MPM simulation of interacting fluids and solids

X. Yan†, C-F. Li², X-S. Chen¹ and S-M. Hu¹

¹Department of Computer Science and Technology, Tsinghua University, Beijing, China
²College of Engineering, Swansea University, Swansea, U.K.

Abstract
The material point method (MPM) has attracted increasing attention from the graphics community, as it combines the strengths of both particle- and grid-based solvers. Like the smoothed particle hydrodynamics (SPH) scheme, MPM uses particles to discretize the simulation domain and represent the fundamental unknowns. This makes it insensitive to geometric and topological changes, and readily parallelizable on a GPU. Like grid-based solvers, MPM uses a background mesh for calculating spatial derivatives, providing more accurate and more stable results than a purely particle-based scheme. MPM has been very successful in simulating both fluid flow and solid deformation, but less so in dealing with multiple fluids and solids, where the dynamic fluid-solid interaction poses a major challenge. To address this shortcoming of MPM, we propose a new set of mathematical and computational schemes which enable efficient and robust fluid-solid interaction within the MPM framework. These versatile schemes support simulation of both multiphase flow and fully-coupled solid-fluid systems. A series of examples is presented to demonstrate their capabilities and performance in the presence of various interacting fluids and solids, including multiphase flow, fluid-solid interaction, and dissolution.

CCS Concepts
•Computing methodologies → Physical simulation;

1. Introduction
Simulations of fluid flow and solid deformation are important in the visual effects industry. Most popular simulation methods are based on either particle- or grid-based solvers. Particle methods such as smoothed particle hydrodynamics (SPH) are popular for simulating liquids [MSKG05], deformable solids [MKN°04], granular materials, etc. Particle solvers are often simpler to implement, more robust in coping with geometric and topological changes, and more suitable for GPU-based parallel acceleration. However, they are often vulnerable to stability issues, and for example, near-surface SPH particle configurations may become distorted and form clusters [DR95, YL°16]. In comparison, grid-based methods can often achieve higher simulation accuracy for the same number of unknowns, and provide better simulation stability. Grid methods are particularly popular for simulating smoke [Stu99], fire [NF02], and deformable solids [CGF06]. However, they often have a higher computational cost, especially for large scenes, and it is harder to cope with complex free surfaces such as those found in splashes. Both particle- and grid-based methods are well established, and both support simulation of complex coupled systems, e.g. multiphase flow [RLY°14, YCR°15], and interacting and reacting fluids and solids [YL°16, YCL°17].

Recently, hybrid approaches combining strengths of both kinds of method have attracted increasing attention from the graphics community. The PIC/FLIP method was introduced by Zhu and Bridson [ZB05] for simulating sand; mass conservation is solved by particle advection while the constitutive and momentum equations are solved using a grid. The material point method (MPM) performs better for history-dependent materials; it was introduced to computer graphics by Stomakhin et al. [SSC°13]. It solves the constitutive equations using particles and computes forces by mapping the divergence of the stress tensor to a grid. While both FLIP and MPM share a conceptual basis with PIC, historically they have been considered separately by the engineering community, with FLIP for fluid simulation and MPM for solid simulation. Taking advantage of both particle and grid schemes, the adoption of FLIP and MPM represent a milestone for visual simulation. However, compared to particle and grid solvers, such hybrid methods are less well equipped for coupled systems involving multiple interacting fluids and solids.

Our work extends the MPM framework to permit simulation of multiple interacting fluids and solids. It is compatible with standard MPM. The new hybrid method solves the constitutive equations using particles, and the momentum equations on a grid, handling multiple liquids, gases, fluid-solid interactions, dissolution, etc. in a uniform manner. Our approach has the following advantages:
1. it provides a uniform MPM framework for simulating both fluids and solids,
2. it accurately models contact interactions between fluids and deformable solids,
3. it allows simulation of multiple fluids, gas and liquid, miscible and immiscible,
4. it can readily be extended to incorporate other interactions, such as the dissolution of solids in liquids.

2. Previous Work
Hybrid simulation methods have been extensively investigated, particularly FLIP and MPM methods; we briefly recap this work here. We also review the latest research on fluid-solid coupling and simulation of multiple fluids, and analyse its relationship to our work.

