.g., heavy objects) or the solid does not move at all. For this case, we propose to use a more efficient one-way solid-to-fluid coupling. In this one-way coupling, the solid influences the fluid, but not vice versa. The rigid-body velocity at the contact point in (11) or (16), respectively, thereby simplifies to

$$\mathbf{v}_{cp,i}(t+h) = \mathbf{v}_{cp,i}^*(t+h) \tag{21}$$

for one-way coupling or

$$\mathbf{v}_{cp,i}(t+h) = \mathbf{v}_{cp,i}(t) = 0 \tag{22}$$

for static boundaries, respectively. The fluid velocity can be calculated from the boundary conditions (8) in the usual way.

As the rigid body is integrated prior to the control force calculation and as the rigid body is not affected by any corrections, we can calculate the velocity and position correction for the fluid in one step. This saves one collision detection step compared to the two-way coupling. As the collision detection is comparatively time-consuming, the

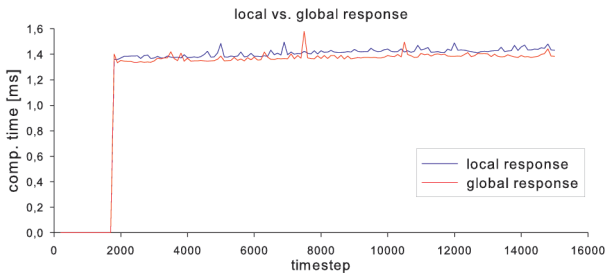


Fig. 4. The diagram shows the computation time for the local and the global response schemes for the sinking ship scene. Both are linear in the number of contacts and make up only a small fraction of the total computation time of 250-280 ms.

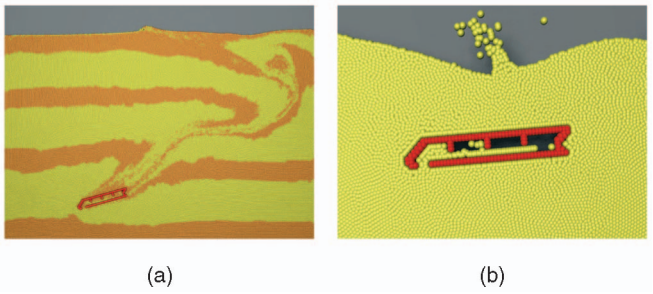


Fig. 5. Two-way coupling of a sinking ship with a fluid. Colored particles on the left indicate the flow of the fluid. The picture on the right indicates the accuracy of the boundary handling.

efficiency can be significantly improved using the one-way coupling. Additionally, there is no need to solve a system of equations, as the fluid velocities can be directly computed from the boundary conditions. All stages of a single simulation step for the one-way coupling are summarized in Algorithm 2.

Algorithm 2 Pseudocode for one-way coupled boundaries
Require: n fluid particles, m rigid bodies
1: Detect fluid-fluid collisions
2: Calculate fluid and rigid-body forces
3: Integrate fluid and rigid body (prediction)
$\mathbf{x}(t) \rightarrow \mathbf{x}^*(t+h)$, $\mathbf{v}(t) \rightarrow \mathbf{v}^*(t+h)$
4: Detect fluid-rigid collisions
5: Calculate fluid velocity and position (correction)
$\mathbf{x}^*(t+h) \rightarrow \mathbf{x}(t+h)$, $\mathbf{v}^*(t+h) \rightarrow \mathbf{v}(t+h)$

5.6 Parameters

The time step for the simulation is chosen according to the Courant-Friedrichs-Lewy (CFL) convergence condition. The resulting time step is generally rather restrictive with respect to stability of the fluid and it has turned out that the simulations remain stable when the two- or one-way coupling is incorporated. For the rigid-fluid interaction, we basically need to ensure that two particles do not move more than their diameter toward each other in one time step. In the experiments, we use time steps ranging from $6 \cdot 10^{-5}$ to $1.5 \cdot 10^{-4}$.

The estimation of appropriate parameters for different boundary conditions can be a tedious task, particularly as the effects of the parameters typically depend on the time step. In our approach, the handling of parameters is comparatively easy. First of all, the total number of free parameters is only two, namely the damping parameters ε and δ. Second, ε and δ are always in the interval [0, 1].