2.1. FLIP method
The FLIP method was introduced to the graphics community by Zhu and Bridson [ZB05] for sand simulation. Boyd and Bridson [BB12] extended it to multiFLIP to simulate two-phase fluids with glugging effects. Ando et al. [ATW15] used FLIP with stream functions to enforce incompressibility of fluids, and later they improved the computational efficiency of FLIP by use of narrow band calculations [FAW16]. Cornelis et al. [CIP14] proposed the IISPH-FLIP method for simulating incompressible fluids, in which IISPH and FLIP are combined to conserve mass and reduce the computational load. Yang et al. [YLH14] proposed a FLIP method for droplet and spray simulation. The FLIP approach has been very successful and popular, providing impressive fluid simulation results, but it is seldom used for simulating deforming solids, due to the difficulty of handling history-dependent materials when the constitutive equations are set up on a Eulerian grid.

2.2. Material point method
MPM was first introduced to computer graphics by Stomakhin et al. [SSC13] for the simulation of snow; it was further extended to handle phase change and varied materials [SSJ14]. MPM has also been used to simulate foams using a shear-dependent flow model [YSB15]. Jiang et al. [JSS15] proposed the affine particle-in-cell method, which avoids the noise encountered in FLIP. Klár et al. [KGP16], and Daviet and Bertails-Descoubes [DBD16] simulated granular materials with MPM. Ram et al. [RGI15] proposed an MPM scheme for viscoelastic fluids, foams and sponges. More recently, Tampubolon et al. [TGK17] proposed an MPM approach for simulating porous sand and water mixtures. Gao et al. [GTJS17] gave a generalized interpolating MPM method for elastoplastic materials, in which the specific region of interest is adaptively refined. Predominantly used for simulating solids, MPM is seldom used to model fluids or interactions of multiple fluids and solids.

2.3. Fluid-solid coupling
Various research works have considered fluid-solid coupling in different simulation contexts. Akinci et al. [AIA12] presented an SPH method to simulate fluid interacting with rigid bodies; it was later extended to include elastic solids [ACAT13]. This method was also used by Macklin et al. [MMCK14] in a unified particle framework. Yang et al. [YLM16] proposed a versatile approach for simulating interactions at solid-solid and fluid-solid interfaces. Batty et al. [BBB07] proposed a high performance variational framework for solid-fluid coupling. Guendelman et al. [GSLBO5] presented a solid-fluid coupling method for thin solid materials such as cloth. Robinson-Mosher et al. [RMG08] proposed a coupling method in which fluid simulation uses a Eulerian scheme and solid simulation uses a Lagrangian scheme. Narain et al. [NGL10] simulated the coupling between sand and rigid bodies. Carlson et al. [CMT04] proposed the rigid fluid method, which treats a rigid body as a fluid to solve the coupling between the rigid body and fluid. Robinson-Mosher et al. [RMEF09] proposed a method to more accurately calculate velocities in fluid-solid coupling. In all these coupling techniques, either the fluid and solid are simultaneously simulated with the same particle method, or are separately simulated with a particle solver and a grid solver. As a result, these coupling techniques cannot be directly transferred to hybrid approaches like FLIP and MPM, in which fluids and solids are simultaneously modelled with both particles and a grid.

2.4. Multiple-fluids simulation
Multiple fluid simulation has attracted much attention in recent years. Based on SPH, Müller et al. [MSKG05] proposed a fluid-fluid interaction method for this purpose. Solenthaler and Pajarola [SP08] addressed the density discontinuity problem. Losasso et al. [LSSF06] tracked the interface between fluids using the level set method in a grid-based solver. Misztal et al. [MEB12] used unstructured moving meshes to capture multiphase flow in immiscible fluids. Kang et al. [KPN10] considered miscible and immiscible fluids using a diffusion model. Ren et al. [RLY14] introduced the concept of volume fraction into SPH, using a mixture model to simulate various multi-fluid phenomena. This work was further extended for multiphase interaction by introduction of a Helmholtz energy [YCR15]. More recently, within the SPH framework, Yan et al. [YL16] simulated multiple fluids and solids as well as their various interactions. Shin et al. [SRC15] proposed a hybrid method for simulating viscous fingering phenomena and intermolecular diffusion. Recently, Gao et al. [GPH18] proposed an MPM approach for simulation of particle-laden flow and archived impressive results.

3. MPM simulation of fluids and solids
Fluids and solids both obey mass and momentum conservation laws. The major difference between them lies in the constitutive laws obeyed, i.e. the relationship between stress and deformation. The governing equations of a deformable solid can be expressed as:

$$\frac{\partial \rho}{\partial t} = 0, \quad \rho \frac{\partial u}{\partial t} = \nabla \cdot \sigma + pg$$

(1)

where $D(\cdot)/Dt$ denotes the material derivative, $\rho$ the density, $u$ the velocity, $g$ the external body force, and $\sigma$ the stress tensor. The governing equations of a non-viscous fluid are similar:

$$\frac{\partial \rho}{\partial t} = 0, \quad \rho \frac{\partial u}{\partial t} = -\nabla p + pg$$

(2)

where $p$ is the fluid pressure.
3.1. Basic MPM theory

In MPM, continua are sampled using Lagrangian particles, and space is covered by a Eulerian grid. The momentum equation is represented by particle advection. All derivative terms are computed on a grid and physical quantities are transferred between particles and the grid by an invertible particle-to-grid mapping. In the following description, we use \( A_i \) to represent the variable \( A \) on grid node \( I \) and \( a_i \) to represent the variable \( A \) on particle \( i \). The particle-to-grid mappings for mass \( m \) and momentum \( p \) are:

\[
m_I = \sum_{j \in H_I} m_j N_j(x_I),
\]

\[
p_I = \sum_{j \in H_I} p_j N_j(x_I) = \sum_{j \in H_I} \rho_j m_j N_j(x_I),
\]

where \( H_I \) represents all particles in the vicinity of grid node \( I \) and \( N_j(x_I) \) is the shape function. There are different options for the shape function, e.g., uniform GIMP [ZCL16], B-spline interpolation [SSJ14, SKB08] and adaptive GIMP [GTJS17]. For simplicity, we use uniform GIMP, defined as:

\[
N_j(x_I) = S_I \left( \frac{x_{I,j} - x_{I,x}}{h} \right) S_I \left( \frac{x_{I,y} - x_{I,y}}{h} \right) \times S_I \left( \frac{x_{I,z} - x_{I,z}}{h} \right)
\]

\[
S_I(x) = \begin{cases} 
7 - 10\alpha^2 & x \leq 0.25 \\
8 & 0.25 < x \leq 0.75 \\
(5-4\alpha)^2 & 0.75 < x \leq 1.25 \\
16 & \text{otherwise}
\end{cases}
\]

where \( x_{I,x}, x_{I,y}, x_{I,z} \) represent the coordinates of particle \( j \), \( x_{I,x}, x_{I,y}, x_{I,z} \) represent the coordinates of grid node \( I \), and \( h \) is the MPM grid spacing. Function \( S_I(x) \) is given by:

The velocity on grid node \( I \) is computed as:

\[
u_I = \frac{p_I}{m_I}
\]

The velocity gradient is computed on the grid and mapped back to the particles, which allows the constitutive equations to be solved on the particles. Then, the pressure and the stress are mapped from the particles to the grid, which allows the computation of the pressure gradient and stress divergence on the grid. The details are explained in § 3.2 and § 3.3.

The velocity and force equations are solved on the grid, and then mapped back to the particles for advection. The grid-to-particle mapping for velocity is:

\[
u_i = \sum_{j \in H_i} \frac{p_j N_j(x_i)}{m_j}
\]

where \( H_i \) represents all grid nodes in the vicinity of particle \( i \). The grid-to-particle mapping for force is:

\[
a_j = \sum_{j \in H_j} \frac{f_j N_j(x_j)}{m_j}
\]

where \( a_j \) denotes the acceleration of particle \( i \), and \( f_j \) is the force on the grid node \( J \).

To solve the momentum equation, the velocity term needs to be mapped to particles using Eqn. (8), which results in large dissipation. The common way of addressing this problem is to use FLIP/PIC interpolation [SSC13]. We map both acceleration and velocity from the grid to particles, and update the particle velocities as follows:

\[
\ddot{u}_i = a_i \Delta t + u_i^{\text{old}}, \quad u_i^{\text{new}} = \alpha u_i + (1 - \alpha) \ddot{u}_i,
\]

where \( \alpha \) is a control coefficient. A small \( \alpha \) value makes the material more active, which is more suitable for fluid simulation, while a large \( \alpha \) makes the material more stable. We set \( \alpha = 0.05 \) for fluids and \( \alpha = 1 \) for solids. The position of each particle is then updated as \( x_i^{\text{new}} = x_i + \Delta \ddot{x} \).