5.7 Implementation Issues

The efficient detection of particle-particle contacts is one of the fundamental issues in particle-based fluid simulations. Similar to [57], we use a uniform spatial subdivision and store the results in a hash table [58]. We also follow [57] in employing temporal coherence, i.e., we only update the information of particles if their grid cell has changed. Due to the restrictive time step, temporal coherence significantly speeds up the insertion of particles into the hash table.

6 Results

In this section, we illustrate the capabilities of our boundary handling technique with 2D and 3D experiments that range from some simple explanatory scenes to high-velocity impacts. We make use of both the local and global approaches for updating the boundary velocity. The following experiments are performed: As many authors use penalty methods in their simulations, we first compare the proposed approach to the penalty approach in [2]. Then, we demonstrate different slip conditions in a 2D setting. Handling high-velocity impacts and one-way coupling is illustrated with a stone impacting a water basin. Sections 6.5 and 6.6 illustrate buoyancy and drag effects. Finally, we show some advanced two-way coupled scenes.

Table 1 gives an overview of the performance for the scenarios. All performance measurements are given with respect to an Intel Dual Core 2.13 GHz with 4 Gbytes of RAM, running a single-threaded version of the simulation. In all simulations, we use an explicit leapfrog integration scheme. If not stated otherwise, the viscosity is set to $\nu = 0.1$. As pointed out in [8], the compressibility of the fluid is governed by the speed of sound. The speed of sound has been chosen with respect to the maximum relative velocity between fluid and solids to ensure a certain density ratio. Both values, the speed of sound and the measured density ratio, are stated in Table 1. We also give the performance for a single fluid calculation step without boundary handling and a complete simulation step including the boundary handling.

For the performed experiments, we have used either the local or the global approach. Both approaches are linear in the number of contacts and make up less than 1 percent of the total computation time in most scenarios. Both approaches show plausible results, whereas in the global approach the boundary velocities are met more accurately. All timings in Table 1 are given using the local approach. In Fig. 4, we compare the local and global responses with respect to their performance for the sinking ship scene illustrated in Fig. 5.

For the reconstruction of the fluid surface, we employ marching cubes [59], while single particles are handled as blobs. Triangulated surfaces are employed for visualizing the rigid bodies. Surfaces and blobs are rendered in POV-Ray (http://www.povray.org). In some cases, we visualize the underlying particle simulation for illustration purposes.

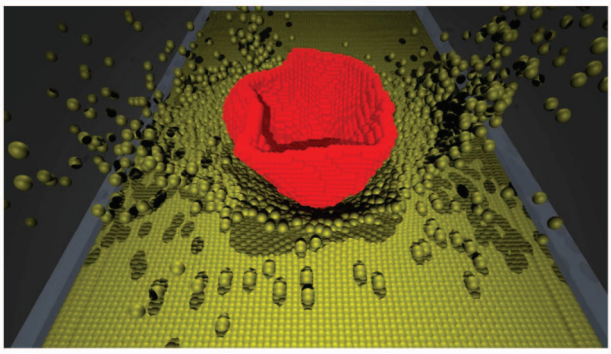


Fig. 6. A vessel is floating in a water basin. No fluid is leaking through the boundary.

6.1 Comparison to a Penalty-Based Approach

Many authors employ penalty methods for handling boundaries in particle-based simulations. We compare the proposed local approach with the penalty-based approach in [2]. We have chosen a 2D example of a leaking ship sinking into a fluid to illustrate the effects of both boundary methods without getting distracted by the surface reconstruction. Large parts of the ship are only represented by a single layer of particles. As for the penalty method, several effects can be observed in the experiment:

- Penalty methods offer only limited control. To ensure nonpenetration, large penalty forces have to be applied. This leads to elastic collisions with an unknown coefficient of restitution.
- Penalty methods can only react in a subsequent time step, if a penetration has already occurred. Therefore, the distance of fluid particles to the boundary slightly varies over time. Additionally, unnatural accelerations can occur. For the sinking ship, this leads to the effect that the ship is not correctly filled and single particles are bouncing on the surface.

The proposed local and global approaches can cope with these issues. As velocity and position are controlled in different substeps, nonpenetration and inelastic collision can be realized at the same time. As velocity and position are predicted and corrected in the same simulation step, constant distances can be realized and unnatural accelerations of fluid particles at the boundary are avoided. The ship is properly filled with fluid particles and sinks into the basin. Even in case where it is fully submerged, nonpenetration with constant distance can be enforced. Some results can be seen in Fig. 5.

6.2 Accuracy

To illustrate accuracy and nonpenetration of our method in 3D, we simulate a vessel falling into a large basin of fluid. Fig. 6 illustrates this setting. Although the boundary is represented only by a single layer of particles, no fluid is leaking through the boundary. The velocity update is performed using the local approach.

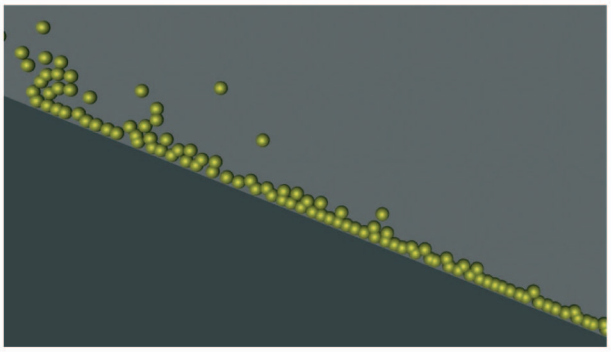


Fig. 7. Experimental setup for the slip condition.

6.3 Slip Condition

Imposing different kinds of slip conditions is a challenging issue. However, using the proposed method, slip can easily be controlled. In Fig. 7, we show the experimental setup for the illustration of different slip conditions. Particles are emitted on the left-hand side, flowing down a static ramp. Different slip conditions ranging from $\varepsilon = 0.0$ (no-slip) to $\varepsilon = 1.0$ (free-slip) lead to different flow properties. We refer the reader to the accompanying video to assess the effects in a dynamic simulation.

6.4 One-Way Solid-to-Fluid Coupling

Fig. 8 illustrates one-way solid-to-fluid coupling with an impact scene of an asteroid model. Due to the high velocity of the asteroid, the speed of sound is set to 600. The maximum density ratio at the time of the impact is 1.19. For the rest of the simulation, the density ratio is below 1.1. The experiment indicates that high relative velocities can be handled.

6.5 Buoyancy Effects

In Fig. 9, we illustrate that buoyancy effects are properly captured using the proposed global response scheme. Three cuboids of different densities are dropped into a fluid. As expected, the lightest cube (red) is floating, the cube with

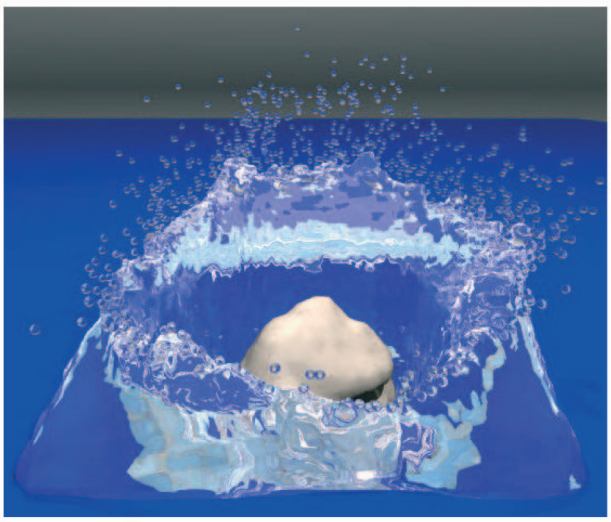


Fig. 8. Impact: High-velocity impact of an asteroid.

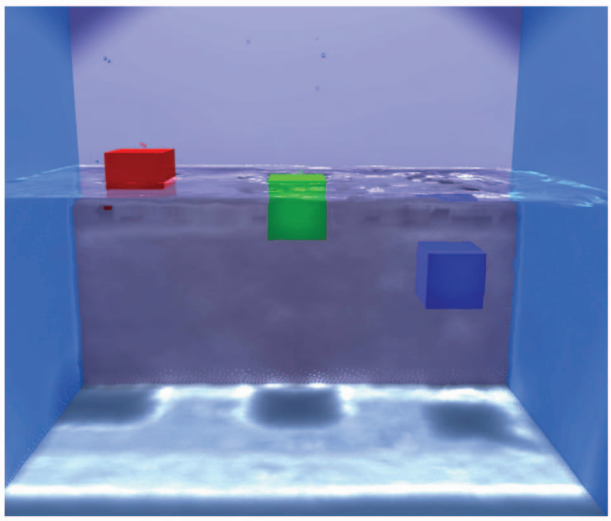


Fig. 9. Buoyancy effects: Cuboids of different densities dropped into a fluid.