3.2. MPM solid simulation

The key to solving the solid equation Eqn. (1) is to compute the stress \( \sigma \). We take a stress rate approach. Specifically, the constitutive equation for a linear elastic solid can be expressed as:

\[
\sigma = C : \dot{\varepsilon} + K \text{Tr} \dot{\varepsilon} / 3 + 2G \nu \dot{\varepsilon} - \text{Tr} \dot{\varepsilon} / 3 + \dot{\omega} \cdot \sigma - \sigma \cdot \dot{\omega}
\]

where \( \sigma \) is the stress rate, \( C \) the stiffness tensor, \( K \) the volume bulk, \( G \) the shear modulus, \( \dot{\varepsilon} \) the identity tensor, \( \varepsilon \) the strain rate, and \( \omega \) the rotation rate tensor. The relationship between shear modulus and bulk modulus is \( G = E / (2(1 + \nu)) \) and \( K = E / [3(1 - 2\nu)] \) where \( \nu \) is the Poisson ratio. The strain rate and rotation rate tensors are defined as:

\[
\dot{\varepsilon}_i = \frac{1}{2} (\nabla u_i + \nabla u_i^T), \\
\dot{\omega}_i = \frac{1}{2} (\nabla u_i - \nabla u_i^T),
\]

where

\[
\nabla u_i = \sum_{j \in H_i} u_j \otimes \nabla N_j(x_i).
\]

After updating the stress rate for the particles, the stress can be computed as:

\[
\sigma_i^{\text{new}} = \sigma_i^{\text{old}} + \dot{\sigma} \Delta t.
\]

The stress divergence in Eqn. (1) is computed as:

\[
f_{I,\text{stress}} = - \sum_{j \in H_I} p_j \nabla N_j(x_j)
\]

where \( \rho_j \) is the density of particle \( j \).

MPM overcomes tensile instability that occurs commonly in SPH simulation; it is caused by the poor distribution of particles near the surface. Some research has been undertaken to address tensile instability [CBP05, SB12, MM13, HWZ14], but these methods...
require additional calculations. This issue is avoided in MPM as all velocity gradients are interpolated on grid nodes. Figure 2 illustrates the problem, and MPM’s resolution of it.

3.3. MPM fluid simulation

Solving the fluid equation, Eqn. (2), may be done in a similar manner to solving for solids, but the key is now to compute the fluid pressure, which depends on the fluid density. There are different ways to do this, using different interpolation schemes. [SSC+13] and [ZCL16] estimate the fluid density at position $x$ as:

$$\rho(x) = \sum_{j \in H} m_j \frac{\theta(x - x_j)}{V_j},\quad(16)$$

where $V_j^p = h^3$ is the particle volume, $\theta(x - x_j)$ can be the shape function or Dirichlet function, and $H$ denotes the particle set in the vicinity of $x$. An alternative approach is to use SPH interpolation:

$$\rho(x) = \sum_{j \in H} m_j W(x - x_j, h_p)\quad(17)$$

where $W(x - x_j, h_p)$ is the kernel function and $h_p = 1.25h$ is the smoothing radius. Our limited experiments indicate that SPH interpolation performs better.

After obtaining the particle density, the particle pressure can be computed according to the weakly compressible equation [TGK+17] by:

$$p_i = k_s((\rho_i/\bar{\rho})^7 - 1)\quad(18)$$

where $k_s$ is the stiffness coefficient, $p$ pressure, and $\bar{\rho}$ rest density. Finally, the pressure gradient in Eqn. (2) can be calculated as:

$$f_{\text{pressure}} = \sum_{j \in H_i} m_j \rho_j \nabla N_i(x_j)\quad(19)$$

3.4. MPM work flow

\begin{algorithm}
\begin{algorithmic}[1]
\State repeat
\For {each grid node $I$}
\State map mass and momentum to the grid using Eqns. (3,4) and compute velocity with Eqn. (7) (See Fig. 3(a))
\EndFor
\For {each particle $i$}
\State compute velocity gradient for solid particles with Eqn. (13) and compute pressure for fluid particles with Eqn. (18) (See Fig. 3(b))
\EndFor
\For {each grid node $I$}
\State compute the divergence or the gradient of the pressure and stress tensor on the grid with Eqns. (15,19) (See Fig. 3(c))
\EndFor
\For {each particle $i$}
\State map force and velocity from grids to particles using Eqns. (8,9) (See Fig. 3(d))
\State update acceleration and velocity for each particle using Eqn. (10)
\EndFor
\State $t \leftarrow t + \Delta t$
\Until {end of simulation}
\end{algorithmic}
\end{algorithm}

The MPM work flow is summarized in Algorithm 1, and a graphical illustration is given in Figure 1. We use a leapfrog approach to
offset the velocity $u$, $p$, $\sigma$, $\epsilon$ and $\omega$ for $1/2$ time steps. Other variables such as $\epsilon$, $\omega$, $\sigma$ and $f$ are not offset. In detail, for the $n$th time step, the leapfrog computation can be summarized as follows:

1. The grid velocity $(\mathbf{u}_I)^{n-1/2}$ is updated via the grid momentum $(\mathbf{p}_I)^{n-1/2}$.
2. The velocity gradient $(\nabla \mathbf{u}_I)^{n-1/2}$ is updated by $(\mathbf{u}_I)^{n-1/2}$ to compute $(\epsilon_I)^{n-1/2}$ and $(\omega_I)^{n-1/2}$, and we then update $\sigma_I$.
3. The stress $(\sigma_I)^{n-1/2} = (\sigma_I)^{n-1} + \Delta t(\sigma_I)^{n-1/2}$ is computed to update the force term $f_I^0$ and sequentially the velocity $(\mathbf{u}_I)^{n+1/2}$ and $(\mathbf{u}_I)^{n+1/2}$.