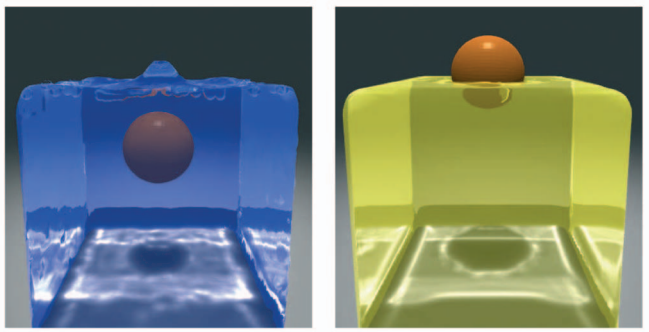
medium density (green) is sinking slowly, and the heaviest cube (blue) is sinking fast.

6.6 Drag Effects

Some approaches such as [46] take into account only the pressure forces acting on the boundary. Effects due to dynamic forces such as viscosity are not properly captured. The following experiment illustrates that the influence of dynamic forces on moving rigid bodies is properly captured with our method. A rigid sphere is dropped into fluids of different viscosities $(\nu = 0.1, \nu = 3.0)$. Fig. 10 depicts both scenarios at the same time point. The images show that the sphere is sinking deeper in the low-viscosity fluid compared to the high-viscosity fluid. The local velocity update is employed for the boundary handling in this setting.

6.7 Two-Way Coupling

The following two experiments further illustrate the proposed two-way coupling approach. Figs. 1 and 11 show a stone-skipping experiment. Due to its high velocity, the stone is reflected at the fluid surface. The initial velocity is about 90 percent of the velocity of the impact scenario. For



(a) (b)

Fig. 10. Viscous effects: Spheres dropped into fluids with different viscosities. Viscosity is set to 0.1 in the left image and to 3.0 in the right image.

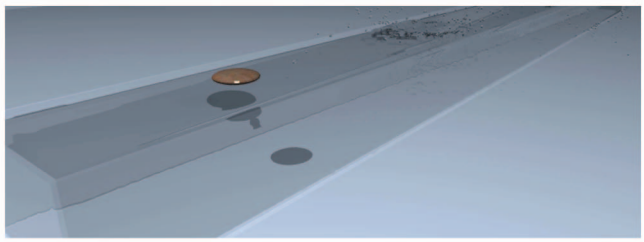


Fig. 11. Stone skipping.

low velocities at the end, the stone finally sinks. Again, we would like to refer the reader to the accompanying video to assess the dynamics. In this scenario, the local and the global boundary handling approaches show very similar dynamics. Fig. 11 is simulated using the global approach. Fig. 1 illustrates the local approach.

A second experiment to illustrate the two-way coupling is shown in Fig. 12. Here, complex-shaped rigid objects are floating on a wave. Again, the boundary handling is performed using the local approach.

7 Conclusion and Future Work

We have presented an efficient Lagrangian method for the handling of fixed and moving boundaries. Direct forcing is employed to realize a large range of slip and Neumann boundary conditions. The proposed technique can be used for one- and two-way coupling with arbitrarily shaped boundaries that are represented with particles. Static and dynamic forces are properly taken into account to allow for buoyancy and drag effects. In contrast to previous Lagrangian methods, overlaps of fluid and rigid-body particles are avoided. The proposed method is superior to previously used penalty-based approaches such as [2]. It offers a greater amount of control, ensures nonpenetration, and does not introduce stiffness to the system. At the same time, it is computationally efficient and scales linearly in the number of contact points. We have made several tests using two variants of the approach, namely handling each contact

Fig. 12. Boundary handling for arbitrarily shaped objects illustrated with flotsam on the beach.

point separately and solving all boundary velocities at once. Both methods show plausible results and a very similar performance. While the global approach is more accurate, the local approach allows to directly process collision pairs. The setup of a system of equations is thereby avoided.