4. Coupling and interaction between fluids and solids

As explained in §3, MPM solid simulation and MPM fluid simulation share the same work flow, with the main difference lying in the treatment of the constitutive equations. As both solid and fluid equations are handled by particle advection, parallel computing may be readily used in both cases. As all spatial derivatives are computed on a grid, both solid and fluid simulations are potentially more accurate and more stable than those produced by conventional particle methods. However, the lack of efficient and robust coupling and interaction techniques remains a major hurdle hampering the wider application of MPM.

Two types of interactions are considered in this work: fluid-fluid interaction and fluid-solid interaction. The former deals with multiple fluids; both immiscible and miscible fluids are considered. Fluid-solid interaction can also be considered as being immiscible or miscible. In the former case, their interaction is treated as coupling through contact forces. In the latter case, their interaction is regarded as a mixing process, i.e. the solid dissolves. Solid-solid interaction in the MPM framework has been well studied and documented elsewhere [ZCL16, SSC*13], so is not considered in this work. Tampubolon et al. [TGK*17] used two background grids for sand and water simulation. In our approach, we use a single background grid for all different material phases.

4.1. Coupling between fluids and solids

Evaluating fluid-solid coupling involves two tasks:

1. computing the contact force between fluid and solid,
2. preventing the fluid from penetrating the solid.

Dealing with the contact force between fluid and solid is straightforward in particle methods, but how to do so is less clear for MPM because inter-particle forces are not considered as part of the computation. Instead, for a given grid node, we compute separately a force term from the fluid pressure and a force term from the solid stress, which are then combined to give the total force on grid node $f_{I, \text{total}}$:

$$f_{I, \text{total}} = -f_{I, \text{pressure}} - f_{I, \text{stress}}.$$  \hspace{1cm} (20)

After obtaining the total force, we update the velocity on the grid, and map the velocity and force from the grid to both fluid and solid particles. The fluid and solid velocities are driven individually by the fluid pressure and the solid stress respectively. By using the shared force term $f_{I, \text{total}}$ at the fluid-solid interface, we ensure consistent velocities at the interface.

As Eqn. (10) is used to control dissipation in fluid simulation, it is difficult to prevent fluid-solid penetration. In SPH simulations, it is typically prevented by adding artificial forces to particles [AIL*12], but this simple treatment does not work here as inter-particle forces are not intrinsically incorporated in MPM. Stomakhin et al. [SSS*14] proposed a collision detection method to handle collision, but it relies on interface tracking, which is rather inconvenient in the presence of multiple fluids and solids. Instead, we propose an anti-penetration model consistent with the MPM workflow that prevents fluid-solid penetration on the grid. Let $p_i^f$ denote the momentum of the fluid at grid node $I$, and $p_i^s$ the momentum of the solid. We map the fluid and solid velocities onto the grid, and allow the fluid and the solid to fully collide on the grid node and reach the same velocity. Based on momentum conservation, the interaction force during this collision process can be derived at the grid node:

$$f_{I, \text{collision}} = p_i^f m_i^f - p_i^s m_i^s / (m_i^f + m_i^s) \Delta t$$ \hspace{1cm} (21)

where $\Delta t$ is the time step. Here, we ignore the viscous force of the fluid, so the contact force between fluid and solid is merely a normal force at the interface; it may be estimated as below:

$$f_{I, \text{ap}} = -f_{I, \text{ap}} = \beta [f_{I, \text{collision}} \cdot n_I] n_I,$$ \hspace{1cm} (22)

where $n_I$ is the normal direction at the interface and $\beta$ is a control coefficient (set to $1/2$ in simulations for efficient prevention of penetration). The normal direction $n_I$ is estimated from the mass distribution of fluid and solid:

$$\bar{n}_i^f = \sum_{j \in H_f^I} m_j \nabla N_j(x)_i / \left| \sum_{j \in H_f^I} m_j \nabla N_j(x)_i \right|,$$ \hspace{1cm} (23)

$$\bar{n}_i^s = \sum_{j \in H_s^I} m_j \nabla N_j(x)_i / \left| \sum_{j \in H_s^I} m_j \nabla N_j(x)_i \right|,$$ \hspace{1cm} (24)

$$n_I = -n_I = \bar{n}_I - \bar{n}_I / \left| \bar{n}_I - \bar{n}_I \right|,$$ \hspace{1cm} (25)

where $H_f^I$ and $H_s^I$ are the fluid particle set and the solid particle set in the vicinity of grid node $I$. The anti-penetration force $f_{I, \text{ap}}$ is mapped from the grid to fluid and solid particles separately in opposite directions $n_I^f$ and $n_I^s$. Note that this is only done when a collision is detected on the grid, i.e. when $\left( \bar{u}_I - \bar{u}_I \right) \cdot n_I > 0$. The current velocity $\bar{u}_I$ is updated on the grid using the grid force:

$$\bar{u}_I = \bar{u}_I + f_{I, \text{ap}} \Delta t / m_I.$$  \hspace{1cm} (26)

Figure 3 illustrates the computation involved in fluid-solid coupling, while its workflow is summarized in Algorithm 2.

4.2. Multi-fluid simulation

Simulation of multiple fluids has already been considered using the SPH framework with impressive results: fluid-fluid interaction is modelled using a mixture model in [RLY*14] and a Helmholtz free
§ 4.1. Specifically, all pressure forces are added up on grid nodes and then mapped to particles. To prevent interpenetration of immiscible fluids, an anti-penetration force is introduced at the fluid-fluid interface:

\[ f_{I, \text{contact}}^f = -f_{I, \text{contact}}^f = \beta \left( \frac{p_i^f m_i^f - p_j^f m_i^j}{(m_i^f + m_i^j) \Delta t} \right) \mathbf{n}_I. \] (26)

The superscripts \( f_1 \) and \( f_2 \) denote two immiscible fluids, while \( \beta \) is a control coefficient. Larger \( \beta \) prevents penetration more strictly (it is set to 0.3 in our simulations). The contact force is 0 if \((\hat{u}^{f_1}_I - \hat{u}^{f_2}_I) \cdot \mathbf{n}_I < 0\). For a fluid mix with three or more different fluids, we treat each fluid phase separately, with all other phases considered as one single phase.

**Miscible fluids** To model interaction between miscible fluids, we use a diffusion model:

\[ \frac{D\alpha^k}{Dt} = k_d \nabla^2 \alpha^k, \] (27)

where \( k_d \) is the diffusion coefficient, \( \alpha^k \) the concentration of the fluid phase \( k \), and \( \sum_k \alpha^k = 1 \). In the MPM formulation, the diffusion equation is handled by concentration advection, where the associated spatial derivatives are computed on the grid using:

\[ \nabla \alpha^k = \frac{\sum m_i \alpha^k \nabla N_I(x_i) \| \sum m_i \alpha^k \nabla N_I(x_i) \|}{\sum m_i \alpha^k \nabla N_I(x_i)}, \] (28)

\[ \nabla^2 \alpha^k = \sum m_i \alpha^k \cdot \nabla N_I(x_i). \]

Then, we use \( \nabla^2 \alpha^k \) to compute the material derivative of \( \alpha^k \) in Eqn. (27) and update the fluid concentrations. After doing so on particles, the mass and density values are updated accordingly: the particle mass is computed as \( m_i = \sum \alpha^k \hat{m}^k \), where \( \hat{m}^k \) is the rest mass of phase \( k \) on each particle, and the particle rest density is computed as \( \bar{\rho} = \sum \alpha^k \hat{\rho}^k \), where \( \hat{\rho}^k \) is rest density of phase \( k \).

### 4.3. Dissolution

If a solid is miscible with a fluid, the interaction takes the form of dissolution, instead of contact. Dissolution processes such as sugar dissolving in hot drinks are common in daily life, and they can be...
described by the Noyes-Whitney equation [NW97]:

$$\frac{DC}{Dt} = \kappa(C_s - C),$$

(29)

where \(C\) represents the volume fraction of the dissolved solid phase, \(C_s\) is the saturation volume fraction, and \(\kappa\) is the dissolution coefficient.