The presented schemes work with compressible and weakly compressible models. This restriction allows one to avoid global computations, i.e., computations that take into account the state of the whole fluid domain. Since particle-based fluids scale well for large scenes, the proposed boundary handling approach is particularly interesting for complex scenes with irregular simulation domains. Further, the underlying model for controlling the relative velocities is easy to adjust, as the number of free parameters is low. Additionally, the range of the parameters is known.

We currently do not handle simultaneous contact of a single fluid particle with more than one rigid body and simultaneous contact of several rigid bodies in a fluid. We believe that sophisticated but often expensive methods such as contact graphs could be employed to handle such settings.

ACKNOWLEDGMENTS

The digital asteroid model of the impact scene is courtesy of Scott Hudson, Washington State University. The bunny model is courtesy of the Computer Graphics Laboratory, Stanford University. The authors would like to thank the reviewers for their valuable suggestions during the review process that helped improve the global approach.

REFERENCES

[1] R. Bridson and M. Müller-Fischer, "Fluid Simulation: SIGGRAPH 2007 Course Notes," *ACM SIGGRAPH '07 Courses*, pp. 1-81, 2007.
[2] J. Monaghan, "Smoothed Particle Hydrodynamics," *Reports on Progress in Physics*, vol. 68, no. 8, pp. 1703-1759, 2005.
[3] S. Hieber, "Particle-Methods for Flow-Structure Interactions, " PhD dissertation, Swiss Fed. Inst. of Technology Zürich (ETHZ), 2007.
[4] T. Shinar, C. Schroeder, and R. Fedkiw, "Two-Way Coupling of Rigid and Deformable Bodies," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA)*, 2008.
[5] T. Shinar, "Simulation of Coupled Rigid and Deformable Solids and Multiphase Fluids, " PhD dissertation, Stanford Univ., June 2008.
[6] J. Bonet and S. Kulasegaram, "A Simplified Approach to Enhance the Performance of Smooth Particle Hydrodynamics Methods," *Applied Math. and Computation*, vol. 126, nos. 2-3, pp. 133-155, 2002.
[7] J. Monaghan, "Simulating Free Surface Flows with SPH," *J. Computational Physics*, vol. 110, no. 2, pp. 399-406, 1994.
[8] M. Becker and M. Teschner, "Weakly Compressible SPH for Free Surface Flows," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '07)*, pp. 63-72, Aug. 2007.
[9] J. Chen and N. Lobo, "Toward Interactive-Rate Simulation of Fluids with Moving Obstacles Using Navier-Stokes Equations," *Graphical Models and Image Processing*, vol. 57, no. 2, pp. 107-116, 1995.
[10] N. Foster and D. Metaxas, "Realistic Animation of Liquids," *Graphical Models and Image Processing*, vol. 58, no. 5, pp. 471-483, 1996.
[11] N. Foster and D. Metaxas, "Controlling Fluid Animation," *Computer Graphics Int'l*, vol. 97, pp. 178-188, 1997.
[12] J. Stam, "Stable Fluids," *Proc. ACM SIGGRAPH '99*, pp. 121-128, 1999.
[13] R. Fedkiw, J. Stam, and H.W. Jensen, "Visual Simulation of Smoke," *Proc. ACM SIGGRAPH '01*, pp. 15-22, 2001.
[14] N. Foster and R. Fedkiw, "Practical Animation of Liquids," *Proc. ACM SIGGRAPH '01*, pp. 23-30, 2001.
[15] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki, "Realistic Animation of Fluid with Splash and Foam," *Computer Graphics Forum*, vol. 22, no. 3, pp. 391-400, 2003.
[16] G.D. Yngve, J.F. O'Brien, and J.K. Hodgins, "Animating Explosions," *Proc. ACM SIGGRAPH '00*, pp. 29-36, 2000.
[17] F. Losasso, F. Gibou, and R. Fedkiw, "Simulating Water and Smoke with an Octree Data Structure," *Proc. ACM SIGGRAPH '04*, pp. 457-462, 2004.