Let \(s\) denote the solid phase that is soluble in the fluid phase \(f\). To solve Eqn. (29) in the MPM framework, we first need to compute the concentration \(C\) at grid node \(I\):

$$C_I = \frac{\sum_{j \in H_f^I} m_j \alpha_s^j, N_j(x_j)}{m_I^f},$$

(30)

where \(H_f^I\) represents all fluid particles in the vicinity of node \(I\), \(m_j\) the particle mass, \(\alpha_s^j\) the solid volume fraction on the fluid particle \(j\), and \(m_I^f\) the fluid mass at node \(I\). Then the concentration change can be computed on the grid using:

$$\Delta C_I = \kappa(C_s - C_I) dt.$$

(31)

Finally, this concentration change is mapped from the grid to the particles, allowing updating of the particle volume fraction of the solid phase:

$$\Delta \alpha_s^i = \sum_{j \in H_f^i} \frac{m_j^s}{m_j^f + m_j^s} \Delta C_I N_j(x_i),$$

(32)

for fluid particle \(i\),

$$\Delta \alpha_s^i = -\sum_{j \in H_f^i} \frac{m_f^i}{m_f^j + m_f^i} \Delta C_I N_j(x_i),$$

for solid particle \(i\),

where \(m_I^s\) is the solid mass at grid node \(I\), \(m_I^f\) is the fluid mass there, and \(H_f^i\) represents all grid nodes in the vicinity of particle \(i\). The volume fractions can be readily updated for particles using

$$\alpha_s^i = \alpha_s^i + \Delta \alpha_s^i, \quad \alpha_f^i = 1 - \alpha_s^i.$$

In our implementation, we label each particle as a fluid or solid particle. If the concentration \(\alpha_s^j\) of solid \(s\) in the solid particle meets \(C_s\), we relabel this solid particle as a fluid particle; the fluid particles never become solid particles in our approach. Both \(\alpha\) and \(C\) can be computed independently from the standard MPM framework. In our approach, we update \(\alpha\) and \(C\) after line 18 in Algorithm 2.

5. Results

We have implemented our approach on an NVIDIA GeForce GTX980 16GB GPU. A series of examples is presented to demon-
strate the simulation capacity of the proposed hybrid approach. Corresponding performance data are given in Table 1.

**Fluid simulation** Case 1 (see Fig. 4) shows a dam-break example, where a block of water collapses due to gravity, strikes the wall, splashes into the air, and gradually settles under gravity. This standard test confirms that the proposed MPM approach can successfully simulate single-phase fluid flow.

**Coupling between fluid and solid** Case 2 (see Fig. 5) is designed to test fluid-solid coupling. Three elastic balls are dropped into water separately; the geometry in all three tests remains constant. The three balls also have identical shear modulus and Poisson ratio, $G = 10^5$ and $v = 0.5$. The balls have different densities, with density ratios between ball and water of $1 : 1$, $1 : 2$ and $1 : 3$ respectively. The heaviest ball drops deeply into the water producing a large hole, and is then covered by water directly. The medium ball hits the water making a small hole, and bounces up and down before becoming partially immersed in the water. The lightest ball hits the water with little penetration, and then drifts on the water surface. This example confirms that our approach can correctly capture dynamic buoyancy effects. An efficiency comparison is made between simulations with and without the anti-penetration force.

**Multiple fluid simulation** Case 3 (see Fig. 6) shows three dam breaks with different fluids. The red and green liquids are miscible with each other, but they are both immiscible with the blue liquid. The densities of the blue, red and green liquids are set to 3, 2 and 1 respectively, and the diffusion coefficient is set as $k_d = 0.0003$. Driven by gravity, the three liquids collapse and interact with each other, forming a complex mix. As time passes, the green and red liquids become fully mixed with each other, but the mix remains separate from the blue liquid, with a sharp interface. This example confirms that the proposed MPM approach can correctly handle fluid-fluid interaction, capturing both miscible and immiscible effects.

**Smoke simulation** Case 4 (see Fig. 7) shows an example of rising smoke, which is simulated as a two-phase flow, with smoke and air phases. The densities of smoke and air are set to 1 and 2 respectively, allowing the smoke to rise while retaining visible interaction with the surrounding air. To accelerate the rate of rising of the smoke, an additional temperature field is also introduced to add artificial buoyancy effects. Temperature evolution follows a diffusion model, which is solved in a similar way to Eqn. (32) using the MPM formulation. For clarity of demonstration, the smoke phase is assumed to be immiscible with the air phase. In Fig. 7, the results on the left are obtained using the fluid-fluid interaction Eqn. (26), while the results on the left are obtained without fluid-fluid interaction. Clearly, the fluid-fluid interaction is essential to keep the smoke in the form of a mushroom cloud, instead of it moving randomly under the artificial buoyancy.

**Dissolution** Case 5 (see Fig. 8) shows an elastic bunny dissolving into water. After a dam break, water crashes into the elastic bunny and causes it to drift around. During this time the bunny dissolves gradually into the water, turning it red. The densities of bunny and water are set to 1 and 2 respectively. The saturation concentration is $0.5$ and the dissolution coefficient is $0.0003$. This example confirms that the proposed MPM approach can deal with miscible solids and fluids.