[18] G. Irving, E. Guendelman, F. Losasso, and R. Fedkiw, "Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques," *Proc. ACM SIGGRAPH '06*, pp. 805-811, 2006.
[19] B.M. Klingner, B.E. Feldman, N. Chentanez, and J.F. O'Brien, "Fluid Animation with Dynamic Meshes," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 820-825, 2006.
[20] N. Chentanez, B.E. Feldman, F. Labelle, J.F. O'Brien, and J.R. Shewchuk, "Liquid Simulation on Lattice-Based Tetrahedral Meshes," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '07)*, pp. 219-228, 2007.
[21] N. Chentanez, T.G. Goktekin, B.E. Feldman, and J.F. O'Brien, "Simultaneous Coupling of Fluids and Deformable Bodies," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '06)*, pp. 83-89, 2006.
[22] B.E. Feldman, J.F. O'Brien, B.M. Klingner, and T.G. Goktekin, "Fluids in Deforming Meshes," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '05)*, pp. 255-259, 2005.
[23] C. Batty, F. Bertails, and R. Bridson, "A Fast Variational Framework for Accurate Solid-Fluid Coupling," *Proc. ACM SIGGRAPH '07*, p. 100, 2007.
[24] Y. Zhu and R. Bridson, "Animating Sand as a Fluid," *Proc. ACM SIGGRAPH '05 Papers*, pp. 965-972, 2005.
[25] M. Carlson, P. Mucha, and G. Turk, "Rigid Fluid: Animating the Interplay between Rigid Bodies and Fluid," *Proc. ACM SIGGRAPH '04*, pp. 377-384, 2004.
[26] O. Génevaux, A. Habibi, and J. Dischler, "Simulating Fluid-Solid Interaction," *Proc. Graphics Interface Conf.*, 2003.
[27] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw, "Coupling Water and Smoke to Thin Deformable and Rigid Shells," *Proc. ACM SIGGRAPH '05*, pp. 973-981, 2005.
[28] Y. Liu, X. Liu, and E. Wu, "Real-Time 3D Fluid Simulation on GPU with Complex Obstacles," *Proc. 12th Pacific Conf. Computer Graphics and Applications (PG '04)*, pp. 247-256, 2004.
[29] C. Peskin, "Flow Patterns around Heart Valves: A Numerical Study," *J. Computational Physics*, vol. 10, pp. 252-271, 1972.
[30] C. Peskin, "The Immersed Boundary Method," *Acta Numerica*, vol. 11, pp. 479-517, 2002.
[31] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations," *J. Computational Physics*, vol. 161, no. 1, pp. 35-60, 2000.
[32] M. Uhlmann, "An Immersed Boundary Method with Direct Forcing for the Simulation of Particulate Flows," *J. Computational Physics*, vol. 209, pp. 448-476, 2005.
[33] J.-I. Choi, R.C. Oberoi, J.R. Edwards, and J.A. Rosati, "An Immersed Boundary Method for Complex Incompressible Flows," *J. Computational Physics*, vol. 224, no. 2, pp. 757-784, 2007.
[34] D. Le, B. Khoo, and J. Peraire, "An Immersed Interface Method for Viscous Incompressible Flows Involving Rigid and Flexible Boundaries," *J. Computational Physics*, vol. 220, no. 1, pp. 109-138, 2006.
[35] R. Fedkiw, "Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation with the Ghost Fluid Method," *J. Computational Physics*, vol. 175, no. 1, pp. 200-224, 2002.
[36] C. Hirt, A. Amsden, and J. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *J. Computational Physics*, vol. 14, no. 3, pp. 227-253, 1974.
[37] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw, "Two-Way Coupled SPH and Particle Level Set Fluid Simulation," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 4, pp. 797-804, July/Aug. 2008.
[38] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, and R. Fedkiw, "Two-Way Coupling of Fluids to Rigid and Deformable Solids and Shells," *ACM Trans. Graphics*, vol. 27, 2008.
[39] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, and M. Gross, "A Unified Lagrangian Approach to Solid-Fluid Animation," *Proc. Eurographics Symp. Point-Based Graphics (PBG '05)*, pp. 125-134, 2005.

[40] B. Solenthaler, J. Schläfli, and R. Pajarola, "A Unified Particle Model for Fluid Solid Interactions: Research Articles," *Computer Animation and Virtual Worlds,* vol. 18, no. 1, pp. 69-82, 2007.
[41] R. Keiser, B. Adams, P. Dutré, L. Guibas, and M. Pauly, "Multiresolution Particle-Based Fluids, " technical report, ETH Zurich, 2006.
[42] S. Falappi and M. Gallati, "SPH Simulation of Water Waves Generated by Granular Landslides," *Proc. 32nd Congress of the Int'l Assoc. Hydraulic Eng. and Research (IAHR),* 2007.
[43] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross, "Interaction of Fluids with Deformable Solids," *Computer Animation and Virtual Worlds,* vol. 15, no. 34, pp. 159-171, 2004.
[44] X. Hu and N. Adams, "A Multi-Phase SPH Method for Macroscopic and Mesoscopic Flows," *J. Computational Physics,* vol. 213, no. 2, pp. 844-861, 2006.
[45] J. Morris, P. Fox, and Y. Zhu, "Modeling Low Reynolds Number Incompressible Flows Using SPH," *J. Computational Physics,* vol. 136, no. 1, pp. 214-226, 1997.
[46] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant, "Two-Dimensional SPH Simulations of Wedge Water Entries," *J. Computational Physics,* vol. 213, no. 2, pp. 803-822, 2006.
[47] R. Weinstein, J. Teran, and R. Fedkiw, "Dynamic Simulation of Articulated Rigid Bodies with Contact and Collision," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 3, pp. 365-374, May/June 2006.
[48] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position Based Dynamics," *J. Visual Comm.,* vol. 18, no. 2, pp. 109-118, 2007.
[49] R. Weinstein, E. Guendelmann, and R. Fedkiw, "Impulse-Based Control of Joints and Muscles," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 1, pp. 37-46, Jan./Feb. 2008.
[50] M. Gissler, M. Becker, and M. Teschner, "Local Constraint Methods for Deformable Objects," *Proc. Virtual Reality Interactions and Physical Simulations (VriPhys '06),* pp. 25-32, Nov. 2006.
[51] E. Sifakis, T. Shinar, G. Irving, and R. Fedkiw, "Hybrid Simulation of Deformable Solids," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '07),* pp. 81-90, 2007.
[52] N. Thürey, K. Iglberger, and U. Rüde, "Free Surface Flows with Moving and Deforming Objects with LBM," *Proc. Vision, Modeling, Visualization (VMV '06),* pp. 193-200, 2006.
[53] E. Guendelman, R. Bridson, and R. Fedkiw, "Nonconvex Rigid Bodies with Stacking," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 871-878, 2003.
[54] R. Gingold and J. Monaghan, "Smoothed Particle Hydrodynamics—Theory and Application to Non-Spherical Stars," *Monthly Notices of the Royal Astronomical Soc.,* vol. 181, pp. 375-389, 1977.
[55] L. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis," *Astronomical J.,* vol. 82, no. 12, pp. 1013-1024, 1977.
[56] B. Adams, M. Pauly, R. Keiser, and L.J. Guibas, "Adaptively Sampled Particle Fluids," *Proc. ACM SIGGRAPH '07,* p. 48, 2007.
[57] N. Bell, Y. Yu, and P.J. Mucha, "Particle-Based Simulation of Granular Materials," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '05),* pp. 77-86, 2005.
[58] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, and M. Gross, "Optimized Spatial Hashing for Collision Detection of Deformable Objects," *Proc. Vision, Modeling, Visualization (VMV '03),* pp. 47-54, 2003.
[59] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Proc. ACM SIGGRAPH '87,* pp. 163-169, 1987.

Markus Becker received the diploma in mathematics from the University of Kaiserslautern in 2005. He is currently working toward the PhD degree in the Computer Science Department, Albert-Ludwigs-University Freiburg, where he is a member of the Computer Graphics Group. His research interests are fluid dynamics and deformable modeling.

Hendrik Tessendorf received the diploma in computer science from the Albert-Ludwigs-University Freiburg in 2008. His research interests are optimization techniques and shared-memory parallelization.

Matthias Teschner received the PhD degree in electrical engineering from the University of Erlangen-Nuremberg in 2000. He is a professor of computer science and the head of the Computer Graphics Group, Computer Science Department, Albert-Ludwigs-University Freiburg. From 2001 to 2004, he was a research associate at Stanford University and at ETH Zurich. His research interests include physical simulation, computer animation, computational geometry, collision handling, and real-time rendering with applications in entertainment technology and medical simulation. He has published about 60 research papers. He has served on program committees of major computer graphics conferences including Eurographics, IEEE Visualization, ACM SIGGRAPH/Eurographics Symposium on Computer Animation, and Computer Graphics International. He also participates in the review process of a number of major computer graphics journals.