6. Conclusions

In this paper, we have developed a unified MPM framework with efficient and versatile coupling techniques for multiple fluid-fluid and fluid-solid interactions; we have demonstrated the new approach on a wide range of challenging phenomena. For fluid-solid interactions, we give a method to prevent the fluid from penetrating the solid, and an MPM-based dissolution simulation method based on the Noyes-Whitney equation. For fluid-fluid interactions, we proposed a multi-fluid scheme for both miscible and immiscible fluids in the MPM framework. Using our MPM-based multi-fluid simulation method, smoke can be simulated by incorporating air particles into the scene to interact with smoke particles.

The MPM scheme has the intrinsic advantage of readily coping with geometric and topological changes, and the convenience of ease of GPU implementation. Compared with the SPH method, MPM interpolates physical quantities using the background grid, thereby improving stability during solid simulation, and avoiding the tensile instability found in SPH (see Fig. 2). In SPH, fluid-solid coupling [AIA*12, ACAT13], multiple fluids [RLY*14], and dissolution [YJL*16] have been developed more thoroughly. However, wider applications of MPM have to now been hampered by the lack of robust and flexible interaction and coupling techniques between multiple fluids and solids. We have extended the MPM framework with robust coupling schemes to fill this gap.

7. Limitations and future work

The main limitation of the proposed MPM approach lies in its method of fluid simulation, in particular for incompressible fluids. We use the weakly compressible equation to obtain the pressure...
Table 1: Performance data; the time step used in all cases is $10^{-4}$ s.

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
<th>Phases</th>
<th>Particles</th>
<th>Performance (steps/s)</th>
<th>Performance (frame/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single phase dam break</td>
<td>1</td>
<td>$3.42 \times 10^3$</td>
<td>55 steps/s</td>
<td>0.55 frame/s</td>
</tr>
<tr>
<td>2</td>
<td>Ball in water</td>
<td>2</td>
<td>$5.56 \times 10^5$</td>
<td>21 steps/s</td>
<td>0.42 frame/s</td>
</tr>
<tr>
<td>3</td>
<td>Multiple fluids</td>
<td>3</td>
<td>$3.94 \times 10^5$</td>
<td>56 steps/s</td>
<td>1.01 frame/s</td>
</tr>
<tr>
<td>4</td>
<td>Smoke</td>
<td>2</td>
<td>$1.66 \times 10^6 - 1.85 \times 10^6$</td>
<td>11 steps/s</td>
<td>0.44 frame/s</td>
</tr>
<tr>
<td>5</td>
<td>Bunny dissolution</td>
<td>2</td>
<td>$1.63 \times 10^5$</td>
<td>67 steps/s</td>
<td>1.33 frame/s</td>
</tr>
</tbody>
</table>

Figure 7: Rising smoke; the smoke is injected from the ground. Left: Good results with fluid-fluid interaction. Right: Poor results without fluid-fluid interaction.

Figure 8: Left to right, top to bottom: An elastic red bunny dissolves in water, turning it red.

from the fluid density. Pressure accuracy drops at the simulation boundary where the particle distribution is typically poorer. A more accurate approach would be to enforce the incompressibility condition and solve for the pressure implicitly. This is beyond the scope of this work, but is worth pursuing in future research.

For fluid-solid coupling, we use an anti-penetration force which results in a small gap between the solid and the fluid. When the solid is soluble, the gap will slow down its dissolution. If the force is very large, it may even prevent it. In order to maintain a normal dissolution rate, the larger the anti-penetration force is, the greater the control coefficient $\beta$ in Eqn. (26) needs to be to compensate.

Since the coarse background grid smooths the flow field during grid-based interpolation, the simulated liquid will seem more viscous. One solution is to decrease $\alpha$ in Eqn. (10) so that the kinetic energies of particles increase. However, this will cause the particles to become more disordered. Activity and stability form a trade-off in MPM. Furthermore, a multi-fluid scheme is used to simulate gas in the MPM framework, but in order to achieve detailed smoke simulation, more air particles are needed. Refining the simulation can address these two issues, at the cost of increased simulation time. Some recent research promises to be relevant, e.g. using adaptive GIMP [GTJS17], and will be pursued in future work.

Acknowledgements

This work was supported by the National Key Technology R&D Program (Project Number 2017YFB1002701), the Natural Science Foundation of China (Project Number 61521002). The authors would like to thank Professor Ralph R. Martin at Cardiff University, UK, for his constructive comments on this work and the help on the English presentation.


